Chapter 10

Model for dynamic website optimization

In this chapter we develop a mathematical model for dynamic website optimization. The model is designed to optimize websites through autonomous management of design patterns via analysis of user behavior. To incorporate autonomous management in our model, the model needs to infer user preferences through the click behavior of web visitors when presented with pages with different design patterns. At the same time, the model needs to select the design patterns that result in optimized web pages. Hence, there is a tension between exploration (learning) and exploitation (optimal selection). We start by explaining this tension in Section 10.1 and resolve this issue by developing a Bayesian Markov decision model. The model is hard to solve in practice, however, we develop an efficient implementation of which the details are discussed in Section 10.2. A proof of concept of the model has been build to optimize websites in practice. This is discussed in Section 10.3. In Section 10.4 we illustrate how a website with two different groups of web visitors can be optimized for wild searching. By means of numerical experiments we evaluate in Section 10.5 the performance of our algorithm for dynamic website optimization. Finally, in Section 10.6 we end the chapter with conclusions.
10.1 Mathematical model

In this section we develop a mathematical model that on the one hand analyzes user behavior for a given website, and on the other hand dynamically optimizes the website given the analysis of the user behavior. This requires the model to provide a balance between learning and optimization. To illustrate the tension between learning and optimization, we discuss the following example first.

Consider two versions, say A and B, of the same website, i.e., two websites with the same content but with different looks using potentially different design patterns. Suppose that we present 30 web visitors with the two different versions, and 10 of the web visitors like version A better and 20 like version B better. Is this conclusive to rank version B higher than version A? What if the ratios were 20 to 40? In both cases twice as many people like version B better, but the latter case provides more information than the former. Statistically, the case with the 20 to 40 ratio is based on a larger sample size and provides a better estimate for the preference in the population. In this example, both websites were shown to the web visitors, however, in practice, only one website can be shown. It this case one needs to make a decision: does one present the ‘better’ version of the website, or does one need to gather more information on the other version to form a better judgement on the rankings of both versions. It is this tension between learning (i.e., gathering more information by potentially not choosing the website with the highest ranking) and optimization (i.e., choosing the best website based on the estimates) that the model should take into account.

10.1.1 The theory and practice of website optimization

As we have seen in the previous chapter, website optimization is still in its infancy and a few commercial companies already offer basic tools to support website optimization. For example, Clicktale [64], Mouseflow [161], and Userfly [226] are three companies that record user behavior on webpages by storing mouse clicks and mouse movements. Based on this data, a heat map is generated. This data, however, is not used in an optimization algorithm that solves the webpage selection problem. Google Website Optimizer [180] provides website optimization functionality that comes closest to our objective. They present different versions of websites and generate reports that provide statistics on the different website versions. Google Website Opti-
mizer gathers data and generates report, and does not solve the website selection problem in which the tension between learning and optimization is addressed.

We have also seen in Chapter 8 that in the model for the click behavior of web visitors the reward function was difficult to specify in practice. Therefore, there is a need for a model that is specified from the viewpoint of the web visitor. From this viewpoint, it is mainly relevant if the web visitor has achieved his goal, whereas his complete history of click behavior is of minor importance. Thus, one can make a model in which the number of successful visits (the visitor has reached his goal) and the number of failed visits (the visitor did not reach his goal) can be counted. Based on this information, the right website can then be selected.

From a mathematical point of view, the website selection problem, in its simplest form, can be classified as a standard class of problems, known as the class of multi-armed bandit problems [42]. Multi-armed bandit problems are selection problems in which the decision maker is faced with several one-armed bandits (i.e., slot machines that are found in the casino), each having their own payout rate which is unknown to the decision maker. The decision maker has the objective to maximize his revenues while he learns about the uncertain payout rates of the different arms. In our case, the one-armed bandit can be identified with a version of the website, and the payout rate is equivalent to the success probability.

The multi-armed bandit problem is a hard problem for which the optimal strategy cannot be stated explicitly. Sondik [209] develops algorithms to numerically compute the optimal strategy. Gittins [93] characterizes the optimal selection strategy through Gittins indices, but the indices are hard to compute as well. Kumar [134] gives an overview of several methods to deal with the unknown parameters. Several authors have tried to exploit the relationship between learning and optimization to develop simple selection rules [94, 43, 50]. However, despite all the literature, there is no analytical characterization of the optimal policy for the multi-armed bandit problem.

In the sequel, we cast the website selection problem as a multi-armed bandit problem. We incorporate learning in our mathematical model by using the Bayes’ rule. This, however, leads to a high-dimensional state space so that numerically deriving the optimal strategy becomes intractable. To overcome this problem, we develop sufficient statistics to reduce the dimensionality of the state space, which leads to an efficient numerical procedure to solve the website selection problem. In the following subsection we explain
all the intermediary steps.

10.1.2 Bayes’ rule

In our website optimization problem, we want to create a model that forms a judgement of different versions of a website based on data on user behavior. We will use Bayes’ rule to model learning based on the data. To intuitively explain the formula, let $X$ denote the judgement of the websites, and let $Y$ denote the data, i.e., user behavior. Then, Bayes’ rule states that

$$P(X | Y) = \frac{P(Y | X)P(X)}{P(Y)}.$$  \hfill (10.1)

In words, the formula states that in order to form a judgement of the website given the data (i.e., $X | Y$), one needs to explain the data where we take the judgement as given (i.e., $Y | X$). The formula represents, in a discrete probabilistic setting, the way both situations are related to each other. In a continuous probabilistic setting (using probability densities $f$), the equivalent of Bayes’ rule above is given by

$$f_X(x | Y = y) = \frac{f_Y(y | X = x)f_X(x)}{f_Y(y)}.$$  \hfill (10.2)

10.1.3 Markov decision problem

We now proceed to the formulation of the Markov decision problem for website optimization. To this purpose, we assume that we have $N$ versions of a website. This represents different versions with different uses of design patterns; these versions are generated automatically and use a mix of design patterns that address well-known website issues as described in the transition diagrams in Section 7.5. To each website $i$, we assign a probability $\theta_i \in [0, 1]$ that plays the role of the utility of the user for the website for $i = 1, \ldots, N$. Intuitively one can say that the probability that a user will achieve his goal when presented with website $i$ is represented by $\theta_i$, the utility. This utility represents the overall user judgement of the website, where 1 is the highest ranking and 0 the lowest. Alternatively, when $\theta_i$ is multiplied by 10, then this could be seen as a grade attached to that website. In practice, the utility vector $\theta$ cannot be observed.

Each time a web visitor wants to visit the website, we need to decide which version $i$ to display, for $i = 1, \ldots, N$. This decision is based upon
the utility vector $\theta$ that is not available. Therefore, we formulate a Markov
decision problem that learns the value of $\theta$ while at the same time tries to
optimally select the versions of the website to display such that the utility
is maximized. In this decision making we need to incorporate the tension
between learning the value of $\theta$ and maximizing the utility as discussed
above.

To use Bayes’ rule in our formulation, we need to assume a prior distri-
bution $f_i$ on the possible values of $\theta_i$. This prior distribution will then be
used in Equation (10.2) to derive an updated distribution, called the pos-
terior distribution. In general, starting with an arbitrary prior distribution
can lead to a posterior distribution that does not follow an easy mathemat-
ical description. However, for mathematical tractability, we want to have a
class of distributions from which we choose the prior distribution such that
the posterior distribution remains within the same class. This class of dis-
tributions is called a conjugate family of distributions. In our setting, this
conjugate family is given by the class of all Beta distributions (see, DeG-
root [74]). The Beta distribution for $\theta_i$ is parameterized by $\alpha_i$ and $\beta_i$ to
determine the shape of the distribution. The probability density is given by

$$f_i^{\alpha_i+1,\beta_i+1}(x) = \frac{\alpha_i + \beta_i + 1}{\alpha_i!\beta_i!} x^{\alpha_i} (1 - x)^{\beta_i}, \quad (10.3)$$

for $\alpha_i, \beta_i \geq 0$ and $x \in [0, 1]$.

Now suppose that version $i$ is shown to a web visitor, and the web visitor
has a positive evaluation (i.e, a success, denoted by $Y = 1$) of that version.
Then the posterior distribution given this outcome is calculated as

$$\mathbb{P}(x \leq \theta_i \leq x + h \mid Y = 1) = \frac{\mathbb{P}(x \leq \theta_i \leq x + h, Y = 1)}{\mathbb{P}(Y = 1)}$$

$$= \frac{\mathbb{P}(x \leq \theta_i \leq x + h)}{\mathbb{P}(Y = 1)} \cdot \frac{\mathbb{P}(x \leq \theta_i \leq x + h, Y = 1)}{\mathbb{P}(x \leq \theta_i \leq x + h)}$$

$$= \frac{\mathbb{P}(x \leq \theta_i \leq x + h)\mathbb{P}(Y = 1 \mid x \leq \theta \leq x + h)}{\mathbb{P}(Y = 1)}.$$

Dividing by $h$ and taking the limit $h \to 0$ gives (with $f_Y$ denoting the density
of $Y$)

$$f_{\theta_i \mid Y=1}(x) = \frac{f_{\theta_i}(x)\mathbb{P}(Y = 1 \mid \theta_i = x)}{\mathbb{P}(Y = 1)} = \frac{xf_{\theta_i}(x)}{\mathbb{P}(Y = 1)}.$$
Hence, this result tells us that when we have a Beta distribution with parameters $\alpha_i$ and $\beta_i$, the update when a success is observed is given by a Beta distribution with parameters $\alpha_i + 1$ and $\beta_i$. Similarly, the same line of argument shows that when a failure is observed then the posterior distribution is given by a Beta distribution with parameters $\alpha_i$ and $\beta_i + 1$. This is a useful result, since it allows us to formulate a Markov decision problem without the need to store complete distributions. Instead we can suffice with only the parameters of the distributions, where $\alpha_i$ and $\beta_i$ count the number of successes and failures, respectively.

We now have all ingredients to formulate the Markov decision problem for website optimization. Define the state space by $S = \{0, 1, \ldots\}^{2N}$. An element $s \in S$ is represented by $s = (\alpha_1, \beta_1, \ldots, \alpha_N, \beta_N)$ providing information on the parameters of all the different versions of the website. Define the action space by $A = \{1, 2, \ldots, N\}$, i.e., the different versions of the website that can be selected to display. The transition probabilities are given by

$$p(s' \mid s, a) = \begin{cases} \frac{\alpha_a+1}{\alpha_a+\beta_a+2}, & \text{for } s' = s + e_{2a-1} \\ \frac{\beta_a+1}{\alpha_a+\beta_a+2}, & \text{for } s' = s + e_{2a} \\ 0, & \text{otherwise} \end{cases}$$

where $e_i$ is the vector with all zeros except with the $i$-th entry a 1. The transition probabilities determine how the posterior distribution is determined based on the observation. Given that in state $s$ action $a$ is chosen, the probability of observing a success is given by the expectation of the Beta distribution with parameters $\alpha_a$ and $\beta_a$. This expectation is exactly $(\alpha_a + 1)/(\alpha_a + \beta_a + 2)$. In that case $s + e_{2a-1}$ results in a state in which $\alpha_a$ is changed to $\alpha_a + 1$. Similarly, a failure is observed with probability $(\beta_a + 1)/(\alpha_a + \beta_a + 2)$. The posterior distribution is then reflected by state $s + e_{2a}$ which updates $\beta_a$ to $\beta_a + 1$. Finally, we take as reward function the utility of the website in case of a success. Thus $r(s, a) = (\alpha_a + 1)/(\alpha_a + \beta_a + 2)$.

The tuple $(S, A, p, r)$ determines the Markov decision problem for the website optimization problem. The description of the Markov decision problem is now completed by describing the criterion function. For that purpose, let $\gamma \in (0, 1)$ be the discount factor, and $\pi$ be a fixed strategy. Then the discounted reward criterion function $V^\pi(s)$ is defined by

$$V^\pi(s) = \mathbb{E}_s^\pi \sum_{t=0}^{\infty} \gamma^t r(S_t, A_t),$$
with $S_t$ and $A_t$ the random variables denoting the state and action at time $t$, respectively. The Markov decision problem is to find a strategy $\pi^*$ such that $V(s) = V^{\pi^*}(s) = \sup\{V^{\pi}(s)\}$. Since the rewards are bounded by 1, it follows that there exists an optimal deterministic stationary strategy. Moreover, $V^{\pi^*}$ is the unique solution to the optimality equations that are given by

$$V(s) = \max_{i=1, \ldots, N} \left\{ \frac{\alpha_i + 1}{\alpha_i + \beta_i + 2} \left[ 1 + \gamma V(s + e_{2i-1}) \right] + \frac{\beta_i + 1}{\alpha_i + \beta_i + 2} \gamma V(s + e_{2i}) \right\},$$

where $V(s)$ denotes the optimal discounted reward starting from state $s \in S$ satisfying $s = (\alpha_1, \beta_1, \ldots, \alpha_N, \beta_N)$.

The optimality equation provides a full characterization of the website optimization problem. However, the description is rather abstract and not directly implementable in real websites. In the next sections, we provide more intuition into how the model works and how it can be implemented in real-time web servers.

### 10.2 Implementation

In the previous section we have cast the website optimization problem as a Markov decision problem. The idea behind the model is to store two numbers $\alpha_i$ and $\beta_i$ for version $i$ of the website. When a new visitor visits the website, the model determines based on the state $s = (\alpha_1, \beta_1, \ldots, \alpha_N, \beta_N)$ which version of the website to show. This decision is based on the values of $(\alpha_i + 1)/(\alpha_i + \beta_i + 2)[1 + \gamma V(s + e_{2i-1})] + (\beta_i + 1)/(\alpha_i + \beta_i + 2) \gamma V(s + e_{2i})$ for version $i$. Hence, to obtain this value, it is necessary to evaluate the unknown function $V$ in several states. The function $V$ can be obtained by solving the set of optimality equations. Since it is cumbersome to solve this set of equations online in the present form, there is a need for a more efficient procedure to obtain the desired values to make the optimal decision.

From a mathematical perspective, the optimality equations have a nice structure. The equations are so-called contraction mappings and due to the fixed-point theorem it has a unique solution [28]. Moreover, the fixed-point theorem also provides an efficient iterative algorithm to find the solution. Let $V_0(s) = 0$ for all $s \in S$, and define

$$V_{k+1}(s) = \max_{i=1, \ldots, N} \left\{ \frac{\alpha_i + 1}{\alpha_i + \beta_i + 2} \left[ 1 + \gamma V_k(s + e_{2i-1}) \right] + \frac{\beta_i + 1}{\alpha_i + \beta_i + 2} \gamma V_k(s + e_{2i}) \right\},$$
for all \( k = 0,1,\ldots \). Then the solution to the optimality equations can be obtained by recursively calculating \( V_k \) and is given by \( V = \lim_{k \to \infty} V_k \). However, (nearly) optimal policies can be obtained for relatively low values of \( k \). The following pseudo-code provides an algorithm, called value iteration, for deriving policies that differ \( \varepsilon \) from the optimal values.

1. Set \( V_0 \) to 0 for all states \( s \in S \), and let \( k = 1 \).
2. Calculate (by recursion) \( V_k \).
3. Calculate \( M = \max_s[V_k(s) - V_{k-1}(s)] \) and \( m = \min_s[V_k(s) - V_{k-1}(s)] \).
4. If \( M - m > \varepsilon \) then set \( k \) to \( k + 1 \) and continue with step 2.
5. The value of \( V_k \) is accurate enough to yield policies that are optimal within \( \varepsilon \) difference with respect to the optimal value.

This value iteration algorithm is a very efficient algorithm to solve the optimality equations. It has quadratic performance, i.e., the convergence of the algorithm to the solution is quadratic in the number of iterations. Hence, it can be done in real-time on web servers.

From a computer science perspective, the implementation of this algorithm requires little effort. One way to implement the ideas in this chapter would be to store every version of the website in its own subdirectory along with the \( \alpha_i \) and \( \beta_i \) in a separate file for version \( i \). In the top directory one could build a webpage that reads all the variables from the different subdirectories and invokes the value iteration algorithm. Note that the value iteration algorithm does not need to be run with \( s = (\alpha_1, \beta_1, \ldots, \alpha_N, \beta_N) \), the complete vector of parameters simultaneously. Instead, one could run the value iteration algorithm several times for pairs of websites so that gradually the best website emerges. Thus, define \( i_1 = 1 \), then one could start with \( s = (\alpha_{i_1}, \beta_{i_1}, \alpha_2, \beta_2) = (\alpha_1, \beta_1, \alpha_2, \beta_2) \) and denote the index of the best website by \( i_2 \). Then the algorithm can be run with \( s = (\alpha_{i_2}, \beta_{i_2}, \alpha_3, \beta_3) \) after which one denotes the index of the best website by \( i_3 \). Hence, in the \( k \)-th run, one would compare \( s = (\alpha_{i_k}, \beta_{i_k}, \alpha_{k+1}, \beta_{k+1}) \). After the \( N \)-th iteration, the value of \( i_N \) would point to the best website. In this way, the dimensionality of the problem is further reduced so that the algorithm becomes efficient and scalable for large numbers of websites (i.e., \( N \) large). Based on the outcome of the algorithm the webpage redirects the browser to the right subdirectory. In addition, each version of the website analyzes the user behavior and determines if the web visitor has successfully reached his goal. Based on this
information each version updates the values of the Beta distribution in its own subdirectory. Hence, every web visitor provides information that is used in determining which version to show when the next web visitor arrives.

The ideas of this chapter have been implemented and tested on a real web server. In the next section we illustrate the ideas based on experiments on this web server.

10.3 Website optimization: an illustration

In the previous sections we have outlined the mathematical algorithm and the implementation details. In this section we discuss the results of the implementation of the mathematical model for a real website. For illustrative purposes we will restrict ourselves to two versions of the website, so that the inner workings of the model become more clear. Figure 10.1 shows a website of the Joomla\textsuperscript{1} community. In this version of the website the navigation support is on the left side.

Figure 10.2 shows the same Joomla community website with a different design pattern. The difference with version 1 (Figure 10.1) is that the navigation support is on the right side in version 2 (Figure 10.2). To make the difference clearer we use a green background and green text color in version 2. On the top of both websites we see a link “About Joomla” (see red oval on the top in Figures 10.1 and 10.2). The information behind this link is about what requirements you need to run the Joomla CMS. Figures 10.3 and 10.4 show a screenshot of the “About Joomla” page. In the navigation support on the left and right side of version 1 and 2, respectively, we see a link with the name “More about Joomla” (see red oval on the left and right side in Figures 10.1 and 10.2). The information behind this link is about the Joomla project. Figures 10.5 and 10.6 show a screenshot of the “More about Joomla” page. Suppose we want to find out which version of the website is optimal to find the information about the Joomla project. Among the other links, the web visitor can click on the “About Joomla” link or on the “More about Joomla” link to look for that information. To optimize the website, and thus to find out which design pattern (navigation support on the left or on the right side) we should use, we have defined the link “More about Joomla” as a successful action (because the project information is be-

\textsuperscript{1}Joomla is an award-winning content management system, which enables you to build websites and powerful online applications. See also http://www.joomla.org.
Figure 10.1: Version 1, with navigation support on the left, of the Joomla website.

hind this link). Each time when a web visitor clicks on this link a success is stored in a text file. For illustration purposes we show the successes and failures below the header “MDP” in both versions (see Figures 10.5 and 10.6). Successes and failures are just two numbers that increase dependent on the success or failure of a web visitor. The mathematical model, which we have implemented, reads these numbers in the text file. Dependent on the successes and failures its shows dynamically the version of the website that maximizes the expected utility.

In our setting of the example, the state of the website is described by the vector \((\alpha_1, \beta_1, \alpha_2, \beta_2)\). We initialized this vector to \((0, 0, 0, 0)\) before web visitors could visit the website. Note that this choice of the parameter results in a uniform distribution for both \(\theta_1\) and \(\theta_2\), i.e., no information is available on the quality of the two versions of the website (see also Figure 10.7). When the vector \(s = (0, 0, 0, 0)\) is used to calculate the best version of the website to display, it turns out that both versions are optimal. This is not strange, since we have the same information on both versions. Hence, we choose...
Figure 10.2: Version 2, with navigation support on the right, of the Joomla website.

version 1 to display. Suppose that the web visitor that is presented with version 1 finds the right page and thus reaches the goal. Then the updated vector $s$ is then set to $s = (1, 0, 0, 0)$. The distributions are now different for both websites, see Figure 10.8. When this new vector $s$ is given to the algorithm, the optimal version to display turns out to be 1. This is not strange as version 1 is doing better than version 2. A rough estimate of $\theta_1 = 2/3$ and $\theta_2 = 1/2$. Suppose that the next web visitor also finds the webpage, resulting in $s = (2, 0, 0, 0)$. Then, obviously, by the same line of reasoning the algorithm will show version 1 again. The distributions now look as depicted in Figure 10.9. Suppose that the next web visitor is not able to find the webpage and thus results in a failure. Then the vector $s$ is updated to $s = (2, 1, 0, 0)$. Figure 10.10 shows that the distribution for $\theta_1$ is shifting to the left, so that lower utilities become more likely. However, version 1 still seems to be the better choice. Now suppose that the next web visitor also fails to find the webpage. This will result in $s = (2, 2, 0, 0)$. Thus, both version have an equal number of successes and failures. Therefore, a
Figure 10.3: The information behind the link “About Joomla” in version 1.

A rough estimate of the parameters would be $\theta_1 = 1/2$ and $\theta_2 = 1/2$. However, as Figure 10.11 shows, there is more uncertainty in the estimate of $\theta_2$. Based on the new vector $s$ the algorithm chooses to examine version 2. Note that the example above seems to imply that when $\alpha_1/(\alpha_1 + \beta_1) > \alpha_2/(\alpha_2 + \beta_2)$, thus when the estimate of $\theta_1$ is greater than the estimate of $\theta_2$, it is optimal to show version 1. However, this is not true, since in state $s = (3, 2, 0, 0)$ the estimate for $\theta_1 = 4/7 > \theta_2 = 1/2$, but the optimal version to display is 2. This displays that the algorithm takes the uncertainty into account of the other version.

The previous discussion tries to provide insight into how the algorithm works. In the initial phase, when the algorithm starts with $s = (0, 0, 0, 0)$, the algorithm explores the different versions to gain more information on the success rates of the versions. As the algorithm gains more information, the algorithm converges to one version, since the distinction between the two distribution is clearer. To illustrate this, Figure 10.12 shows the distribution for $s = (40, 10, 20, 30)$. Hence, the algorithm starts with an exploration phase in which it gathers information in a smart manner. Then, it moves on
to a phase in which it converges to the best version of the website. Finally, it has sufficient information and selects the best version of the website.

10.4 Website optimization for wild searching

In the previous section, we have shown how the mathematical model can be used to dynamically select the best version of a website based on the user click behavior. In the illustration, we assumed that we had a homogeneous group of web visitors, i.e., all web visitors had the same interest and user behavior leading to consistent estimates of what the best version of the website is. However, in practice, there are multiple heterogeneous groups of web visitors. Different groups might prefer different versions of the website and the model needs to account for this. In this section, we show how the model can deal with this situation so that the model can be valuable in practice.

In this section, we are faced with the challenge of optimizing websites for wild searching. In Chapter 7.3 we have discussed that two common
Figure 10.5: The information behind the link “More about Joomla” in version 1.

Figure 10.6: The information behind the link “More about Joomla” in version 2.
design patterns to support wild searching are tooltips and snapshots. These solutions are related to problem 1 from Chapter 6: “Web visitors do not get the expected information behind links or menu items”. Based on the web visitor’s domain knowledge, the tooltips or snapshots can be presented, e.g., in technical jargon for experts or in simple and elaborated language for laymen or non-experts.

To illustrate how a website with two different groups of web visitors can be optimized for wild searching we address in the next example a website (e.g., the Oxford Journals website\(^2\)) of which the main web visitors are scientists (e.g., web visitors from universities or scientific institutions). Next

\(^2\)http://www.oxfordjournals.org/.
to scientists, the website may be visited by people who have not in depth knowledge of the scientific topics presented on the website, but who are just interested in a particular topic.

Before we can subject the website to our mathematical model we have to distinguish the two groups of web visitors. This can be done by monitoring and comparing IP addresses. For example, a list with IP addresses of different universities and scientific institutions should be compared against the IP addresses of the web visitors. Note that the information can also be inferred from the network prefix of the IP address so that the list is reduced significantly in size. If a web visitor’s IP address matches an IP address from the list we can be sure that this web visitor is visiting the website through
a computer from a university or scientific institution\(^3\). In this way we can distinguish the two different groups of web visitors.

For both groups of visitors we run their own copy of the mathematical model. Thus, we have a set of model parameters for each group separately. Once we know to which group a web visitor belongs to, the model parameters for this group are retrieved. This information is fed into the mathematical model for optimization for this specific group. Then, we can present the web visitor a specific version of the website based on their model parameters. From here on the principle is the same as described in Section 10.3.

\(^3\)It is also possible to trace the name of the university or scientific institution from the IP address by requesting the hostname from a DNS server.
The above procedure leads to a system that learns the specific preferences of the two groups and presents a version of the website that matches these preferences. For example, let version 1 be the scientific version and version 2 the non-scientific version. Suppose that the model parameters of the universities and scientific institutions are given by \((\alpha_1, \beta_1) = (40, 10)\) and for the non-experts by \((\alpha_2, \beta_2) = (10, 30)\) after 90 visits (50 by experts and 40 by non-experts). This means that web visitors belonging to the scientific group (experts) are presented a website with tooltips or snapshots in technical jargon. Web visitors belonging to the non-expert group are presented a website with tooltips or snapshots in simple and elaborated language. Based on the web visitor’s behavior, the model observes either a success or a failure and will learn the best version for each group. Note that it could well be that there is little difference between the groups (e.g., this could happen if the model parameters for the non-expert group were \((\alpha_2, \beta_2) = (30, 10)\)). However, this is not a problem, since the model adapts to the behavior of the groups. The web visitor’s behavior is the basis on which the model will dynamically determine whether it will present a version with tooltip or a version with snapshot for each of the two groups. The process of website optimization for wild searching for this example is illustrated in Figure 10.13.

---

**Figure 10.13:** The process of website optimization for wild searching for two types of groups of web visitors.
In general, one could create a database of web visitors and store unique identifiers to identify the visitors. Based on available information, e.g., IP addresses translating to country, internet providers, and universities, one can cluster the identifiers into homogeneous groups of which the web visitors are expected to behave nearly homogeneously. When a user visits the website, then two things can happen: either his identifier is stored in the database, or he is classified as a new web visitor. If the identifier is in the database, then his parameters $\alpha$ and $\beta$ for the prior distribution can be retrieved. For every new web visitor, one could try to match the groups to the characteristics of the new visitor. If such a group is found, the parameters of this group can be used for the prior distribution of the new visitor. If such a group is not found, then the overall behavior of the web visitors in the database can be used for determining the parameters of the prior distribution. The identifier for the new web visitor is then added to the database along with his determined parameters. In this way, a revisit to the website customizes the appearance tailored to the behavior of each individual web visitor, but the initial parameters on which the algorithm works are determined by the group behavior.

10.5 Numerical experiments

In this subsection, we evaluate the performance of our algorithm for dynamic website optimization by means of numerical experiments. For this purpose, we assume that we have a homogeneous group of web visitors and two versions of a website only. These assumptions enable us to compare our algorithm to A/B testing, which is a leading method for dynamic website optimization in which only two versions are compared. Note that this is a special case of multivariate testing in which multiple versions with overlapping changes are compared. However, as described in the previous chapter, many website optimization algorithms in practice use A/B testing (in modified form) to assess the best version in a multivariate setting as well. Hence, it is insightful to benchmark our dynamic website optimization algorithm against A/B testing.

Our experimental setup consists of two versions of a website. In practice, the behavior of our homogeneous group of web visitors determines the probabilities $\theta_1$ and $\theta_2$ for the two versions. However, since a large-scale experimental setup with web visitors is not feasible, we simulate the behavior
of web visitors on a computer. Thus, before starting the experiment, we fix the values of $\theta_1$ and $\theta_2$. We hide these parameters from our algorithm, so that the algorithm still has to make decisions without this knowledge. We then generate web visitors. For every web visitor, we run the algorithm and determine the version of the website that has to be displayed. If the algorithm decides to show version $i$, then the visit of the web visitor to version $i$ results in a success with probability $\theta_i$ and in a failure with probability $1 - \theta_i$ for $i = 1, 2$. This information, i.e., if the version generated a success or a failure, is fed back into the algorithm for better decision making. This process continues for all the web visitors. In our experiments we generate 100 web visitors. Since this simulation is just one realization of a stochastic process, we need to redo this several times to collect reliable statistics on the number of times that the version with the higher probability was chosen, the number of successes obtained, and the estimates of the parameters. In our experiments, we redo each experiment 100,000 times.

The A/B testing method has been set up in a similar fashion. We generate 100 web visitors. In practice, most A/B testing methods are conducted in a static fashion, i.e., a dashboard displays the performance of the previous visitors and based on those statistics one can choose the best version. In order to have a fair comparison, we implement a dynamic A/B testing method. In this method, after every web visit, the method looks at the estimate $\hat{\theta}_i$ for $\theta_i$, which is the number of successes for version $i$ divided by number of times version $i$ has been shown. The next web visitor is then presented version 1 if $\hat{\theta}_1 \geq \hat{\theta}_2$, and version 2 otherwise (note that this algorithm corresponds to our algorithm in which no learning takes place, i.e., $\gamma = 0$). We collect the same statistics on the number of times that the version with the higher probability was chosen, the number of successes obtained, and the estimates of the parameters by repeating the experiment 100,000 times.

In our first experimental setup, we choose $\theta_1 = 0.3$ and $\theta_2 = 0.7$. These parameter settings correspond with two versions of which one is obviously not as well evaluated by the web visitors than the other. Hence, any website optimization algorithm should find that version 2 is the better version, and should display that version as many times as possible. The A/B testing algorithm starts with $\hat{\theta}_1 = \hat{\theta}_2 = 0.5$. These probabilities are then updated as described before. Our dynamic website optimization algorithm starts in state $s = (0,0,0,0)$ and uses the policy generated in the previous section. However, we need to specify how much exploration is done by the algorithm to learn more about the uncertainty in the estimates. This exploration is
Numerical experiments

Figure 10.14: The number of times the best version is displayed, $\theta = (0.3, 0.7)$.

represented by the parameter $\gamma$. We first start the algorithm with a little exploration by setting $\gamma = 0.1$.

Figure 10.14 shows a histogram in which the number of times that the best version (i.e., version 2) is displayed both under A/B testing and the algorithm for dynamic WSO. The results show that A/B testing has quite a good performance with a mean equal to 90.6. However, the graph also shows that the A/B testing method goes wrong sometimes by pure chance and results in cases in which almost never the best version is selected. This is the root cause for a variance equal to 492.7. The algorithm for dynamic WSO has a mean of 84.1. This seems to be worse than under A/B testing. Note that this is to be expected, because the algorithm tries to learn a bit more about version 1 to assess that it is really not the best version. Since the difference between $\theta_1$ and $\theta_2$ is rather big, a greedy approach suffices here and in fact no exploration is necessary. However, in light of the variance, which is equal to 49.5, the algorithm does give a guaranteed good performance. Hence, the performance of the algorithm for dynamic WSO is quite well as compared to the A/B testing method.

In Figure 10.14 we compared the number of times the best version was shown. However, for most websites the conversion rates are of importance. Hence, one is interested in the number of successes that have occurred both
under the presentation of the good and bad versions. Therefore, in Figure 10.15 we study the number of times a success has been observed both under the presentation of version 1 and version 2. The mean number of successes under A/B testing is equal to 66.2. However, as we saw in the previous graph, not always the best version is chosen. This results in a variance of 94.5. The algorithm for dynamic WSO has on average 63.7 successes that are obtained. This is not a big difference as compared to the mean number of successes under A/B testing. However, the variance in this result is much lower and is equal to 24.1. Hence, this graph again shows that the algorithm for dynamic WSO has a more consistent performance and guarantees good performance.

Finally, we turn our attention to the estimates of $\theta_1$ and $\theta_2$. In Figure 10.16 we show the graphs of the estimates under the A/B testing and the algorithm for dynamic WSO. The A/B testing method has a mean estimate of $\hat{\theta}_1 = 0.33$ and $\hat{\theta}_2 = 0.67$. The algorithm for dynamic WSO has a mean estimate of $\hat{\theta}_1 = 0.28$ and $\hat{\theta}_2 = 0.70$. The variance of the estimate of $\theta_1$ under the A/B testing method is smaller than under the dynamic WSO method with 0.0035 and 0.014, respectively. However, the variance of the estimate of $\theta_2$ is larger for the A/B testing method with 0.011 as compared to the variance of 0.0025 under the dynamic WSO method. This is to be expected when the previous graphs are studied. The algorithm for dynamic
WSO selects the best version more consistently than under the A/B testing method. Hence, the estimate of $\theta_2$ is more accurate under the dynamic WSO method.

Our first experimental setup shows that under the choice $\theta_1 = 0.3$ and $\theta_2 = 0.7$ a greedy algorithm slightly outperforms the algorithm for dynamic WSO. This is, as mentioned already, not that surprising, because no exploration is needed to make a well-informed decision on the best version of the website. However, the greedy algorithm does not have consistent performance, as the fluctuations in the results are rather larger. Hence, when both the performance and the consistency thereof are taken into account, then the algorithm for dynamic WSO is preferred over the greedy algorithm. In our next experimental setup, we take $\theta_1 = 0.6$ and $\theta_2 = 0.7$ and compare both algorithms. In this case, the two versions are somewhat more similar, and it is harder to distinguish which version is the best. We again compare both algorithms on the points discussed so far.

Figure 10.17 shows the number of times that the best version has been shown to the web visitor. The A/B testing method has shown 46.5 times the best version with a variance of 2289.0. The algorithm for dynamic WSO has shown the best version 72.7 times with a variance of 149.8. Hence, the algorithm for dynamic WSO clearly outperforms the A/B testing method here. Apparently, since $\theta_1$ and $\theta_2$ are so close to each other, the A/B testing method has a hard time distinguishing the best version of the website.

In Figure 10.18 we can see the results of the experiments with respect to the number of successes obtained under both algorithms. On initial sight, the two graphs look similar, however, there are a few differences to be observed. First, the A/B testing method has a mean number of successes equal to 64.6 with a variance of 39.4. The algorithm for dynamic WSO has a mean number of successes that is equal to 67.3 with a variance of 22.9. The algorithm for dynamic WSO is thus better in distinguishing which version is better, and as a result also obtains more successes. Again, we can observe that the algorithm for dynamic WSO has a consistent performance and has a good performance with higher guarantees.

Figure 10.19 depicts the estimates for $\theta_1$ and $\theta_2$ under both methods. The A/B testing method has estimates $\hat{\theta}_1 = 0.49$ and $\hat{\theta}_2 = 0.57$ with variances of 0.018 and 0.021, respectively. The dynamic WSO method has estimates $\hat{\theta}_1 = 0.59$ and $\hat{\theta}_2 = 0.70$ with variances 0.013 and 0.0030, respectively. It is clear that the algorithm for dynamic WSO has estimates the probabilities $\theta_1$ and $\theta_2$ better than the A/B testing method and with higher accuracy as well.
Hence, we can conclude that the algorithm for dynamic WSO outperforms the A/B testing method.

Extensive numerical experiments for different values of $\theta_1$ and $\theta_2$ consistently show that the A/B testing method provides results with a higher variance and with less precise estimates. The algorithm for dynamic WSO provides good performance over the broad range of parameters values of $\theta_1$ and $\theta_2$, and thus outperforms the A/B testing method. The previous experiments were all conducted under the parameter setting $\gamma = 0.1$. When $\gamma$ increases, the algorithm for dynamic WSO explores the different versions a bit more than for lower values of $\gamma$. We have also studied the performance under different values of $\gamma$, and the results suggest that the best value for $\gamma$ is found for $\gamma$ within the interval 0.1 and 0.2. There does not seem to be significant differences in the results obtained for the values within this interval. Hence, $\gamma = 0.1$ is a good choice to run the algorithm for dynamic WSO.

10.6 Conclusion

In this chapter we have dealt with the problem of autonomous management of design patterns for website optimization. Websites can significantly improve through the right selection of design patterns. However, the actual implementation of design patterns is based on a lot of choices, e.g., color, position, and style. Moreover, the combination of different design patterns might enforce each other or might have adverse effects that are not known beforehand. Therefore, there is a need to automatically select the right version of the website out of a set of versions with different implementations of the design patterns based on automatic analysis of user behavior.

We have developed a mathematical model that learns which design patterns lead to an optimized website through the click behavior of web visitors. In this setting, the model typically needs to balance the exploration problem (learning) versus the exploitation problem (optimal selection) in a dynamic setting. The learning algorithm is modeled by a Bayesian algorithm, and the optimal selection is modeled by a Markov decision problem. We combined the two parts leading to a Bayesian Markov decision model. The resulting model, however, has a high-dimensional state space prohibiting both tractable analytical and numerical solutions to the problem. We resolve this issue through the reduction of the dimensionality of the state space by de-
developing sufficient statistics. This sufficient statistic is essential to derive a computationally efficient algorithm for website optimization. We have discussed the implementation details of the model that scales well with the number of design patterns and the size of the website. Moreover, we have shown a proof of concept on a real web server. Experiments on the web server show that the model has very good performance and automatically selects the website version with the highest expected utility. We have described how the mathematical model can be applied to optimize websites for wild searching.
Figure 10.16: Estimates of $\theta_1$ and $\theta_2$, $\theta = (0.3, 0.7)$. 
Figure 10.17: The number of times the best version is displayed, $\theta = (0.6, 0.7)$.

Figure 10.18: The number of times a success is obtained, $\theta = (0.6, 0.7)$.
Figure 10.19: Estimates of $\theta_1$ and $\theta_2$, $\theta = (0.6, 0.7)$. 
Chapter 11

Summary

Websites have become an essential part of every profession, making a strong impact on the image of every company. Therefore, it is of high importance that websites are designed such that users can find the information they seek easily. However, good design and management of websites is not straightforward as websites are growing in size and becoming more and more complex due to dynamic content. The number of web pages and the complexity make websites very difficult to manage and to maintain. Therefore, optimizing websites in a static and non-autonomous way is a time-consuming and laborious process, which is often not feasible in practice. In this thesis, we bridge this gap by developing a model for dynamic website optimization through autonomous management of design patterns.

11.1 Search behavior & design patterns

The World Wide Web (WWW) is growing with millions of web pages every day. It has an enormous economic value through e-commerce. But, before a website becomes economically relevant, the website has to be found first and needs to have a navigation structure such that users can find relevant information easily. Website optimization (WSO) is a new research area and is a process of improving internal (e.g., layout of web pages and content) and external (e.g., promotion of the website and link building) aspects of web pages to increase the traffic the website receives from search engines. Findability heavily depends on how people search for information. Therefore, we first observed people in “non-web” situations (e.g., shopping malls and
city centers) to find out how people search in “non-web” situations and how their goals are influenced (see Chapter 4). Our main observation was that participants deviated from their initial goals because of parallel goals and because of state triggers that influenced their interests.

The WWW is a relatively new context for people to search in. We did not know whether the search behavior on the WWW would be the same as the search behavior in “non-web” situations. Therefore, we observed the search behavior of people on the WWW (see Chapter 5). We observed that web design problems can cause website visitors to adapt their search behavior. We also observed similarities between searching in “non-web” settings and searching on the WWW. To improve searching on the WWW, the WWW should support all search behaviors (e.g., wild searching) that are accepted in our daily life outside the WWW. To make this possible we formulated design patterns for static website optimization. We also formulated a design pattern for wild searching as wild searching is not supported very well on the WWW.

11.2 Dynamic website optimization

Many design patterns are suggested by experts for common design problems [231, 4, 248, 225, 2]. In this thesis we have defined some new design patterns and discussed the relevance of those design patterns. It is very hard to decide which design pattern optimizes your website the best as there are many design patterns. A combination of different design patterns might enforce each other to improve the website or might have adverse effects that are not known beforehand. Moreover, the design patterns (or sub patterns) can also be used in different styles leaving room for a lot of freedom in usage. It is often not feasible to try all the combinations of the design patterns in a static and non-autonomous way to optimize websites. This would be a time-consuming and laborious process. Therefore, we have developed a mathematical model to optimize websites dynamically (as opposed to Google Website Optimizer) through autonomous management of design patterns. We discussed and implemented the model (see Chapter 10) and built a proof of concept on a real webserver. Our model automatically selects the right website version out of a set of versions with different implementations of the design patterns based on automatic analysis of web visitor behavior (see Section 10.3 and 10.4 for examples). Experiments demon-
strated that the model has very good performance and automatically selects the optimized website. With this model we contribute to the WSO area as there is little to none dynamic and autonomous ways to optimize websites until now.

11.3 Future directions

Our adaptive and self-learning model optimizes websites on the basis of the search behavior of web visitors. We know that websites generate a lot of data (e.g., which browser the web visitor is using, what webpage the visitor previously visited, the path that the visitor followed on the website, what links the visitor clicked on, how long the visitor spent on a webpage, how many times the visitor visited the website, what operating system the web visitor is using, what is the web visitor’s IP address). Website owners use these data to build profiles for the different web visitors and to show them content based on this profile (see e.g., http://www.amazon.com). Based on these profiles it is possible to personalize websites to match the interests of the web visitor. Hence, by extending the model with more information, the model is able to optimize websites based on a population of web visitors in which it differentiates based on profiles. We described such an example in Section 10.4 to illustrate how a website could be optimized for two different types of groups of web visitors who have different domain knowledge.
Samenvatting

Dynamische website optimalisatie door middel van autonoom beheer van design patterns

Websites zijn een essentieel onderdeel geworden van elke bedrijfstak en laten een sterke indruk achter van het bedrijfsbeeld. Het is daarom van groot belang dat websites goed ontworpen zijn zodat gebruikers de informatie makkelijk kunnen vinden. Echter, goed ontwerp en beheer van websites wordt bemoeilijkt door de groei in omvang van websites en door de toenemende complexiteit ten gevolge van dynamische inhoud. Het aantal webpagina’s en de complexiteit maken het erg moeilijk om websites te beheren en te onderhouden. Daarom is het optimaliseren van websites op een statische en niet-autonome wijze een tijdrovende en moeizaam proces, dat vaak niet haalbaar is in de praktijk. In dit proefschrift richten we ons op het ontwikkelen van een model voor de uitvoering van dynamische website optimalisatie door middel van autonoom beheer van design patterns.

Zoekgedrag & design patterns

Het World Wide Web (WWW) groeit dagelijks met miljoenen webpagina’s. Het heeft een enorme economische waarde door middel van e-commerce. Echter, voordat een website economisch relevant wordt, moet de website eerst gevonden worden. De website moet tevens een zodanige navigatiestructuur hebben dat gebruikers de relevante informatie gemakkelijk kunnen vinden. Website optimalisatie (WSO) is een nieuw onderzoeksgebied. WSO is een proces om de interne (bijvoorbeeld lay-out van webpagina’s en inhoud) en externe (bijvoorbeeld het promoten van de website en link building) aspecten van webpagina’s te verbeteren, zodanig dat het aantal bezoekers
dat via zoekmachines de website bezoekt, verhoogd wordt. Vindbaarheid is sterk afhankelijk van de manier waarop mensen zoeken naar informatie. We hebben daarom eerst mensen geobserveerd in “niet-web” situaties (bijvoorbeeld winkelcentra en stadscentra) om uit te zoeken hoe mensen zoeken in “niet-web” situaties en hoe hun doelen beïnvloed worden (zie hoofdstuk 4). Onze belangrijkste observatie was dat de deelnemers afwijken van hun oorspronkelijke doelen vanwege parallelle doelen en wegens state-triggers die hun interesses beïnvloeden.

Het WWW is een relatief nieuw medium voor mensen om erop te zoeken. Het is bij voorbaat niet duidelijk of het zoekgedrag op het WWW hetzelfde is als het zoekgedrag in “niet-web” situaties. Daarom hebben we het zoekgedrag van mensen op het WWW geobserveerd (zie hoofdstuk 5). We namen waar dat webbezoekers hun zoekgedrag aanpasten door problemen in de web design. We hebben ook overeenkomsten waargenomen tussen het zoeken in “niet-web” situaties en zoeken op het WWW. Om het zoeken op het WWW te verbeteren, zou het WWW alle zoekgedragingen (zoals wild searching) moeten ondersteunen, die in onze dagelijkse leven zijn geaccepteerd buiten het WWW om. Om dit mogelijk te maken hebben wij design patterns geformuleerd voor statische website optimalisatie. Tevens hebben wij een design pattern geformuleerd voor wild searching, omdat wild searching nog niet goed ondersteund wordt op het WWW.

**Dynamische website optimalisatie**

Er zijn veel design patterns voor algemene design problemen voorgesteld door tal van deskundigen [231, 4, 248, 225, 2]. In dit proefschrift hebben wij enkele nieuwe design patterns gedefinieerd en de relevantie ervan besproken. Het is moeilijk te bepalen welke design pattern je website het beste optimaliseert omdat er zoveel design patterns zijn. Een combinatie van verschillende design patterns kan elkaars werking versterken om de website te verbeteren. Het zou ook nadelige effecten kunnen hebben, die niet op voorhand bekend zijn. Daarnaast kunnen de design patterns (of sub patterns) ook nog eens in verschillende visuele stijlen gebruikt worden waardoor het aantal mogelijke manieren in gebruik erg veel vrijheid toelaat. Het is vaak niet haalbaar om alle combinaties van de design patterns op een statische en niet-autonome manier uit te proberen om websites te optimaliseren. Dit zou een tijdrovend en moeizaam proces worden. We hebben daarom een
wiskundig model ontwikkeld om websites dynamisch (in tegenstelling tot de Google Website Optimizer) te optimaliseren door autonoom beheer van design patterns. We hebben dit model besproken (zie hoofdstuk 10) en een proof of concept gebouwd op een echte webserver. Ons model selecteert, op basis van automatische analyse van het gedrag van de website bezoeker, automatisch de juiste website versie uit een set van versies met verschillende implementaties van de design patterns (zie hoofdstuk 10.3 en 10.4 voor voorbeelden). Experimenten tonen aan dat het model goed presteert en dat het automatisch de geoptimaliseerde website selecteert. Met dit model leveren we een bijdrage aan het WSO onderzoeksgebied, omdat er tot nu toe nog weinig tot geen dynamische en autonome manier was om websites te optimaliseren.
Appendices
Appendix A

Interview questions

A.1 Interview questions for group N in ‘non-web’ situations

Below you will find the interview questions we used for group N (ethnography) in “non-web” situations described in Chapter 4.

1. How familiar are you with this location?

   Total unfamiliar  1  2  3  4  5  Very familiar

2. Did you reach your goal?

   (a) Yes. Why do you think that you have reached your goal?
   (b) No. Why did you not reach your goal?

3. Is the goal (purchase) for yourself or for someone else?

4. Did you deviate from or adapt your goal?

   (a) Yes. Why you deviated from or adapted your goal?
       (e.g., background noise, display influence, etc.)
   (b) No.

5. Did you gain information on your goal somewhere else?
   (e.g., inquiry in other shops, friends, etc.)

   (a) Yes. Where?
(b) No.

6. Which resources did you use to reach your goal? 
   (e.g., boards, maps, consulting personal, etc.)

7. To which age group do you belong? 
   
   Age group:  16-20  21-25  26-30  31-35  36-40
   41-45  46-50  51-55  55+

8. Where are you from?

A.2 Interview questions for group T in ‘non-web’ situations

Below you will find the interview questions we used for group T (task based observation) in “non-web” situations described in Chapter 4.

1. How familiar are you with this location?

   Total unfamiliar  1  2  3  4  5  Very familiar

2. How familiar were you with the goal?

   Total unfamiliar  1  2  3  4  5  Very familiar

3. Did you reach your goal easily?

   (a) Yes. Why?
   (b) No. Why?

4. Which resources did you use to reach your goal? 
   (e.g., boards, maps, consulting personal, etc.)

5. What is your highest education level?

6. What is/was your occupation?

7. To which age group do you belong?

   Age group:  16-20  21-25  26-30  31-35  36-40
   41-45  46-50  51-55  55+

8. Where are you from?
A.3 Interview questions for situation N on the WWW before observation

Below you will find the interview questions we used for situation N (ethnography) on the WWW described in Chapter 5. The interview was held before the observation.

1. How would you classify yourself as computer user?  
   (novice to expert on a scale from 1 to 5)

2. How would you classify yourself as internet user?  
   (novice to expert on a scale from 1 to 5)

3. What are you going to do (what is your goal)?

4. What is/was your occupation?

5. What is your highest education level?

6. What is your start page/homepage in your browser?

7. Do you use search engines?  
   (a) Yes. Which one?  
   (b) No.

8. Do you ever click on banners?  
   (a) Yes. Why?  
   (b) No.
A.4 Interview questions for situation N on the WWW after observation

The following interview questions we used for situation N (ethnography) on the WWW described in Chapter 5. The interview was held after the observation.

1. Did you reach your goal?
   (a) Yes. Why do you think that you have reached your goal?
   (b) No. Why did you not reach your goal?

2. Are you satisfied with the result(s)?
   (a) Yes.
   (b) No. Why?

3. Did you deviate from or adapt your goal?
   (a) Yes. Why you deviated from or adapted your goal?
   (b) No.

4. Did you experience any problems in your search?
   (a) Yes. Why?
   (b) No.

5. Do you like a website with dark background color and light font color or vice versa and why?

6. Do you like an introduction animation on a website?
   (a) Yes. Why?
   (b) No. Why?

7. Do you prefer many text or images on websites and why?

8. To which age group do you belong?
   Age group: 16-20 21-25 26-30 31-35 36-40 41-45 46-50 51-55 55+
A.5 Interview questions for situation T on the WWW

The following interview questions we used for situation T (task-based observation) on the WWW (see Chapter 5). The interview was held after the observation.

1. How familiar were you with the goals?

   Total unfamiliar  1  2  3  4  5  Very familiar

2. Did you reach your goal easily?

   (a) Yes. Why?
   (b) No. Why?

3. Did you searched differently than usual or is this the approach you always use on the Internet when you are searching for things?

4. Where should the navigation support appear on a webpage (right, left, top, or bottom) and why?

5. Did you experience any problems in your search?

   (a) Yes. Why?
   (b) No.
Appendix B

Forms

B.1 Observation form for group N and T

The following form we have developed and used to observe the search behavior of participants in ‘non-web’ situations in Chapter 4.
Figure B.1: Observation form for search behavior in ‘non-web’ situations.
B.2 Observation form for situation N and T (WWW)

The following form we have developed and used to observe the search behavior of participants on the WWW in Chapter 5.
Figure B.2: Observation form for search behavior on the WWW.
Appendix C

Text used to inform participants

In this appendix you will find the translated text which is used to inform the participants about the study. The participants were informed orally.

1. The goal of this study is to find out how people are searching and how their goals are influenced. This study is a part of a major research project which investigates how to optimize websites.

2. During the observation I will not answer any questions. You can ask questions after the observation session. An observation session ends after 30 minutes or when you make a purchase or complete a task.

3. After the observation I will take a short interview. The interview will take 5 to 10 minutes.

4. We do not record any names or personal data in this study. The observation and interview data will be linked to a number. In this way your privacy will be guaranteed. Everything that I observe and that I will discuss with you in the interview will be treat confidentially.

5. Do you have any questions right now?
Text used to inform participants
Appendix D

Related studies

In this appendix we give a summary of some projects which were done by students [160, 217] and which were supervised by the author. We mention these projects here because they are related to our research.

D.1 User’s search and navigation behavior in non-profit domains

D.1.1 The study

We were approached by a consultancy company Cemit\(^1\) that focuses on optimization of websites to do this study. The study investigated the users’ search and navigation behavior in non-profit websites in the Netherlands and the U.S.A. This company advises organizations in making their website more effective, which is done among other things with the so called Total Review\(^\text{TM}\). This method was most suitable for e-commerce websites and needed to be made applicable for non-profit organizations. The study tried also to assess how successful the websites are in facilitating users in reaching their goals, in order to reach the non-profits organizational goals. The study focused on two non-profit sectors: local government and charities.

\(^1\)Cemit is an abbreviation for Customer Experience Measurement Improvement Techniques. [http://www.cemit.nl](http://www.cemit.nl).
D.1.2 Approach

For local government we chose Dutch provinces because there are a manageable number of provinces (12) in the Netherlands. All these provinces have their own website, all with the same goals. One of the guidelines of the government was that by the end of 2005 45% of the services of the Dutch provinces needed to be available on-line. This percentage needed to be 65% in 2007 (see also [133]). Province websites are visited and used mainly with the purpose of utilizing public administrative services.

Charity websites are different: visitors voluntarily approach a website mainly by motivations related to the domain of interest of the charity. They are supposed to find information about the charity’s goal and are often stimulated to make a donation. Some of these organizations have a long history and business is done (like with provinces) on-line as well as off-line. Other charity organizations are pure on-line organizations.

Two methods were used to evaluate and review websites.

1. The consultancy company that commissioned the study has developed the so-called Total Review™ method. This method was originally designed for e-commerce websites and needed to be adapted for non-profit organizations.

2. We also applied a user questionnaire that was developed for this study.

The Total Review™ method applies information collected from the sites by a usability expert who answers standard questions that are chosen from a database. The user questionnaire asks actual users of a website to perform certain tasks and to answer questions in relation to the interaction.

Selection and identification of the sites

The official websites of the Dutch provinces were identified. In most cases other websites existed that were related in different ways to the local provincial government. Those websites had different goals, like acquisition of tourism or of industrial enterprises. The official government websites can be found in Table D.1.

The selection of the charities websites was based on company’s commission for our project: the company supports Net4Kids and asked us to assess their website. We choose ten other non-profit organizations to compare with Net4Kids, based on our knowledge of their reputation and because they all
User’s search and navigation behavior in non-profit domains

Table D.1: URLs of the official Dutch province websites.

<table>
<thead>
<tr>
<th>Province</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drenthe</td>
<td><a href="http://www.drenthe.nl">www.drenthe.nl</a></td>
</tr>
<tr>
<td>Flevoland</td>
<td><a href="http://www.flevoland.nl">www.flevoland.nl</a></td>
</tr>
<tr>
<td>Friesland</td>
<td><a href="http://www.fryslan.nl">www.fryslan.nl</a></td>
</tr>
<tr>
<td>Gelderland</td>
<td><a href="http://www.gelderland.nl">www.gelderland.nl</a></td>
</tr>
<tr>
<td>Groningen</td>
<td><a href="http://www.provinciegroningen.nl">www.provinciegroningen.nl</a></td>
</tr>
<tr>
<td>Limburg</td>
<td><a href="http://www.limburg.nl">www.limburg.nl</a></td>
</tr>
<tr>
<td>Noord-Brabant</td>
<td><a href="http://www.brabant.nl">www.brabant.nl</a></td>
</tr>
<tr>
<td>Noord-Holland</td>
<td><a href="http://www.noord-holland.nl">www.noord-holland.nl</a></td>
</tr>
<tr>
<td>Overijssel</td>
<td><a href="http://www.prv-overijssel.nl">www.prv-overijssel.nl</a></td>
</tr>
<tr>
<td>Utrecht</td>
<td><a href="http://www.provincie-utrecht.nl">www.provincie-utrecht.nl</a></td>
</tr>
<tr>
<td>Zeeland</td>
<td><a href="http://www.zeeland.nl">www.zeeland.nl</a></td>
</tr>
<tr>
<td>Zuid-Holland</td>
<td><a href="http://www.zuid-holland.nl">www.zuid-holland.nl</a></td>
</tr>
</tbody>
</table>

Table D.2: URLs of charities websites.

<table>
<thead>
<tr>
<th>Organization</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Net4Kids</td>
<td><a href="http://www.net4kids.org">www.net4kids.org</a></td>
</tr>
<tr>
<td>Children international</td>
<td><a href="http://www.children.org">www.children.org</a></td>
</tr>
<tr>
<td>Global Giving</td>
<td><a href="http://www.globalgiving.com">www.globalgiving.com</a></td>
</tr>
<tr>
<td>Habitat</td>
<td><a href="http://www.habitat.nl">www.habitat.nl</a></td>
</tr>
<tr>
<td>Heifer</td>
<td><a href="http://www.heifer.nl">www.heifer.nl</a></td>
</tr>
<tr>
<td>KWF</td>
<td><a href="http://www.kwfkankerbestrijding.nl">www.kwfkankerbestrijding.nl</a></td>
</tr>
<tr>
<td>Net Aid</td>
<td><a href="http://www.netaid.org">www.netaid.org</a></td>
</tr>
<tr>
<td>Plan Nederland</td>
<td><a href="http://www.plannederland.nl">www.plannederland.nl</a></td>
</tr>
<tr>
<td>Unicef</td>
<td><a href="http://www.unicef.nl">www.unicef.nl</a></td>
</tr>
<tr>
<td>Warchild</td>
<td><a href="http://www.warchild.nl">www.warchild.nl</a></td>
</tr>
<tr>
<td>WNF</td>
<td><a href="http://www.wnf.nl">www.wnf.nl</a></td>
</tr>
</tbody>
</table>

have projects people can donate for. All charity websites in our study do their business on-line, so they need a very effective website. The charity organizations and their associated websites are provided in Table D.2.
The evaluation and review methods

Total Review™ method
The so called Total Review™ method is used by the consultancy company to evaluate and review websites. The Total Review™ method is based on a database with questions about so called main categories like usability, quality of information, functionality, thoroughness, charisma, branding, trust, content, etcetera. The questions were composed by Cemit, based on well known guidelines like described in [168]. The Total Review™ method applies effectiveness criteria, categories, questions and weighing factors. Categories belong to one or more effectiveness criteria and have weighing factors for each effectiveness criterion they contribute to. Questions are assigned to categories and again have a weighting factor. The average of the scores of the effectiveness criteria is considered to measure effectiveness of the website. For details on the Total Review™ method, see [1].

The Total Review™ method as we used it was adapted for non-profit organizations. In order to make the Total Review™ method applicable for non-profit websites like the provincial and charity websites, many questions specifically about e-commerce and questions around selling products were deleted from the database.

For the research about the Dutch provinces the criteria were based on the strategic goals for the websites of the Dutch provinces. The guidelines and requirements [133, 183, 184] the government has officially declared for websites of Dutch provinces were used for this study. Based on this, questions were added about e-government and participation. Questions about e-government were not relevant for the charity organizations. For both the charity organizations and the Dutch provinces questions were added about participation. On the basis of these changes effectiveness criteria resulted that are described in Table D.3.

User test
The second method was developed to empirically evaluate and review Dutch province websites. It consists of a User Test in the form of a questionnaire. This questionnaire could be answered at each user’s convenience. The User Test was intended to complement the results of the Total Review™ method. The original User Test is in Dutch, because all intended participants would be living in the Netherlands and Dutch websites intended for Dutch users had to be evaluated. The User Test consists of five assignments. These questions
had to be answered twice, for two different provinces. The investigation ends with a question about which of the two websites is the best according to the participants. The participants were asked to record the time needed for each assignment. To this end users were asked to use a stopwatch or other time measuring device. This timer should be started on arrival at the homepage of the visited website and stopped as soon as the information was found that was needed for the assignment. Participants were allowed to stop the task if the information could not be found within three minutes. This limit was chosen to prevent participants from dropping out in case they might not be able to accomplish some of the assignments. Writing down the answers to the questions was not timed.

### User Test procedure

The User Test was only applied to the websites of the Dutch provinces and presented to internet users known by at least one of the authors. All were approached through email and, consequently all were known internet users. We wanted to compare each province with each other and we did not want the user to visit more than two provinces. So we needed 132 participants to compare a unique pair of Dutch provinces ($12 \times 11 = 132$). Of the 132 participants, 62 were either university students in a faculty of science, or ICT professionals (so-called ‘heavy’ users), and 70 participants were known to use internet only for domestic and leisure purposes (so-called ‘light’ users).

For two different Dutch provinces, these participants were asked to com-
plete five assignments. These Dutch provinces were randomly assigned to each participant.

The participants received a personal email with a MS Word document that contained both series of assignments and questions. They were asked to return it as soon as possible. The content of the returned document was put in a database in order to calculate the statistics we needed.

User Test score

The questions in the User Test were usually answered with ‘Yes’ and ‘No’. ‘Yes’ was coded 1 and ‘No’ was coded 0. In some cases users could also answer ‘I do not know’, which was scored 0.5. The aggregated answers to the User Test were expressed in percentages. 100% means that all users were able to find the specific information or to complete an assignment. By counting how many times each Dutch province was elected as “best Dutch province”, divided by the times a Dutch province could be nominated, the preference score for each Dutch province was calculated.

D.1.3 Description of the assignments

1. Assignment 1: find the URL of province X.
   We asked the participants to indicate whether they knew they were at the official website of the Dutch province or not. Next, the participants were asked to write down the URL of the website. This was used to check which website the participants visited.

2. Assignment 2: find the physical address and telephone number of province X.
   The participant had to indicate if he could find the address and telephone number, and how many seconds it took him to find that information.

3. Assignment 3: find the risk map of province X.
   The participant had to go to the homepage and find the webpage with information about the risk map. Then the participant had to find out how the risk map could be consulted. If the risk map was on-line, the participant could stop with the assignment and write down the seconds needed to find the webpage. If the risk map was not on-line available,
the question for the participant was if it was clear for him how he could consult the risk map off-line.

4. Assignment 4: find a possibility to submit a complaint about the environment.
The participant had to go to the homepage and find the webpage with information about submitting environment complaints. Then the participant had to find out how he could submit an environment complaint and if he could submit his complaint on-line. After finding the webpage with information the timer was stopped.

5. Assignment 5: apply for a subsidy request.
The participant had to go to the homepage and find the webpage with information about applying for a subsidy for culture and welfare. Then the participant had to find out how he could apply for a subsidy request and if he could apply it on-line. Applying on-line means that the participant had to find an on-line interactive form for completion. So a MS Word or PDF document was not sufficient. After finding the webpage with information the timer was stopped.

After these assignments the participants had to answer which Dutch province in their opinion had the best website, compared to the other Dutch province they did the assignments for. If a participant had a comment, he could write this down.

D.1.4 Results

We found the following results:

- Visitors were frequently unable to find the information they were looking for on the websites. This was often due to the lack of a search and good navigation support, and not optimal search facilities provided by the websites.

- Visitors had difficulty with navigation through the websites because the menu names were not chosen from the point of view of the visitors.

- Visitors got lost easily because they did not know where they were in their search in the website.
• Visitors indicated that the written information on the websites was not easy to understand for the average citizen.

• Visitors indicated that the URL of some websites was not logical. This made it difficult to guess the organization’s URL.

D.1.5 Conclusion

The study showed that the average score of effectiveness for the Dutch provinces is 54% vs. 58% for the charities. This is not good. The results of the Total Review™ Method for the websites of the provinces showed us that the websites of the provinces score poor on the effectiveness criteria. The average score on usability is 57%. This means that the interaction with the province websites was not very well. Together with a poor functionality (45%) of the websites this makes it for the users very difficult to find the desired information. This is also true for the website of the charities. The websites of the charities scored on the effectiveness criterion usability and functionality a little bit better than the websites of the provinces; respectively 65% and 56% vs. 57% and 45%. We see that the results of the Total Review™ Method for the province websites are in line with the results of the User Test for the province websites when we compare the results of the Total Review™ Method for the province websites.

The User Test showed that the websites of provinces do not always have clear information for their users. They do not explain very well how to obtain information when it is not available on the web (e.g., risk map). The navigation is not always easy due to the meaningless terms chosen for the buttons or links. The search facilities provided by the websites are not optimal. Bad navigation and lack of search support make it difficult for the user to find the desired information. It takes too much clicks and therefore time to find the information (if it is available on the website). The consequence is that users will be irritated, annoyed and leave the site unsuccessfully.

As the success of non-profit websites is mainly dependent on finding the desired information easily, the websites involved in this study should be optimized (in particular on the points we found in the results section) and made also accessible for users doing a wild search.
D.2 Path breadcrumbs in practice

D.2.1 Research question
How can we make the actual path followed by a visitor visible on a website and can we provide this as a usable way to provide step back facilities?

D.2.2 Study
The goal of this study was to collect information about breadcrumbs and to show the possible ways to implement path breadcrumb. We created a framework for path breadcrumbs which showed the visitor’s history in the website, i.e. the path the visitor has taken to reach the current page. So, path breadcrumbs are dynamic instead of static.

D.2.3 Approach
We implemented 5 frameworks and worked them out:

1. A PHP framework with sessions,
2. A PHP framework with cookies,
3. A PHP framework with external file,
4. A JavaScript framework, and
5. A PHP framework with database support.

The advantages and the disadvantages of all the frameworks were discussed and compared with each other on performance.

D.2.4 Results
Table D.4 shows the advantages and disadvantages of the implemented frameworks.

D.2.5 Conclusion
This study showed that using path breadcrumbs is a possible way to follow the visitor’s path on a website. Path breadcrumb is also a benefit for the visitor as he can see which path he has taken to reach the current page. The
<table>
<thead>
<tr>
<th>Frameworks</th>
<th>Advantages</th>
<th>Disadvantages</th>
<th>Performance (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Embedded in HTML</td>
<td>Server sided</td>
<td>Easy to code</td>
</tr>
<tr>
<td>PHP framework</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>with sessions</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PHP framework</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>with cookies</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PHP framework</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>with external file</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JavaScript framework</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>PHP framework with database support</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

Table D.4: Advantages and disadvantages of the implemented frameworks.

disadvantage of path breadcrumbs is that there is limited space available on the screen to display all the steps taken by the visitor. Therefore, only a limited number of pages can be displayed in the path breadcrumb on the current page. This can be solved by a “more results” link by which the visitor can see his whole browsing history on the website. In this way we can provide with path breadcrumbs a means to step back.
Bibliography


List of acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AJAX</td>
<td>Asynchronous JavaScript and XML</td>
</tr>
<tr>
<td>CMS</td>
<td>Content Management System</td>
</tr>
<tr>
<td>CRSP</td>
<td>Consumer’s Retail Search Process</td>
</tr>
<tr>
<td>CSS</td>
<td>Cascading Style Sheets</td>
</tr>
<tr>
<td>HCI</td>
<td>Human Computer Interaction</td>
</tr>
<tr>
<td>HTML</td>
<td>HyperText Markup Language</td>
</tr>
<tr>
<td>MDP</td>
<td>Markov decision problems or Markov decision processes</td>
</tr>
<tr>
<td>MVT</td>
<td>Multivariate Testing</td>
</tr>
<tr>
<td>NCSA</td>
<td>National Center for Supercomputing Applications</td>
</tr>
<tr>
<td>RSS</td>
<td>Really Simple Syndication</td>
</tr>
<tr>
<td>TLD</td>
<td>Top-Level Domain</td>
</tr>
<tr>
<td>UI</td>
<td>User Interface</td>
</tr>
<tr>
<td>URL</td>
<td>Uniform Resource Locator</td>
</tr>
<tr>
<td>WSO</td>
<td>Website Optimization</td>
</tr>
<tr>
<td>WWW</td>
<td>World Wide Web</td>
</tr>
<tr>
<td>XHTML</td>
<td>Extensible Hypertext Markup Language</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
</tbody>
</table>
## Glossary

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AJAX</strong></td>
<td>Asynchronous JavaScript and XML (AJAX) is a group of interrelated web development techniques used to create interactive web applications or rich Internet applications. With AJAX, web applications can retrieve data from the server asynchronously in the background without interfering with the display and behavior of the existing page.</td>
</tr>
<tr>
<td><strong>Atomic observation unit</strong></td>
<td>An atomic observation unit is something relevant that is observed and can be described in a single sentence.</td>
</tr>
<tr>
<td><strong>CMS</strong></td>
<td>Content Management System (CMS) is a computer application used to manage workflow needed to collaboratively create, edit, review, index, search, publish and archive various kinds of digital media and electronic text.</td>
</tr>
<tr>
<td><strong>CSS</strong></td>
<td>Cascading Style Sheets (CSS) is a style sheet language used to describe the presentation (that is, the look and formatting) of a document written in a markup language. Its most common application is to style web pages written in HyperText Markup Language (HTML) and Extensible Hypertext Markup Language (XHTML), but the language can be applied to any kind of Extensible Markup Language (XML) document.</td>
</tr>
</tbody>
</table>
**Design pattern**
A design pattern is a formal way of documenting a solution to a common design problem. In its simplest form, a design pattern is an optimal solution to a common problem in specific context(s).

**Direct Social Navigation**
In direct social navigation the communication is two-way (e.g., talking) between the user (i.e., the seeker) and others (e.g., a person or artificial agent). A user can ask questions like “Where am I?” or “Where can I find location X?” to someone. This person answers the user and, perhaps more importantly, can ask the user to clarify his questions. The person can, thus, help clarify a user’s goals or even change them. When a user is uncertain of where he wants to go the advice provider can support him in formulating a destination.

**Dynamic website**
A dynamic website is a database driven website where content and design live separately. The content lives in a database that is placed on a webpage only when needed or asked.

**Epistemic Search Strategy**
A search strategy that relies on consumer’s spatial knowledge (or cognitive map) of the shopping environment, or a strategy that requires the assistance of others (e.g., store employees, other patrons) when consumers get lost. These consumers are concerned with efficiency and would want to complete their shopping trips at the soonest possible time. They more likely plan their purchases and move through the store as quickly as they can.
Ethnography

Ethnography is the scientific study of human social phenomena and communities, through means such as fieldwork or field research. It is considered a branch of cultural anthropology, the branch of anthropology which focuses on the study of human societies.

Exploration

Exploration is navigation without a specific destination. People are not so much interested in a specific location, but they are more interested in exploring the space they are in. They are more open to following a crowd of people or randomly choosing a route; destination and correct route are of less importance.

Findability

Findability refers to the quality of being locatable or navigable. At the item level, we can evaluate to what degree a particular object is easy to discover or locate. At the system level, we can analyze how well a physical or digital environment supports navigation and retrieval.

Full Field Notes

Working out the notes to complete reports. This should not be done later than the morning of the next day because new experiences push the old ones. That is why one should note the last ones first. Writing stimulates the reflection on the events which should be noted too.

Hedonic Search Strategy

A search strategy that comprises the experiential aspect of the retail search activity. Consumers who want to shop likely use hedonic search strategies when they are unfamiliar with their shopping environment and, thus, proceed more slowly while valuing environmental sensory stimulation. They have a higher tendency to make impulsive purchases.
HTML

HyperText Markup Language or HTML is a markup language for web pages. It provides a means to describe the structure of text-based information in a document by denoting certain text as links, headings, paragraphs, lists, and so on and to supplement that text with interactive forms, embedded images, and other objects. HTML is written in the form of tags, surrounded by angle brackets. HTML is constantly undergoing revision and evolution to meet the demands and requirements of the growing Internet audience under the direction of the W3C, the organization charged with designing and maintaining the language.

Impulse Buying

Impulsive buying is a spontaneous and immediate purchase behavior in which the consumer buys a product that he was not actively looking for and had no prior plans to purchase it.

Indirect Social Navigation

In indirect social navigation the communication is in one direction. This means that advice providers (e.g., a shop assistance, an employee) do not have to be aware of the fact that they are giving advice (e.g., by showing the user a path to a department).

Involuntary browsing

Involuntary browsing in an unintentional browsing behaviour in which the user is unaware of any latent goal that might be pursued.

Javascript

A scripting programming language most commonly used to add interactive features to webpages. JavaScript is a trademark of Sun Microsystems.
<table>
<thead>
<tr>
<th>Glossary Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jotted Notes</td>
<td>Making short notes during or immediately after the observations; some words or quotations as reminder. Sometimes there are opportunities to make detailed notes or to work out earlier notes.</td>
</tr>
<tr>
<td>Landmark</td>
<td>Landmarks are objects in a space (e.g., information space) that serve as reference points to people.</td>
</tr>
<tr>
<td>Local breadcrumb</td>
<td>Location breadcrumbs convey the position of the page within the site hierarchy. This is the most common type of breadcrumb on the web because with static pages, this is the only reasonable implementation. Users can often take several different routes to a page, but the breadcrumb will tell them “where they are.”</td>
</tr>
<tr>
<td>Map</td>
<td>A survey representation of an environment. We typically find two variants of the basic map: you-are-here maps (a map that marks the position of the person looking at it), and route maps (maps with an explicit route marked).</td>
</tr>
<tr>
<td>Mental Notes</td>
<td>Expressly focus one’s attention on the things you should note later, like the basis information of the scenes and the episodes, and the remarkable events in it.</td>
</tr>
<tr>
<td>Natural Search Behavior</td>
<td>Search behavior of people in their natural setting.</td>
</tr>
<tr>
<td>Term</td>
<td>Definition</td>
</tr>
<tr>
<td>-------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Object Identification</td>
<td>Object identification deals with new spaces. Spaces consist of objects. In new spaces the first thing people have to do is to identify the different objects and their respective attributes such as identifying the reference points; people have the ability to generalize from past experiences.</td>
</tr>
<tr>
<td>Opportunistic browsing</td>
<td>Opportunistic browsing is an intentional browsing behaviour in which the user is unaware of any goal being pursued. The user’s attitude is ‘let’s see what’s there’.</td>
</tr>
<tr>
<td>Participating Observation</td>
<td>The ethnographic manner to acquire knowledge which is a variant on everyday manners to know something of the life of other people; this ethnographic manner is a mixture of looking at, participating in, and talking with the others.</td>
</tr>
<tr>
<td>Path breadcrumb</td>
<td>Path breadcrumbs show the path the user has taken within the site to get to the current page. The same content from the site can be presented with different breadcrumbs because users can take different routes.</td>
</tr>
<tr>
<td>Pay-Per-Click</td>
<td>Pay-Per-Click (PPC) is an advertising model used on search engines, advertising networks, and content websites/blogs, where advertisers only pay when a user actually clicks on an ad to visit the advertiser’s website.Advertisers bid on keywords they predict their target market will use as search terms when they are looking for a product or service.</td>
</tr>
<tr>
<td>Plugin</td>
<td>A plugin or add-on is a small program that often provides additional functionality that is not available in the standard application.</td>
</tr>
<tr>
<td>Precise goal</td>
<td>A well-defined goal; the goal is clear and unambiguous.</td>
</tr>
<tr>
<td>Term</td>
<td>Definition</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>RSS</td>
<td>Really Simple Syndication (RSS) is a family of web feed formats used to publish frequently updated works such as blog entries, news headlines, audio, and video in a standardized format.</td>
</tr>
<tr>
<td>Search browsing</td>
<td>Search browsing is an intentional browsing behaviour in which the user is aware of the goal being pursued. The goal is defined precisely or otherwise.</td>
</tr>
<tr>
<td>Search unit</td>
<td>Search units are specific unambiguous categories to do searches which can only occur on the WWW or in browsers. Consequently search units are also atomic observation units.</td>
</tr>
<tr>
<td>SEO</td>
<td>Search Engine Optimization (SEO) is the active practice of optimizing a website by improving internal and external aspects in order to increase the traffic the site receives from search engines.</td>
</tr>
<tr>
<td>Social Navigation</td>
<td>Navigation which is based on what others have done or the advice provided by others (e.g., follow people, ask a friend).</td>
</tr>
<tr>
<td>State Trigger</td>
<td>Thing that triggers people at the state in which they are finding themselves in.</td>
</tr>
<tr>
<td>Subjective Search Behavior</td>
<td>The way the participants experienced their behavior.</td>
</tr>
<tr>
<td>Summary observation unit</td>
<td>A summary of the total observation.</td>
</tr>
<tr>
<td>TLD</td>
<td>Top-Level Domain (TLD) is the last part of an Internet domain name, that is, the group of letters that follow the final dot of a fully qualified domain name (the absolute domain name). For example, in the domain name <a href="http://www.example.com">www.example.com</a>, the top-level domain is com.</td>
</tr>
<tr>
<td>Triggered Search Behavior</td>
<td>The behavior we observe when people are completing a task.</td>
</tr>
</tbody>
</table>
URL
Universal Resource Locator is the full unique address of websites/pages/files on the Internet e.g., the URL of the Google website is http://www.google.com.

Usability
Usability (a subset of user-experience) is a property of websites (and other systems and products) that relates to ease of use. For websites this means that usability is making your site easy for your visitors to find the exact information they need when they need it. Anything that makes the process slower (like Flash animation served to a dial-up visitor) inhibits usability. Conversely, easy, intuitive navigation and strong, informative text enhance usability.

User-experience
User-experience focuses on the users’ feeling when they are visiting a website. The aim of user-experience is to give users a good feeling.

Wayfinding
Wayfinding is the activity of going from one place to another.

Website optimization
Website optimization is the process of improving internal (e.g., layout of webpages, content, navigation support, usability) and external (e.g., promotion of the website, link building, building personality and reputation) aspects of your webpages to increase the traffic the website receives from search engines.

Wild Searching
Realizing a user goal, which is not defined precisely or which may change, by browsing or surfing.
SIKS-dissertation series

1998

1998-1 Johan van den Akker (CWI)
DEGAS - An Active, Temporal Database of Autonomous Objects

1998-2 Floris Wiesman (UM)
Information Retrieval by Graphically Browsing Meta-Information

1998-3 Ans Steuten (TUD)
A Contribution to the Linguistic Analysis of Business Conversations within the Language/Action Perspective

1998-4 Dennis Breuker (UM)
Memory versus Search in Games

1998-5 E.W.Oskamp (RUL)
Computerondersteuning bij Straftoemeting

1999

1999-1 Mark Sloof (VU)
Physiology of Quality Change Modelling; Automated modelling of Quality Change of Agricultural Products

1999-2 Rob Potharst (EUR)
Classification using decision trees and neural nets

1999-3 Don Beal (UM)
The Nature of Minimax Search

1999-4 Jacques Penders (UM)
The practical Art of Moving Physical Objects
1999-5 Aldo de Moor (KUB)  
Empowering Communities: A Method for the Legitimate User-Driven  
Specification of Network Information Systems

1999-6 Niek J.E. Wijngaards (VU)  
Re-design of compositional systems

1999-7 David Spelt (UT)  
Verification support for object database design

1999-8 Jacques H.J. Lenting (UM)  
Informed Gambling: Conception and Analysis of a Multi-Agent  
Mechanism for Discrete Reallocation.

====

2000
====

2000-1 Frank Niessink (VU)  
Perspectives on Improving Software Maintenance

2000-2 Koen Holtman (TUE)  
Prototyping of CMS Storage Management

2000-3 Carolien M.T. Metselaar (UVA)  
Sociaal-organisatorische gevolgen van kennistechnologie;  
een procesbenadering en actorperspectief.

2000-4 Geert de Haan (VU)  
ETAG, A Formal Model of Competence Knowledge for User Interface Design

2000-5 Ruud van der Pol (UM)  
Knowledge-based Query Formulation in Information Retrieval.

2000-6 Rogier van Eijk (UU)  
Programming Languages for Agent Communication

2000-7 Niels Peek (UU)  
Decision-theoretic Planning of Clinical Patient Management

2000-8 Veerle Coup (EUR)  
Sensitivity Analysis of Decision-Theoretic Networks

2000-9 Florian Waas (CWI)  
Principles of Probabilistic Query Optimization

2000-10 Niels Nes (CWI)  
Image Database Management System Design Considerations,  
Algorithms and Architecture

2000-11 Jonas Karlsson (CWI)  
Scalable Distributed Data Structures for Database Management
2001

2001-1 Silja Renooij (UU)
Qualitative Approaches to Quantifying Probabilistic Networks

2001-2 Koen Hindriks (UU)
Agent Programming Languages: Programming with Mental Models

2001-3 Maarten van Someren (UvA)
Learning as problem solving

2001-4 Evgueni Smirnov (UM)
Conjunctive and Disjunctive Version Spaces with Instance-Based Boundary Sets

2001-5 Jacco van Ossenbruggen (VU)
Processing Structured Hypermedia: A Matter of Style

2001-6 Martijn van Welie (VU)
Task-based User Interface Design

2001-7 Bastiaan Schonhage (VU)
Diva: Architectural Perspectives on Information Visualization

2001-8 Pascal van Eck (VU)
A Compositional Semantic Structure for Multi-Agent Systems Dynamics.

2001-9 Pieter Jan ’t Hoen (RUL)
Towards Distributed Development of Large Object-Oriented Models,
Views of Packages as Classes

2001-10 Maarten Sierhuis (UvA)
Modeling and Simulating Work Practice
BRAHMS: a multiagent modeling and simulation language
for work practice analysis and design

2001-11 Tom M. van Engers (VUA)
Knowledge Management:
The Role of Mental Models in Business Systems Design

2002

2002-01 Nico Lassing (VU)
Architecture-Level Modifiability Analysis

2002-02 Roelof van Zwol (UT)
Modelling and searching web-based document collections
2002-03 Henk Ernst Blok (UT)
Database Optimization Aspects for Information Retrieval

2002-04 Juan Roberto Castelo Valdueza (UU)
The Discrete Acyclic Digraph Markov Model in Data Mining

2002-05 Radu Serban (VU)
The Private Cyberspace Modeling Electronic Environments
inhabited by Privacy-concerned Agents

2002-06 Laurens Mommers (UL)
Applied legal epistemology;
Building a knowledge-based ontology of the legal domain

2002-07 Peter Boncz (CWI)
Monet: A Next-Generation DBMS Kernel For Query-Intensive Applications

2002-08 Jaap Gordijn (VU)
Value Based Requirements Engineering: Exploring Innovative
E-Commerce Ideas

2002-09 Willem-Jan van den Heuvel(KUB)
Integrating Modern Business Applications with Objectified Legacy Systems

2002-10 Brian Sheppard (UM)
Towards Perfect Play of Scrabble

2002-11 Wouter C.A. Wijngaards (VU)
Agent Based Modelling of Dynamics: Biological and Organisational Applications

2002-12 Albrecht Schmidt (Uva)
Processing XML in Database Systems

2002-13 Hongjing Wu (TUE)
A Reference Architecture for Adaptive Hypermedia Applications

2002-14 Wieke de Vries (UU)
Agent Interaction: Abstract Approaches to Modelling, Programming and
Verifying Multi-Agent Systems

2002-15 Rik Eshuis (UT)
Semantics and Verification of UML Activity Diagrams for Workflow Modelling

2002-16 Pieter van Langen (VU)
The Anatomy of Design: Foundations, Models and Applications

2002-17 Stefan Manegold (UVA)
Understanding, Modeling, and Improving Main-Memory Database Performance
2003-01 Heiner Stuckenschmidt (VU)
Ontology-Based Information Sharing in Weakly Structured Environments

2003-02 Jan Broersen (VU)
Modal Action Logics for Reasoning About Reactive Systems

2003-03 Martijn Schuemie (TUD)
Human-Computer Interaction and Presence in Virtual Reality Exposure Therapy

2003-04 Milan Petkovic (UT)
Content-Based Video Retrieval Supported by Database Technology

2003-05 Jos Lehmann (UVA)
Causation in Artificial Intelligence and Law - A modelling approach

2003-06 Boris van Schooten (UT)
Development and specification of virtual environments

2003-07 Machiel Jansen (UvA)
Formal Explorations of Knowledge Intensive Tasks

2003-08 Yongping Ran (UM)
Repair Based Scheduling

2003-09 Rens Kortmann (UM)
The resolution of visually guided behaviour

2003-10 Andreas Lincke (UvT)
Electronic Business Negotiation: Some experimental studies on the interaction between medium, innovation context and culture

2003-11 Simon Keizer (UT)
Reasoning under Uncertainty in Natural Language Dialogue using Bayesian Networks

2003-12 Roeland Ordeman (UT)
Dutch speech recognition in multimedia information retrieval

2003-13 Jeroen Donkers (UM)
Nosce Hostem - Searching with Opponent Models

2003-14 Stijn Hoppenbrouwers (KUN)
Freezing Language: Conceptualisation Processes across ICT-Supported Organisations

2003-15 Mathijs de Weerdt (TUD)
Plan Merging in Multi-Agent Systems

2003-16 Menzo Windhouwer (CWI)
Feature Grammar Systems - Incremental Maintenance of Indexes to Digital Media Warehouses
2003-17 David Jansen (UT)
    Extensions of Statecharts with Probability, Time, and Stochastic Timing

2003-18 Levente Kocsis (UM)
    Learning Search Decisions

====

2004
====

2004-01 Virginia Dignum (UU)
    A Model for Organizational Interaction: Based on Agents, Founded in Logic

2004-02 Lai Xu (UvT)
    Monitoring Multi-party Contracts for E-business

2004-03 Perry Groot (VU)
    A Theoretical and Empirical Analysis of Approximation in Symbolic Problem Solving

2004-04 Chris van Aart (UVA)
    Organizational Principles for Multi-Agent Architectures

2004-05 Viara Popova (EUR)
    Knowledge discovery and monotonicity

2004-06 Bart-Jan Hommes (TUD)
    The Evaluation of Business Process Modeling Techniques

2004-07 Elise Boltjes (UM)
    Voorbeeldig onderwijs; voorbeeldgestuurd onderwijs, een opstap naar abstract denken, vooral voor meisjes

2004-08 Joop Verbeek (UM)
    Politie en de Nieuwe Internationale Informatiemarkt, Grensregionale politiële gegevensuitwisseling en digitale expertise

2004-09 Martin Caminada (VU)
    For the Sake of the Argument; explorations into argument-based reasoning

2004-10 Suzanne Kabel (UVA)
    Knowledge-rich indexing of learning-objects

2004-11 Michel Klein (VU)
    Change Management for Distributed Ontologies

2004-12 The Duy Bui (UT)
    Creating emotions and facial expressions for embodied agents

2004-13 Wojciech Jamroga (UT)
    Using Multiple Models of Reality: On Agents who Know how to Play

2004-14 Paul Harrenstein (UU)
    Logic in Conflict. Logical Explorations in Strategic Equilibrium
2004-15 Arno Knobbe (UU)  
   Multi-Relational Data Mining

2004-16 Federico Divina (VU)  
   Hybrid Genetic Relational Search for Inductive Learning

2004-17 Mark Winands (UM)  
   Informed Search in Complex Games

2004-18 Vania Bessa Machado (UvA)  
   Supporting the Construction of Qualitative Knowledge Models

2004-19 Thijs Westerveld (UT)  
   Using generative probabilistic models for multimedia retrieval

2004-20 Madelon Evers (Nyenrode)  
   Learning from Design: facilitating multidisciplinary design teams

2005-01 Floor Verdenius (UVA)  
   Methodological Aspects of Designing Induction-Based Applications

2005-02 Erik van der Werf (UM))  
   AI techniques for the game of Go

2005-03 Franc Grootjen (RUN)  
   A Pragmatic Approach to the Conceptualisation of Language

2005-04 Nirvana Meratnia (UT)  
   Towards Database Support for Moving Object data

2005-05 Gabriel Infante-Lopez (UVA)  
   Two-Level Probabilistic Grammars for Natural Language Parsing

2005-06 Pieter Spronck (UM)  
   Adaptive Game AI

2005-07 Flavius Frasincar (TUE)  
   Hypermedia Presentation Generation for Semantic Web Information Systems

2005-08 Richard Vdovjak (TUE)  
   A Model-driven Approach for Building Distributed Ontology-based Web Applications

2005-09 Jeen Broekstra (VU)  
   Storage, Querying and Inferencing for Semantic Web Languages

2005-10 Anders Bouwer (UVA)  
   Explaining Behaviour: Using Qualitative Simulation in Interactive Learning Environments
2005-11 Elth Ogston (VU)
Agent Based Matchmaking and Clustering - A Decentralized Approach to Search

2005-12 Csaba Boer (EUR)
Distributed Simulation in Industry

2005-13 Fred Hamburg (UL)
Een Computermodel voor het Ondersteunen van Euthanasiebeslissingen

2005-14 Borys Omelayenko (VU)
Web-Service configuration on the Semantic Web; Exploring how semantics meets pragmatics

2005-15 Tibor Bosse (VU)
Analysis of the Dynamics of Cognitive Processes

2005-16 Joris Graaumans (UU)
Usability of XML Query Languages

2005-17 Boris Shishkov (TUD)
Software Specification Based on Re-usable Business Components

2005-18 Danielle Sent (UU)
Test-selection strategies for probabilistic networks

2005-19 Michel van Dartel (UM)
Situated Representation

2005-20 Cristina Coteanu (UL)
Cyber Consumer Law, State of the Art and Perspectives

2005-21 Wijnand Derks (UT)
Improving Concurrency and Recovery in Database Systems by Exploiting Application Semantics

====

2006
====

2006-01 Samuil Angelov (TUE)
Foundations of B2B Electronic Contracting

2006-02 Cristina Chisalita (VU)
Contextual issues in the design and use of information technology in organizations

2006-03 Noor Christoph (UVA)
The role of metacognitive skills in learning to solve problems

2006-04 Marta Sabou (VU)
Building Web Service Ontologies

2006-05 Cees Pierik (UU)
Validation Techniques for Object-Oriented Proof Outlines
2006-06 Ziv Baida (VU)
    Software-aided Service Bundling - Intelligent Methods & Tools
    for Graphical Service Modeling

2006-07 Marko Smiljanic (UT)
    XML schema matching – balancing efficiency and effectiveness by means of clustering

2006-08 Eelco Herder (UT)
    Forward, Back and Home Again - Analyzing User Behavior on the Web

2006-09 Mohamed Wahdan (UM)
    Automatic Formulation of the Auditor’s Opinion

2006-10 Ronny Siebes (VU)
    Semantic Routing in Peer-to-Peer Systems

2006-11 Joeri van Ruth (UT)
    Flattening Queries over Nested Data Types

2006-12 Bert Bongers (VU)
    Interactivation - Towards an e-ology of people, our technological environment, and the arts

2006-13 Henk-Jan Lebbink (UU)
    Dialogue and Decision Games for Information Exchanging Agents

2006-14 Johan Hoorn (VU)
    Software Requirements: Update, Upgrade, Redesign - towards a Theory of Requirements Change

2006-15 Rainer Malik (UU)
    CONAN: Text Mining in the Biomedical Domain

2006-16 Carsten Riggelsen (UU)
    Approximation Methods for Efficient Learning of Bayesian Networks

2006-17 Stacey Nagata (UU)
    User Assistance for Multitasking with Interruptions on a Mobile Device

2006-18 Valentin Zhizhkun (UVA)
    Graph transformation for Natural Language Processing

2006-19 Birna van Riemsdijk (UU)
    Cognitive Agent Programming: A Semantic Approach

2006-20 Marina Velikova (UvT)
    Monotone models for prediction in data mining

2006-21 Bas van Gils (RUN)
    Aptness on the Web

2006-22 Paul de Vrieze (RUN)
    Fundamentals of Adaptive Personalisation
2006-23 Ion Juvina (UU)
  Development of Cognitive Model for Navigating on the Web

2006-24 Laura Hollink (VU)
  Semantic Annotation for Retrieval of Visual Resources

2006-25 Madalina Drugan (UU)
  Conditional log-likelihood MDL and Evolutionary MCMC

2006-26 Vojkan Mihajlović (UT)
  Score Region Algebra: A Flexible Framework for Structured Information Retrieval

2006-27 Stefano Bocconi (CWI)
  Vox Populi: generating video documentaries from semantically annotated media repositories

2006-28 Borkur Sigurbjörnsson (UVA)
  Focused Information Access using XML Element Retrieval

=====
2007
=====

2007-01 Kees Leune (UvT)
  Access Control and Service-Oriented Architectures

2007-02 Wouter Teepe (RUG)
  Reconciling Information Exchange and Confidentiality: A Formal Approach

2007-03 Peter Mika (VU)
  Social Networks and the Semantic Web

2007-04 Jurriaan van Diggelen (UU)
  Achieving Semantic Interoperability in Multi-agent Systems: a dialogue-based approach

2007-05 Bart Schermer (UL)
  Software Agents, Surveillance, and the Right to Privacy: a Legislative Framework for Agent-enabled Surveillance

2007-06 Gilad Mishne (UVA)
  Applied Text Analytics for Blogs

2007-07 Natasa Jovanovic’ (UT)
  To Whom It May Concern - Addressee Identification in Face-to-Face Meetings

2007-08 Mark Hoogendoorn (VU)
  Modeling of Change in Multi-Agent Organizations

2007-09 David Mobach (VU)
  Agent-Based Mediated Service Negotiation

2007-10 Huib Aldewereld (UU)
  Autonomy vs. Conformity: an Institutional Perspective on Norms and Protocols
2007-11 Natalia Stash (TUE)
   Incorporating Cognitive/Learning Styles in a General-Purpose Adaptive Hypermedia System

2007-12 Marcel van Gerven (RUN)
   Bayesian Networks for Clinical Decision Support: A Rational Approach to Dynamic Decision-Making under Uncertainty

2007-13 Rutger Rienks (UT)
   Meetings in Smart Environments; Implications of Progressing Technology

2007-14 Niek Bergboer (UM)
   Context-Based Image Analysis

2007-15 Joyca Lacroix (UM)
   NIM: a Situated Computational Memory Model

2007-16 Davide Grossi (UU)
   Designing Invisible Handcuffs. Formal investigations in Institutions and Organizations for Multi-agent Systems

2007-17 Theodore Charitos (UU)
   Reasoning with Dynamic Networks in Practice

2007-18 Bart Orriens (UvT)
   On the development an management of adaptive business collaborations

2007-19 David Levy (UM)
   Intimate relationships with artificial partners

2007-20 Slinger Jansen (UU)
   Customer Configuration Updating in a Software Supply Network

2007-21 Karianne Vermaas (UU)
   Fast diffusion and broadening use: A research on residential adoption and usage of broadband internet in the Netherlands between 2001 and 2005

2007-22 Zlatko Zlatev (UT)
   Goal-oriented design of value and process models from patterns

2007-23 Peter Barna (TUE)
   Specification of Application Logic in Web Information Systems

2007-24 Georgina Ramrez Camps (CWI)
   Structural Features in XML Retrieval

2007-25 Joost Schalken (VU)
   Empirical Investigations in Software Process Improvement
2008-01 Katalin Boer-Sorbn (EUR)
Agent-Based Simulation of Financial Markets: A modular, continuous-time approach

2008-02 Alexei Sharpanskykh (VU)
On Computer-Aided Methods for Modeling and Analysis of Organizations

2008-03 Vera Hollink (UVA)
Optimizing hierarchical menus: a usage-based approach

2008-04 Ander de Keijzer (UT)
Management of Uncertain Data - towards unattended integration

2008-05 Bela Mutschler (UT)
Modeling and simulating causal dependencies on process-aware information systems from a cost perspective

2008-06 Arjen Hommersom (RUN)
On the Application of Formal Methods to Clinical Guidelines, an Artificial Intelligence Perspective

2008-07 Peter van Rosmalen (OU)
Supporting the tutor in the design and support of adaptive e-learning

2008-08 Janneke Bolt (UU)
Bayesian Networks: Aspects of Approximate Inference

2008-09 Christof van Nimwegen (UU)
The paradox of the guided user: assistance can be counter-effective

2008-10 Wauter Bosma (UT)
Discourse oriented summarization

2008-11 Vera Kartseva (VU)
Designing Controls for Network Organizations: A Value-Based Approach

2008-12 Jozsef Farkas (RUN)
A Semiotically Oriented Cognitive Model of Knowledge Representation

2008-13 Caterina Carraciolo (UVA)
Topic Driven Access to Scientific Handbooks

2008-14 Arthur van Bunningen (UT)
Context-Aware Querying; Better Answers with Less Effort

2008-15 Martijn van Otterlo (UT)

2008-16 Henriette van Vugt (VU)
Embodyed agents from a user's perspective
2008-17 Martin Op ’t Land (TUD)
Applying Architecture and Ontology to the Splitting and Allying of Enterprises

2008-18 Guido de Croon (UM)
Adaptive Active Vision

2008-19 Henning Rode (UT)
From Document to Entity Retrieval: Improving Precision and Performance of Focused Text Search

2008-20 Rex Arendsen (UVA)
Geen bericht, goed bericht. Een onderzoek naar de effecten van de introductie van elektronisch berichtenverkeer met de overheid op de administratieve lasten van bedrijven

2008-21 Krisztian Balog (UVA)
People Search in the Enterprise

2008-22 Henk Koning (UU)
Communication of IT-Architecture

2008-23 Stefan Visscher (UU)
Bayesian network models for the management of ventilator-associated pneumonia

2008-24 Zharko Aleksovski (VU)
Using background knowledge in ontology matching

2008-25 Geert Jonker (UU)
Efficient and Equitable Exchange in Air Traffic Management Plan Repair using Spender-signed Currency

2008-26 Marijn Huijbregts (UT)
Segmentation, Diarization and Speech Transcription: Surprise Data Unraveled

2008-27 Hubert Vogten (OU)
Design and Implementation Strategies for IMS Learning Design

2008-28 Ildiko Flesch (RUN)
On the Use of Independence Relations in Bayesian Networks

2008-29 Dennis Reidsma (UT)
Annotations and Subjective Machines - Of Annotators, Embodied Agents, Users, and Other Humans

2008-30 Wouter van Atteveldt (VU)
Semantic Network Analysis: Techniques for Extracting, Representing and Querying Media Content

2008-31 Loes Braun (UM)
Pro-Active Medical Information Retrieval

2008-32 Trung H. Bui (UT)
Toward Affective Dialogue Management using Partially Observable Markov Decision Processes
2008-33 Frank Terpstra (UVA)  
Scientific Workflow Design; theoretical and practical issues

2008-34 Jeroen de Knijf (UU)  
Studies in Frequent Tree Mining

2008-35 Ben Torben Nielsen (UvT)  
Dendritic morphologies: function shapes structure

====

2009

====

2009-01 Rasa Jurgelenaite (RUN)  
Symmetric Causal Independence Models

2009-02 Willem Robert van Hage (VU)  
Evaluating Ontology-Alignment Techniques

2009-03 Hans Stol (UvT)  
A Framework for Evidence-based Policy Making Using IT

2009-04 Josephine Nabukenya (RUN)  
Improving the Quality of Organisational Policy Making using Collaboration Engineering

2009-05 Sietse Overbeek (RUN)  
Bridging Supply and Demand for Knowledge Intensive Tasks - Based on Knowledge, Cognition, and Quality

2009-06 Muhammad Subianto (UU)  
Understanding Classification

2009-07 Ronald Poppe (UT)  
Discriminative Vision-Based Recovery and Recognition of Human Motion

2009-08 Volker Nannen (VU)  
Evolutionary Agent-Based Policy Analysis in Dynamic Environments

2009-09 Benjamin Kanagwa (RUN)  
Design, Discovery and Construction of Service-oriented Systems

2009-10 Jan Wielemaker (UVA)  
Logic programming for knowledge-intensive interactive applications

2009-11 Alexander Boer (UVA)  
Legal Theory, Sources of Law & the Semantic Web

2009-12 Peter Massuthe (TUE, Humboldt-Universitaet zu Berlin)  
Operating Guidelines for Services

2009-13 Steven de Jong (UM)  
Fairness in Multi-Agent Systems
2009-14 Maksym Korotkiy (VU)  
From ontology-enabled services to service-enabled ontologies (making ontologies work in e-science with ONTO-SOA)

2009-15 Rinke Hoekstra (UVA)  
Ontology Representation - Design Patterns and Ontologies that Make Sense

2009-16 Fritz Reul (UvT)  
New Architectures in Computer Chess

2009-17 Laurens van der Maaten (UvT)  
Feature Extraction from Visual Data

2009-18 Fabian Groffen (CWI)  
Armada, An Evolving Database System

2009-19 Valentin Robu (CWI)  
Modeling Preferences, Strategic Reasoning and Collaboration in Agent-Mediated Electronic Markets

2009-20 Bob van der Vecht (UU)  
Adjustable Autonomy: Controlling Influences on Decision Making

2009-21 Stijn Vanderlooy (UM)  
Ranking and Reliable Classification

2009-22 Pavel Serdyukov (UT)  
Search For Expertise: Going beyond direct evidence

2009-23 Peter Hofgesang (VU)  
Modelling Web Usage in a Changing Environment

2009-24 Annerieke Heuvelink (VUA)  
Cognitive Models for Training Simulations

2009-25 Alex van Ballegooij (CWI)  
"RAM: Array Database Management through Relational Mapping"

2009-26 Fernando Koch (UU)  
An Agent-Based Model for the Development of Intelligent Mobile Services

2009-27 Christian Glahn (OU)  
Contextual Support of social Engagement and Reflection on the Web

2009-28 Sander Evers (UT)  
Sensor Data Management with Probabilistic Models

2009-29 Stanislav Pokraev (UT)  
Model-Driven Semantic Integration of Service-Oriented Applications

2009-30 Marcin Zukowski (CWI)  
Balancing vectorized query execution with bandwidth-optimized storage
2009-31 Sofiya Katrenko (UVA)
A Closer Look at Learning Relations from Text

2009-32 Rik Farenhorst (VU) and Remco de Boer (VU)
Architectural Knowledge Management: Supporting Architects and Auditors

2009-33 Khiet Truong (UT)
How Does Real Affect Affect Affect Recognition In Speech?

2009-34 Inge van de Weerd (UU)
Advancing in Software Product Management: An Incremental Method Engineering Approach

2009-35 Wouter Koelewijn (UL)
Privacy en Politiegegevens; Over geautomatiseerde normatieve informatie-uitwisseling

2009-36 Marco Kalz (OUN)
Placement Support for Learners in Learning Networks

2009-37 Hendrik Drachsler (OUN)
Navigation Support for Learners in Informal Learning Networks

2009-38 Riina Vuorikari (OU)
Tags and self-organisation: a metadata ecology for learning resources in a multilingual context

2009-39 Christian Stahl (TUE, Humboldt-Universitaet zu Berlin)
Service Substitution – A Behavioral Approach Based on Petri Nets

2009-40 Stephan Raaijmakers (UvT)
Multinomial Language Learning: Investigations into the Geometry of Language

2009-41 Igor Berezhnyy (UvT)
Digital Analysis of Paintings

2009-42 Toine Bogers
Recommender Systems for Social Bookmarking

2009-43 Virginia Nunes Leal Franqueira (UT)
Finding Multi-step Attacks in Computer Networks using Heuristic Search and Mobile Ambients

2009-44 Roberto Santana Tapia (UT)
Assessing Business-IT Alignment in Networked Organizations

2009-45 Jilles Vreeken (UU)
Making Pattern Mining Useful

2009-46 Loredana Afanasiev (UvA)
Querying XML: Benchmarks and Recursion
2010-01 Matthijs van Leeuwen (UU)
Patterns that Matter

2010-02 Ingo Wassink (UT)
Work flows in Life Science

2010-03 Joost Geurts (CWI)
A Document Engineering Model and Processing Framework for Multimedia documents

2010-04 Olga Kulyk (UT)
Do You Know What I Know? Situational Awareness of Co-located Teams in Multidisplay Environments

2010-05 Claudia Hauff (UT)
Predicting the Effectiveness of Queries and Retrieval Systems

2010-06 Sander Bakkes (UvT)
Rapid Adaptation of Video Game AI

2010-07 Wim Fikkert (UT)
Gesture interaction at a Distance

2010-08 Krzysztof Siewicz (UL)
Towards an Improved Regulatory Framework of Free Software. Protecting user freedoms in a world of software communities and eGovernments

2010-09 Hugo Kielman (UL)
A Politiele gegevensverwerking en Privacy, Naar een effectieve waarborging

2010-10 Rebecca Ong (UL)
Mobile Communication and Protection of Children

2010-11 Adriaan Ter Mors (TUD)
The world according to MARP: Multi-Agent Route Planning

2010-12 Susan van den Braak (UU)
Sensemaking software for crime analysis

2010-13 Gianluigi Folino (RUN)
High Performance Data Mining using Bio-inspired techniques

2010-14 Sander van Splunter (VU)
Automated Web Service Reconfiguration

2010-15 Lianne Bodenstaff (UT)
Managing Dependency Relations in Inter-Organizational Models

2010-16 Sicco Verwer (TUD)
Efficient Identification of Timed Automata, theory and practice
2010-17 Spyros Kotoulas (VU)
   Scalable Discovery of Networked Resources: Algorithms, Infrastructure, Applications

2010-18 Charlotte Gerritsen (VU)
   Caught in the Act: Investigating Crime by Agent-Based Simulation

2010-19 Henriette Cramer (UvA)
   People’s Responses to Autonomous and Adaptive Systems

2010-20 Ivo Swartjes (UT)
   Whose Story Is It Anyway? How Improv Informs Agency and Authorship of Emergent Narrative

2010-21 Harold van Heerde (UT)
   Privacy-aware data management by means of data degradation

2010-22 Michiel Hildebrand (CWI)
   End-user Support for Access to Heterogeneous Linked Data

2010-23 Bas Steunebrink (UU)
   The Logical Structure of Emotions

2010-24 Dmytro Tykhonov
   Designing Generic and Efficient Negotiation Strategies

2010-25 Zulfiqar Ali Memon (VU)
   Modelling Human-Awareness for Ambient Agents: A Human Mindreading Perspective

2010-26 Ying Zhang (CWI)
   XRPC: Efficient Distributed Query Processing on Heterogeneous XQuery Engines

2010-27 Marten Voulon (UL)
   Automatisch contracteren

2010-28 Arne Koopman (UU)
   Characteristic Relational Patterns

2010-29 Stratos Idreos(CWI)
   Database Cracking: Towards Auto-tuning Database Kernels

2010-30 Marieke van Erp (UvT)
   Accessing Natural History - Discoveries in data cleaning, structuring, and retrieval

2010-31 Victor de Boer (UVA)
   Ontology Enrichment from Heterogeneous Sources on the Web

2010-32 Marcel Hiel (UvT)
   An Adaptive Service Oriented Architecture: Automatically solving Interoperability Problems

2010-33 Robin Aly (UT)
   Modeling Representation Uncertainty in Concept-Based Multimedia Retrieval
2010-34 Teduh Dirgahayu (UT)
Interaction Design in Service Compositions

2010-35 Dolf Trieschnigg (UT)
Proof of Concept: Concept-based Biomedical Information Retrieval

2010-36 Jose Janssen (OU)
Paving the Way for Lifelong Learning; Facilitating competence development through a learning path specification

2010-37 Niels Lohmann (TUE)
Correctness of services and their composition

2010-38 Dirk Fahland (TUE)
From Scenarios to components

2010-39 Ghazanfar Farooq Siddiqui (VU)
Integrative modeling of emotions in virtual agents

2010-40 Mark van Assem (VU)
Converting and Integrating Vocabularies for the Semantic Web

2010-41 Guillaume Chaslot (UM)
Monte-Carlo Tree Search

2010-42 Sybren de Kinderen (VU)
Needs-driven service bundling in a multi-supplier setting - the computational e3-service approach

2010-43 Peter van Kranenburg (UU)
A Computational Approach to Content-Based Retrieval of Folk Song Melodies

2010-44 Pieter Bellekens (TUE)
An Approach towards Context-sensitive and User-adapted Access to Heterogeneous Data Sources, Illustrated in the Television Domain

2010-45 Vasilios Andrikopoulos (UvT)
A theory and model for the evolution of software services

2010-46 Vincent Pijpers (VU)
e3alignment: Exploring Inter-Organizational Business-ICT Alignment

2010-47 Chen Li (UT)
Mining Process Model Variants: Challenges, Techniques, Examples

2010-48 Milan Lovric (EUR)
Behavioral Finance and Agent-Based Artificial Markets

2010-49 Jahn-Takeshi Saito (UM)
Solving difficult game positions

2010-50 Bouke Huurnink (UVA)
Search in Audiovisual Broadcast Archives
Next, we will delve into the contributions of various researchers to the field of information systems. Here are some highlights from the dissertation series:

### 2010-51 Alia Khairia Amin (CWI)
Understanding and supporting information seeking tasks in multiple sources

### 2010-52 Peter-Paul van Maanen (VU)
Adaptive Support for Human-Computer Teams: Exploring the Use of Cognitive Models of Trust and Attention

### 2010-53 Edgar Meij (UVA)
Combining Concepts and Language Models for Information Access

---

### 2011-01 Botond Cseke (RUN)
Variational Algorithms for Bayesian Inference in Latent Gaussian Models

### 2011-02 Nick Tinnemeier (UU)
Work flows in Life Science

### 2011-03 Jan Martijn van der Werf (TUE)
Compositional Design and Verification of Component-Based Information Systems

### 2011-04 Hado van Hasselt (UU)
Insights in Reinforcement Learning: Formal analysis and empirical evaluation of temporal-difference learning algorithms

### 2011-05 Base van der Raadt (VU)
Enterprise Architecture Coming of Age - Increasing the Performance of an Emerging Discipline.

### 2011-06 Yiwen Wang (TUE)
Semantically-Enhanced Recommendations in Cultural Heritage

### 2011-07 Yujia Cao (UT)
Multimodal Information Presentation for High Load Human Computer Interaction

### 2011-08 Nieske Vergunst (UU)
BDI-based Generation of Robust Task-Oriented Dialogues

### 2011-09 Tim de Jong (OU)
Contextualised Mobile Media for Learning

### 2011-10 Bart Bogaert (UvT)
Cloud Content Contention

### 2011-11 Dhaval Vyas (UT)
Designing for Awareness: An Experience-focused HCI Perspective

### 2011-12 Carmen Bratosin (TUE)
Grid Architecture for Distributed Process Mining
2011-13 Xiaoyu Mao (UvT)
Airport under Control. Multiagent Scheduling for Airport Ground Handling

2011-14 Milan Lovric (EUR)
Behavioral Finance and Agent-Based Artificial Markets

2011-15 Marijn Koolen (UvA)
The Meaning of Structure: the Value of Link Evidence for Information Retrieval

2011-16 Maarten Schadd (UM)
Selective Search in Games of Different Complexity

2011-17 Jiyin He (UVA)
Exploring Topic Structure: Coherence, Diversity and Relatedness

2011-18 Mark Ponsen (UM)
Strategic Decision-Making in complex games

2011-19 Ellen Rusman (OU)
The Mind ’ s Eye on Personal Profiles

2011-20 Qing Gu (VU)
Guiding service-oriented software engineering - A view-based approach

2011-21 Linda Terlouw (TUD)
Modularization and Specification of Service-Oriented Systems

2011-22 Junte Zhang (UVA)
System Evaluation of Archival Description and Access

2011-23 Wouter Weerkamp (UVA)
Finding People and their Utterances in Social Media

2011-24 Herwin van Welbergen (UT)
Behavior Generation for Interpersonal Coordination with Virtual Humans
On Specifying, Scheduling and Realizing Multimodal Virtual Human Behavior

2011-25 Syed Waqar ul Qounain Jaffry (VU)
Analysis and Validation of Models for Trust Dynamics

2011-26 Matthijs Aart Pontier (VU)
Virtual Agents for Human Communication - Emotion Regulation and
Involvement-Distance Trade-Offs in Embodied Conversational Agents and Robots

2011-27 Aniel Bhulai (VU)
Dynamic website optimization through autonomous management of design patterns
Index

A/B testing, 198
action, 58
action space, 183
atomic observation unit, 68
effectiveness, 22
episodic search strategies, 18, 66
ethnography, 30, 73, 94
expectation, 56
exploration, 19
familiarity, 22, 55, 58
field research, 30
foreknowledge, 55, 58
goal, 13
hedonic search strategies, 18, 67
hypermedia navigation, 22
impulse behavior, 17
indirect social navigation, 20
involuntary browsing, 24
ISBN, 42n
ISSN, 42n
landmark, 20, 67
learning effect, 55, 58
lexical problem, 22
cognitive maps, 23
condition, 58
conjugate family of distributions, 209
cost, see reward
cost function, see reward function
criterium function, 186
hypermedia navigation, 22
design pattern, 114
direct social navigation, 19
dynamic programming, 186n
dynamic website optimization, 205
Bayes’ rule, 208
behavior
  navigation, 15, 21
  search, 21
  subjective search, 73
  triggered search, 88
Beta distribution, 209
boolean operator, 14
familiarity, 22, 55, 58
field research, 30
foreknowledge, 55, 58
goal, 13
hedonic search strategies, 18, 67
hypermedia navigation, 22
impulse behavior, 17
indirect social navigation, 20
involuntary browsing, 24
ISBN, 42n
ISSN, 42n
landmark, 20, 67
learning effect, 55, 58
lexical problem, 22
maps, 20
Markov decision problems, 179, 186, 187
MDPs, see Markov Decision Problems
mental activity, 22
multi-armed bandit problem, 207
Multivariate Testing, 199
MVT, see Multivariate Testing

natural search behavior, 88
navigation interface, 23
navigational aids, 20
note taking, 32
notes, 32
   full field notes, 32
   jotted notes, 32
   mental notes, 32

object identification, 19
observation, 29, 61, 85
   natural settings, 44–53
   task based, 40–44
observation session, 35
opportunistic browsing, 24
optimality equation, 186, 187

participating observation, 30
pattern, 56
pilot study, 29–59
posterior distribution, 209
prior distribution, 209
problem space, 183
raters, 68, 92
reference point, 67
reward, 184
reward function, 185
RSS, 156
scanning, 55, 58
search behavior
   triggered search behavior, 62
search behavior, 29, 61, 85
   natural search behavior, 62
search browsing, 24
search strategy, 22
search tactic, 22
search unit, 93
semantic problem, 22
signs, 20
social navigation, 19
spatial ability, 22
spatial cognition, 22
standard, 56
state
   current, 184
   next, 184
state triggers, 59, 67, 82, 86
strategy, 186
subjective search behavior, 73
summary observation unit, 70
syntactic problem, 22

   task description, 35
   top-level domain, TLD, 149
   transition law, 184
   triggered search behavior, 88
Universal Resource Locator, 23
URL, see Universal Resource Locator
usability, 24
user experience, 24
user interaction design pattern, see design pattern

W3C, see World Wide Web Consortium
wayfinding, 18
web visitor perception, 192
website optimization, 4
wild searching, 2
World Wide Web, 2, 85
World Wide Web Consortium, 128
WWW, see World Wide Web