# A Bandit's Perspective on Website Adaptation

Benoit Baccot[1,2], Vincent Charvillat[1], and Romulus Grigoras[1]

[1] University of Toulouse, IRIT-ENSEEIHT, 2 rue Camichel, 31071 Toulouse, France
[2] Sopra Group, 1 Avenue André Marie Ampère, 31772 Colomiers, France

**Abstract.** Ubiquitous access to websites stresses the importance of adaptation for modern websites. The design and management of decision-taking adaptation engines has become a major research challenge. This paper proposes a bandit-based adaptation decisional model and shows how to describe and manage adaptation policies using a lightweight XML-based description language. The model allows a web marketeer to easily design and deploy adaptation policies. Experimental results on a real website show the effectiveness of our model.

## 1 Introduction and Problem Statement

A tremendous amount of information is available online. Web sites provide a wealth of services, ranging from information broadcast, to electronic commerce or on-line learning. Web sites have become increasingly complex and are now facing an important challenge: ubiquitous access. Indeed, users reach the web from diverse contexts (from a desktop, on the go etc.), and use a variety of terminals and networks. As humans, users have diverse expectations and behaviors while accessing online services. Providing users with the best experience raises many challenges that are being addressed by web-site and, more generally, multimedia adaptation [1–4]. The user is arguably the most important component of the environment. Therefore, recently a lot of research has focused on user-aware multimedia adaptation.

On the web, marketeers are working hard on increasing the effectiveness of web sites. This means, for information sites, making information easily browsable and provide users with information tailored to their needs. For commercial sites, this means increasing the match between the products or services the user is looking for and their characteristics, quality and quantity. Recommendation or advertising related products on a commercial web site is an example of an adaptation tool that a web marketeers can use [5–8]. Generally speaking, web marketeers aim at optimizing the effectiveness of a given website by taking into account the population of users, the website content and the effectiveness criteria (e.g. sales maximization). They also need to have a measure of impact that provides feedback on the quality of the adaptation strategy (called adaptation policy). This paper proposes a system that helps web marketeers to design and deploy adaptation policies, that can range from fully static (statically defined beforehand and applied without change to a whole group of users) to very dynamic (evolving with time and/or finely tailored to individual users).

A natural way of expressing information about adaptation policies is to model them as website metadata. In this work we propose flexible, XML data structures that allow for easy management of website adaptation policies. The associated policies we introduce in this paper are derived from a simple decisional technique (the so-called Bandit problem).

The next section lays the basis of this work. Our Bandit based framework is presented in Section 3 while Section 4 details a practical implementation. We consider a real-life adaptation problem dealing with optimal delivery of richmedia banners. Section 5 concludes the paper and brings avenues for future work.

## 2 An adaptation architecture

### 2.1 Metadata and decision-taking

Achieving interoperable access to distributed richmedia content by shielding users from network and terminal heterogeneity starts with a formal description of the delivery context. Since this description must be interoperable in itself, the role of metadata standards is prominent [9, 10] . Many standards exist. The CC/PP (Composite capabilities/Preference Profiles), the MPEG-21 UED (Usage Environment Description) or the DCO Delivery Context Ontology are compared in the survey of Timmerer and al. [11].

Describing the context is necessary but not sufficient for deciding how to perform adaptation. The decision-taking is a key component for context-aware adaptation [1] and many decisional models have been devised [2, 3]. Picking up a feasible solution is very different than selecting the optimal adaptation decision. Figure 1 introduces a possible architecture for adapting the content in an optimal way.

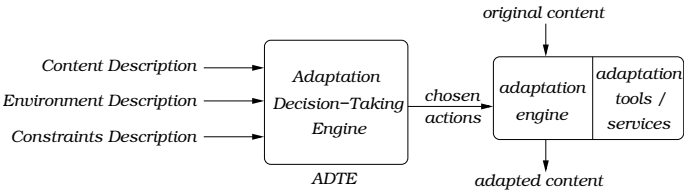


**Fig. 1.** Decision-taking agent

The adaptation decision-taking engine (ADTE [3]) takes a context delivery descriptor as input. This input descriptor is threefold in figure 1 and divided in content, environment and constraints sub-descriptions. The ADTE outputs a decision that is forwarded to the adaptation engine (AE). The actual transformation of the content is performed thanks to available services on the AE.

### 2.2 Handling dynamic environment in closed-loop

The previous architecture can be generalized in highly-varying delivery context. In this case many context features vary dynamically: both the available resources and the user intentions may change at any time. The decisional agent, like the ADTE, must dynamically react to context variations.

An even more sophisticated solution would be to use a learning agent that should itself learn from its interactions with the context and improve gradually. We have already proposed a closed-loop approach in that respect [4]. The main idea is to add a feedback channel to control the open-loop architecture shown in Figure 1. In this approach, we consider an adaptation agent and its environment. This environment abstractly integrates the multimedia content, the user, his mobile terminal, the available network and so on. The three steps of the closed-loop are the following:

- **Perception.** At each instant, the decision agent perceives (at least partially) the current characteristics of the delivery context. The current state of the agent in its environment is built upon these perceived characteristics.
- **Action.** Among various possible adaptation decisions it can make in a given state, a learning agent tries to choose the best action to generate as output (in line with Figure 1).
- **Feedback.** The action changes the state of the environment (the adaptation does influence the user, the subsequent resources, etc.) and the value of this transition is communicated to the agent through a reward (a scalar "reinforcement signal"). The agent policy should choose actions that tend to increase the long-term summation of rewards. It can learn to do this by reinforcing decisions that resulted in good accumulation of rewards and, conversely, by trying to avoid unfruitful decisions.

Such an agent learns over time by reinforcement (or by trial and error) [12]. A difficulty is that the agent must explicitly *explore* its environment to estimate the utility of taking actions in all reachable states. Intuitively each state must be visited, each action must be evaluated before converging to the best policy that maps states to optimal actions. There is then a fundamental trade-off between exploration and exploitation. The dilemma the agent faces at each trial is between "exploitation" of the current "best action" that has the highest expected payoff and "exploration" to get more information about the expected payoffs of the other actions.

### 2.3 Issues

Formally, a learning agent will be defined by a discrete set of environment states, a discrete set of adaptation actions and a reward function that outputs a value for an action in a given state. The main issue is to model a real-life problem using this formal ingredients. Crucial choices must also be made with respect to the problem structure. Either we have an independent problem for each state or we do not. In the latter case, the transition rules between states must be modeled or learnt.

## 3 A Multi-Armed Bandit Solution

As explained above, the reinforcement learning approach fits well with dynamic multimedia adaptation. Indeed, we applied this framework to several problems, such as ubiquitous streaming [4] and user-aware adaptation [13, 14]. Until now, we explicitly took into account the relations and the transitions between decisional states. This led us to model our adaptation problems with so-called Markov Decisional Processes [12]. In this paper, we assess a simpler decisional model: the multi-armed Bandit problem. This model, although less general, is expected to be much easier to use in practice. In particular we show that it can be easily declared in a simple XML format.

### 3.1 A unified decisional state

We first capitalize on previous ideas and introduce a unified decisional state. A state of -(the agent in)- the delivery context is a triplet which integrates:

- some current observations,
- some inferred information,
- an amount of memory.

The current observations are factual elements that allow to describe unambiguously the adaptation target and other features of the adaptation problem. The website content to be adapted, the associated URL, the time within current web session, the characteristics of the terminal may be parts of these observations [4, 14]. The inferred information are composed of partially observable metadata. We called them "subjective semantic descriptors" in [10] (by contrast with "objective" observations). The user "interest level", the "importance" of a given media are contextual metadata that can only be approximately inferred. The memory of the decisional state finally allows to satisfy a self contained property. Intuitively, the aim is to retain all relevant information from the past. If a given adaptation has already been performed, the idea is to adapt the current content in different way. Memorizing previous decisions helps to do this.

It is now straightforward to argue for the use of a Multi-Armed Bandit Problem with such a decisional state.

### 3.2 The Bandit and the adaptation actions

A Multi-Armed Bandit Problem (MABP) is named by analogy to a slot machine. For example, in the $K$-Armed Bandit Problem, a gambler has to choose which of the $K$ slot machines to play. At each time step, he pulls the arm of one machine and receives a reward. His aim is to maximize the sum of rewards he perceives over time. This clearly shows the "exploration vs exploitation" dilemma: the purpose of the gambler is to find, as rapidly as possible, the arm that gives the best expected reward.

In order to solve a Bandit problem (i.e. to find the best arm to play), various strategies can be used [12]. Recent research has proposed various solutions to

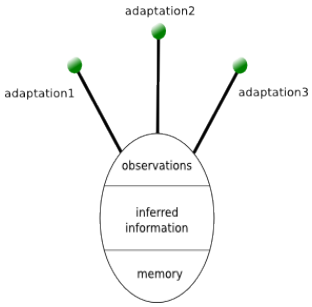


**Fig. 2.** A decisional state (depicted as an ellipse, containing the information triplet describing the context) and the associated 3-armed Bandit problem (adaptation 1, 2 or 3 can be performed).

solve optimally and online this dilemma, minimizing the number of errors over time. One of the most efficient is called Upper Confidence Bound (UCB, [15]). At each play, it computes a priority index for each arm, based on the previous rewards and the number of times it has already been invoked. The index $p_j$ for arm $j$ is given by

$$p_j = \bar{x_j} + \sqrt{\frac{2ln(n)}{n_j}} \tag{1}$$

where $\bar{x_j}$ is the average rewards obtained from arm $j$, $n_j$ the number of times arm $j$ has been chosen and $n$ the overall number of plays done so far. The best arm to choose is the one with the highest priority.

For our adapation problem, we use MABPs as follows:

- we strategically define a set of decisional states,
- we associate one MABP to each state of the context,
- each arm corresponds to a possible adaptation action in a given state,
- rewards are given by an impact/utility measure.

Figure 2 depicts a 3-armed Bandit problem associated to a state. Pushing arm $i$ at a play means performing adaptation $i$ at this step.

As a result, we get a collection of independent multi-armed Bandit problems. Each MAPB can be solved independently using the UCB technique, at the expense of neglecting potential relations between states.

### 3.3 Using a declarative language

The model we propose is really simple, and only states and their associated Bandit have to be defined. Thus, we propose a XML-based language that allows web marketeers to easily design and deploy adaptation policies.

The first thing we have to do is to define and declare a state. A state (figure 3), as said earlier, is composed of three parts: observations, inferred information

and memory. Observations are described as a set of observable elements (current page, terminal information, etc.), memory as a set of past taken actions. For inferred information, a handle (e.g. a an Uniform Resource Identifier (URI) to a service able to produce this information by inference (e.g. by analyzing the logged behaviors of website visitors) must be provided.

```
<state id="s1">
  <observations>
    <observation>terminal</observation>
  </observations>
  <inferences>
    <inference>user-activity</inference>
  </inferences>
  <memory>
    <action>banner3</action>
  </memory>
  <bandit-ref>bs1</bandit-ref>
</state>
```

**Fig. 3.** An XML description of a state ($s1$)

Then, we need to describe the possible adaptations in this state, that is to say the MABP associated to the state. A MABP (figure 4) contains a set of actions. Each action is composed of a handle to the corresponding adaptation engine and the number of times it has been invoked. This last number allows an ADTE to compute the priority index given by UCB (equation 1) in order to choose a correct action. A reward handle is also given in order to measure the impact (or utility) of the chosen action.

```
<bandit id="bs1">
  <actions>
    <action>
      <engine>banner3-injector</engine>
      <nbInvok>0</nbInvok>
    </action>
    ...
  </actions>
  <reward>session-duration</reward>
</bandit>
```

**Fig. 4.** An XML description of a Bandit problem associated to state $s1$

### 3.4 Software Architecture

Based on the previous XML description of our model, we propose in figure 5 a software adaptation architecture.

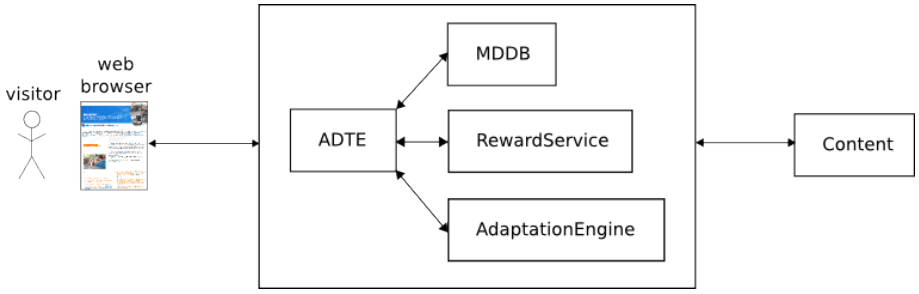


**Fig. 5.** The software architecture

A visitor is browsing a website using his favorite browser. While browsing the site, he is monitored by the *ADTE*, that observes the usage and builds a state using direct and inferred information and past adaptation actions. Our observation platform has already been presented in our community [16]. When a new state is computed, the ADTE asks *the metadata database (MDDB)* and eventually gets an associated Bandit problem. If so, it computes the priority indexes using UCB (equation 1) and chooses the action to perform. This action is then transmitted to *the adaptation engine* that actually performs it. Finally, using *the reward service*, it gets a reward qualifying the degree of success of the taken adaptation decision. The ADTE updates the Bandit information (mean reward, number of times an action has been played) and sends it to the MDDB. Thus, the updated information will be used for next visitors who would be in this state.

## 4 Case study

In this section, we describe how to use our adaptation model on a real website[3]. The site we choose is a collaborative website and it is designed to be used in a professional environment. It is organized in different workspaces, composed of various sections (blogs, wikis, portal, file sharing, etc.), allowing information to be produced and shared by company employees.

### 4.1 User aware banner service

Adaptations on this website consist in adding personalized banners that recommend "hot" parts of the site to users (e.g. a new blog entry, a modified wiki page or an uploaded file).

This adaptation problem can be seen in two dimensions: the content to recommend and the format of the banner. In this study, we intentionally put aside the banner content production issue and consider it, as other authors (e.g. [8]),

[3] http://www.linkforus.org

as a separate question. As a result, we choose to use the existing RSS feed as the content provider for recommendations. Concerning the format of the banner, three types of banner formats are available (figure 6):

- the basic version, only composed of a text (including links to recommended content),
- the video/avatar version. In that case, an avatar in a video serves as a teaser,
- the 3D version. It uses a "carousel" component written in Adobe Flash. Recommendations are included in the different facets.

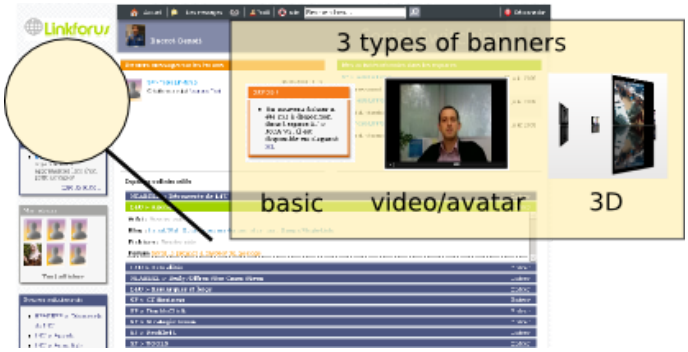


**Fig. 6.** The different types of banners

In line with the work of [17], we choose to display banners in sequence. A sequence contains exactly one banner of each format. Therefore, six sequences are available (basic/video/3D, video/basic/3D, etc.). Thus, adaptations will consist in displaying the banner in a certain order.

Among many possibilities, we chose to use as an objective/target of the adaptation to get users stay longer on the site, i.e. increase their session duration. We naturally use the session duration as an impact measure (reward) for a given sequence.

### 4.2 A simple MABP instance

We have to set the different states and the associated Bandit problems.

For the state, the simplest solution is to use a single state. It only contains an inferred information: the user "activity" (i.e. the number of events produced by the user in a given time window) on the website. When the user activity decreases, a banner is displayed on the site according to the chosen sequence of banners.

As for the associated Bandit problems, actions are the different sequences to display, and thus we consider a 6-armed Bandit problem. The (stochastic) reward is given by the session duration.

### 4.3 Results

In order to get validation, we use a navigation simulator that has been seeded with the data collected from a previous work ([17]) in which we also have sequences of three similar banner types. We have already concluded which sequence of banners is the best (video/3D/basic). Using this "ground truth", we want to determine whether the Bandit problem rediscovers or not this conclusion.

Figure 7 presents the evolution of priority indexes values for each arm of the associated Bandit. As the number of times the Bandit is invoked gets higher, priority indexes decreases. However, while zooming, we notice that the one for sequence video/3D/basic (bold black line) is often greater than the others. It means that the associated arm is pulled more frequently. Using the ground truth, we realize that, indeed, this adaptation action is better that the others with respect to the session duration.

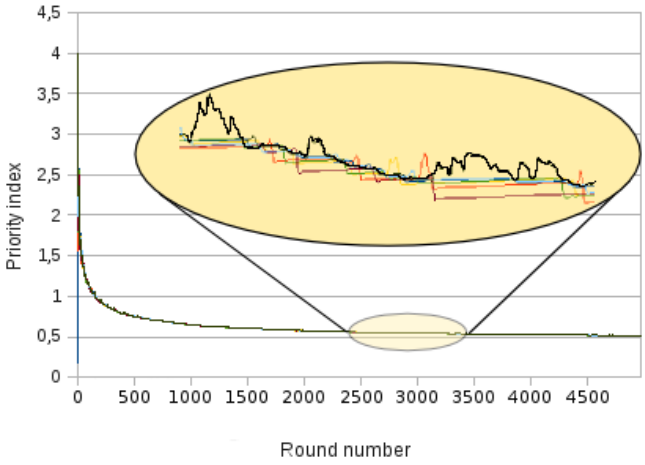


**Fig. 7.** Evolution of priority indexes for each arm of the Bandit problem.

Figure 8 shows the percentage of the optimal action the Bandit has been chosen in function of the number of plays. Let us recall that we know which action is the best thanks to the original dataset. As the number of plays gets higher, the percentage increases, indicating the efficiency of the Bandit strategy. Interesting results are reached after around 10.000 sessions. On our test website, this can be realized in less than a month.

Bandit problems allow us to draw a conclusion similar to [17]: the usefulness of considering sequence for improving the effectiveness of banners. Results show that an improper format of a banner in the beginning of the sequence wipes out the positive effects of the subsequent banners. These results demonstrate the power and simplicity of a Bandit-based adaptation strategy.

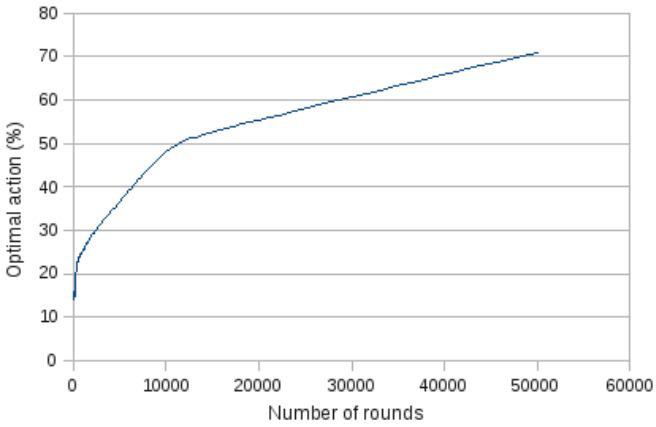


**Fig. 8.** Percentage of optimal action chosen

## 5 Conclusion

Decision-taking engines are an important component of adaptive web sites. To help in designing effective adaptation systems, this paper contributed with a bandit-based model for the website adaptation problem. We proposed a declarative, XML-based language for expressing and managing a multi-armed bandit decision model. In order to experimentally validate our model, we presented a case study that shows the use of the model on a real website. Our model proves to be a simple, lightweight decisional model. It provides finely tunable yet simple to use adaptation policies.

Our model gives web marketeers a great flexibility while managing a policy. It is possible to choose/concentrate on a limited number of states (the state space is virtually very large). On these selected states, it is possible to configure the system to only use a selected/useful subset of all possible adaptation actions.

The perspectives of this work are twofold. First we would like to investigate networks or trees of bandit problems. This would be a possible solution to partially relate a given decisional state to subsequent ones while avoiding the complexity of Markov Decision Processes. Secondly, we believe that the Upper Confidence Bound (UCB) algorithm, when applied to trees (the so-called UCT), would be a good candidate to solve our adaptation problems in that case.

## References

1. Mukherjee, D., Delfosse, E., Kim, J.G., Wang, Y.: Optimal adaptation decision-taking for terminal and network quality-of-service. IEEE Transactions on Multimedia **7**(3) (2005) 454–462

2. López, F., Martínez, J.M., Valdés, V.: Multimedia content adaptation within the cain framework via constraints satisfaction and optimization. In: Adaptive Multimedia Retrieval. (2006) 149–163
3. Jannach, D., Leopold, K., Timmerer, C., Hellwagner, H.: A knowledge-based framework for multimedia adaptation. Appl. Intell. **24**(2) (2006) 109–125
4. Charvillat, V., Grigoras, R.: Reinforcement learning for dynamic multimedia adaptation. J. Network and Computer Applications **30**(3) (2007) 1034–1058
5. Pandey, S., Olston, C.: Handling advertisements of unknown quality in search advertising. In: Twentieth Annual Conference on Neural Information Processing Systems (NIPS). (2006)
6. McCoy, S., Everard, A., Polak, P., Galletta, D.F.: The effects of online advertising. Commun. ACM **50**(3) (2007) 84–88
7. Attenberg, J., Pandey, S., Suel, T.: Modeling and predicting user behavior in sponsored search. In: KDD. (2009) 1067–1076
8. Hauser, J.R., Urban, G.L., Liberali, G., Braun, M.: Website morphing. Marketing Science **28**(2) (2009) 202–223
9. Kosch, H., Böszörményi, L., Döller, M., Libsie, M., Schojer, P., Kofler, A.: The life cycle of multimedia metadata. IEEE MultiMedia **12**(1) (2005) 80–86
10. Lux, M., Granitzer, M., Spaniol, M., eds.: Multimedia Semantics - The Role of Metadata. Volume 101 of Studies in Computational Intelligence. Springer, Berlin (August 2008)
11. Timmerer, C., Jabornig, J., Hellwagner, H.: Delivery context descriptions - a comparison and mapping model. In: Proceedings of the 9th Workshop on Multimedia Metadata (WMM'09). (2009)
12. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press (1998)
13. Plesca, C., Charvillat, V., Grigoras, R.: A formal framework for multimedia adaptation revisited: a metadata perspective. In: BTW Workshops. (2007) 160–178
14. Plesca, C., Charvillat, V., Grigoras, R.: Adapting content delivery to limited resources and inferred user interest. International Journal of Digital Multimedia Broadcasting **2008** (2008) doi:10.1155/2008/171385
15. Auer, P., Cesa-Bianchi, N., Freund, Y., Schapire, R.E.: The nonstochastic multi-armed bandit problem. SIAM Journal on Computing **32**(1) (2003) 48–77
16. Baccot, B., Charvillat, V., Grigoras, R., Plesca, C.: Visual attention metadata from pictures browsing. In: Ninth International Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS'08. (May 2008) 122–125
17. Baccot, B., Choudary, O., Grigoras, R., Charvillat, V.: On the impact of sequence and time in rich media advertising. In: Proceedings of the 17th ACM Conference on Multimedia. (2009)