Adaptive Portals: Adapting and Recommending Content and Expertise

Andreas Nauertz and Martin Welsch
IBM Research and Development
71032 Böblingen, Germany
{andreas.nauertz|martin.welsch}@de.ibm.com

Fedor Bakalov and Birgitta König-Ries
University Jena
07743 Jena, Germany
{fedor.bakalov|koenig}@informatik.uni-jena.de

Abstract

Today, Portals provide users with a central point of access to companywide information. Initially they focused on presenting the most valuable and widely used information to users providing them with quick and efficient information access. But the amount of information accessible quickly grew and finding the right information became more and more complex and time consuming. In this paper, we illustrate options for adapting and recommending content based on user- and context models that reflect users’ interests and preferences and on annotations of resources provided by users. We additionally leverage the entire communitys’ interests, preferences and collective intelligence to perform group-based adaptation. We adapt a Portal’s structure (e.g., navigation) and provide recommendations to be able to reach content being of interest easier. We recommend background information, experts and users with similar interests. We finally construct a Portal’s navigation structure entirely based on the communitys’ behavior. Our main concepts have been prototypically embedded within IBM’s WebSphere Portal.

1 Introduction

In recent years Enterprise Information Portals have gained importance in many companies. As a single point of access they integrate various applications and processes into one homogeneous user interface. Today, typical Portals contain thousands of pages. They are no longer exclusively maintained by an IT department, instead, Web 2.0 techniques are used increasingly, allowing user generated content to be added to Portal pages. This tremendous popularity and success of Portals, has its downsides: Their continuous growth makes access to really relevant information difficult. Users need to find task- and role-specific information quickly, but face information overload and feel lost in hyperspace. The huge amount of content results in complex structures designed to satisfy the majority of users. However, those super-imposed structures defined by Portal authors and administrators are not necessarily compliant to the users’ mental models and therefore result in long navigation paths and significant effort to find the information needed. This becomes even worse, once user generated content is added, where the structure may not follow the design the administrator had in mind. In addition, the more content a Portal offers, the more likely it becomes that users are no longer aware of all the resources available within it. They might thus miss out on resources that are potentially relevant to their tasks, simply because they never come across them. Thus, on the one hand, users obtain too much information that is not relevant to their current task, on the other hand, it becomes cumbersome to find the right information and they do not obtain all the information that would be relevant.

In this paper, we therefore propose steps towards the next generation of Portals: Portals, that are adaptive and context-aware. Instead of providing all possible information, only those should be presented which are relevant in the user’s current needs. To be more precise, we want Portals that

- are able to dynamically adapt their structure, such as the navigation and page structure to better suit users’ needs. These adaptations can be done automatically or can be used to issue recommendations to the user.

- automatically provide additional in-place, in-context, background information on information pieces the user is interested in. For instance, a reference to a place could be supplemented by the Google maps view on that place or a user could be given information about resources with similar content to the one they are just viewing.

- automatically provide links to help if users get lost or are unfamiliar with something. For instance, a user struggling to fill out a form for the first time could be directed to a colleague that frequently uses this specific resource.

So, what do we need to achieve adaptivity and context-awareness? First of all, information on the available resources, the users and their behavior is required. Second, this information needs to be exploited to adapt the Portal. In the following sections, we are first taking a closer look at what information it is exactly that we need and - maybe even more importantly - how this information can be obtained. We will show that a mixture of automated extraction and user input is the most realistic approach here for the time being. We will then explore possibilities to use the information to adapt the Portal in a number of different ways. Finally, we will provide some insights into the results of the evaluations we have carried out so far and into the future work that we intend to perform. Before all this, however, we will give an overview of related work.

2 Related Work

A lot of research has been done in the field of adaptive hypermedia [Brusilovsky, 2001], systems that build and apply
3 Information about Users, Behavior, and Resources

From a conceptual point of view (cp. fig. 1) Portals are comprised of various resources such as pages and portlets (artifacts residing on pages delivering content). These resources are arranged based on Portal models, often initially created by some administrator with the aim to satisfy the majority of all users and not the preferences of each single user. We therefore need information about individual users (or groups of users) and their behavior as a basis for adaptation. We apply different techniques such as web mining to construct user models reflecting users interests and preferences; we use information from their static profile (native language, home country, working location, age, etc.), their interaction behavior (pages and portlets they work with; tags they apply to resources), and their social network to derive knowledge about their needs. We observe the context (date, time, location, ...) in which they interact to partition the user model in so called profiles like private or business. Additionally, we need enriched information about the resources available in the system. We illustrate how we extract information pieces of certain type in order to provide background information by connecting to external sources and to interlink them in order to issue recommendations.

3.1 Extracting Information about Users

User Model

In order to perform reasonable adaptations or to provide users with recommendations we need to understand users’ interests and preferences. Therefore we construct user models reflecting their behavior. We use static information from users’ profiles (describing their age, native language, etc.), as well as dynamical information which we retrieve via web usage mining.

Web Mining [Liu, 2006] is the application of data mining techniques to discover (usage)-patterns within web data. Web usage mining is the extraction of usage patterns from access log data modeling certain aspects of the behavior of users (or the entire community). Our system has been incorporated into IBM’s WebSphere Portal. Analyzing its logs reveals information about, among other things, several events, e.g. when pages (or portlets) are created, read, updated or deleted, when pages (or portlets) are requested, when users are created, updated, deleted and many more.

Analyzing the log allows to understand which pages and portlets a user typically works with. Obviously, the user model must allow the calculation of the utilization of pages and portlets from the historical data available. We do this by measuring how often a user interacts with certain pages and portlets. Of course, we also consider interactions that occurred recently to be more important than interactions that occurred in the past and we hence apply time-weighting factors when calculating the utilization of pages and portlets based on the target hits they received.

More generally, we apply techniques from the area of frequent set mining [Liu, 2006] to analyze the usage of pages and portlets. We use the Apriori algorithm [Agrawal and R., 1994], a standard association rule mining algorithm, to determine items, such as pages and portlets that co-occur frequently. We apply the GSP algorithm [Srikant and Agrawal, 1996], a standard sequential pattern mining algorithm, to determine sequences of items, such as pages and portlets, that co-occur frequently. Comparing the itemsets even allows to find users behaving similarly.

Tagging Behavior Analysis. We additionally analyze users’ tagging behavior to understand both, single users’ as well as the entire communities’ interests and preferences.

user and usage models to adapt web sites to the user’s context (interests, preferences, needs, goals, etc.). One possible approach to derive those models and enable adaptation is to analyze user access data, as Perkowitz and Etzioni [Perkowitz and Etzioni, 1997] propose. Projects in this context include PageGather [Perkowitz and Etzioni, 2000], Letizia [Lieberman, 1995] and WebWatcher [Joachims et al., 1997]. Especially with respect to navigation adaptation Smyth and Cotter [Smyth and Cotter, 2003] describe an approach to speed up navigation in mobile Portals significantly.

Providing background information or interlinking information pieces is based on the ability to either allow users or programmatic, automated, annotators to annotate information pieces. We have described the first approach in [Nauerz and Welsch, 2007] already. The second approach is based on information extraction from unstructured machine-readable documents. Although the approach to perform the extraction is often differing, most papers in this area regard information extraction as a proper way to automatically extract semantic annotations from web content. Most of these systems are based on machine learning techniques, e.g. [Dill et al., 2003].

Generally, regarding the recommendation of expertise, systems that help to find experts are called expertise finders or expertise location engines [Zhang and Ackerman, 2000]. A general architecture for recommendation systems that allow locating experts is described in [McDonald and Ackerman, 2000]. More specifically Streeter et al. present who knows, a system which recommends experts having knowledge in specific topics based on profiles created from observing the documents they have selected and worked with previously [Streeter and Lochbaum, 1988]. Newer systems that use information about social networks to find experts are e.g. [Kautz et al., 1997].

Collaborative ranking, i.e. ranking which takes into consideration entire communities’ interests, has recently become more important. Access patterns are used to assess the importance of single web pages [Caverlee et al., 2006]. Improved versions of the original PageRank [Page et al., 1998] and HITS [Kleinberg, 1998] algorithms have been developed (cp. FolkRank [Hotho et al., 2006], CollaborativeRank [Michaill, 2005]). So far, all these algorithms have mainly been used to improve the ranking of search results returned by search engines as response to users’ queries. We will use the ideas underlying collaborative ranking to calculate recommendations and even to dynamically adapt Portal structures to better suit single users’ or entire communities’ needs.

Other work focuses on personal recommendation of content based on its relatedness to certain tag terms. [Wu et al., 2006] propose a modified version of the HITS algorithm to determine experts and high-quality documents related to a given tag. Tagging systems allow not only recommending content, but also users knowledgeable in certain areas. Based on metrics like ExpertRank [Farrell and Lau, 2006], these users could be recommended and searched. In contrast to the HITS based approach we utilize an improved metric to determine related resources.

3 Information about Users, Behavior, and Resources

From a conceptual point of view (cp. fig. 1) Portals are comprised of various resources such as pages and portlets (artifacts residing on pages delivering content). These resources are arranged based on Portal models, often initially created by some administrator with the aim to satisfy the majority of all users and not the preferences of each single user. We therefore need information about individual users (or groups of users) and their behavior as a basis for adaptation. We apply different techniques such as web mining to construct user models reflecting users interests and preferences; we use information from their static profile (native language, home country, working location, age, etc.), their interaction behavior (pages and portlets they work with; tags they apply to resources), and their social network to derive knowledge about their needs. We observe the context (date, time, location, ...) in which they interact to partition the user model in so called profiles like private or business. Additionally, we need enriched information about the resources available in the system. We illustrate how we extract information pieces of certain type in order to provide background information by connecting to external sources and to interlink them in order to issue recommendations.

3.1 Extracting Information about Users

User Model

In order to perform reasonable adaptations or to provide users with recommendations we need to understand users’ interests and preferences. Therefore we construct user models reflecting their behavior. We use static information from users’ profiles (describing their age, native language, etc.), as well as dynamical information which we retrieve via web usage mining.

Web Mining [Liu, 2006] is the application of data mining techniques to discover (usage)-patterns within web data. Web usage mining is the extraction of usage patterns from access log data modeling certain aspects of the behavior of users (or the entire community). Our system has been incorporated into IBM’s WebSphere Portal. Analyzing its logs reveals information about, among other things, several events, e.g. when pages (or portlets) are created, read, updated or deleted, when pages (or portlets) are requested, when users are created, updated, deleted and many more.

Analyzing the log allows to understand which pages and portlets a user typically works with. Obviously, the user model must allow the calculation of the utilization of pages and portlets from the historical data available. We do this by measuring how often a user interacts with certain pages and portlets. Of course, we also consider interactions that occurred recently to be more important than interactions that occurred in the past and we hence apply time-weighting factors when calculating the utilization of pages and portlets based on the target hits they received.

More generally, we apply techniques from the area of frequent set mining [Liu, 2006] to analyze the usage of pages and portlets. We use the Apriori algorithm [Agrawal and R., 1994], a standard association rule mining algorithm, to determine items, such as pages and portlets that co-occur frequently. We apply the GSP algorithm [Srikant and Agrawal, 1996], a standard sequential pattern mining algorithm, to determine sequences of items, such as pages and portlets, that co-occur frequently. Comparing the itemsets even allows to find users behaving similarly.

Tagging Behavior Analysis. We additionally analyze users’ tagging behavior to understand both, single users’ as well as the entire communities’ interests and preferences.
Tagging - the process of assigning tags to objects - has become a popular technique to describe, organize, categorize and locate resources. A tag is a (relevant) keyword or term associated with or assigned to a piece of information, thus describing the item and enabling keyword-based classification of information. Our concept allows users to annotate uniquely identifiable resources of a Portal, such as pages, portlets, and even other users.

Tagging behavior analysis is based on the assumption that tagging expresses interest in a resource. Hence, resources being tagged more often by a user are of higher importance to him. And since tagging is a collaborative process we can also assume that resources being tagged more often by all users are of higher importance to the entire community. Thus, analyzing users’ tagging behavior allows us to better understand both, single users’, as well as the entire community’s interests and preferences.

A second assumption is that different tags being used in the system are semantically related. This means that they have a different semantic distance which can be calculated. Generally, if the same two tags $T_1$ and $T_2$ are applied to the same resources $R_1 \ldots R_n$ often, they often have a small semantic distance, or, in other words are strongly semantically related. Understanding the semantic relation between tags we can perform various adaptations and recommendations, e.g., reorder pages in the navigation hierarchy or recommend related content to users based on their current selection.

A third assumption is, that analyzing and comparing the tagging behavior between all users allows partitioning them into groups of “similar behavior”. Users within the same “behavioral cluster” can be provided recommendations and adaptations based on what a major subset of other users being part of the same cluster have already done.

Finally, by analyzing and comparing users’ tagging behavior we can determine experts for certain (content) areas. We can assume that a user tagging certain resources has knowledge about how to deal with them. Hence, we can recommend this user to the other users as an expert knowledgeable about the tagged resources.

Social Network Analysis. Finally, the analysis of users’ explicit contacts allows to determine users’ interests and preferences, too. The assumption is that the fact that users directly know each other can be an indication for similar job roles and hence for sharing similar knowledge.

Context Model

Focusing on user models only neglects the context users are acting in. Hence, these could be regarded suitable models, only, if the role, the interests and preferences of users will not change too much over time. In reality, needs usually change if a user’s context changes. For example, a user who is in the process of planning a business trip will need resources that provide information about hotels, rental cars, and flights. When the same user returns to his tasks as a project manager, he will need a completely different set of resources. Of course his interests and preferences will be totally different in both roles and obviously he needs access to totally different resources (pages, portlets, etc.).

The analysis of users’ tagging behavior can even be used to evaluate users’ context and to determine resources being of special interest in certain contexts.

Generally we can analyze how tags are applied in correlation to values of certain context attributes. For instance, we can analyze when (date and time) certain tags are applied. As an example, if a user applies the tag private only on Saturdays and Sundays we can assume that resources tagged with this tag are of special interest on these days only. Alternatively we can analyze which device is used when certain tags are applied. E.g., if a user applies the tag traveling only if using his PDA we can assume that resources tagged with this tag are of special interest when using this device.

Vice versa, we can analyze tags that already have been assigned to resources being used to determine and switch the context. E.g., if a user starts to use resources mainly tagged private we might want to switch to the corresponding context profile.

Our solution allows single users to have several context profiles between which either the system switches automatically, based on context attributes being observed (date, time, location, etc.), or the user switches manually. New profiles can be defined using a profile management portlet which allows to specify the initial settings of a profile (which theme to use, which skin to use, etc.) and to associate it with a set of context attributes (date, time, location, etc.) which define when it should become active.

Figure 1: Conceptual overview
Our adaptation and recommendation components utilize both the information stored in the user and context model, to perform their operations (i.e. to adapt structures such as the navigation). Technically, the user model is partitioned in a separate partition for each context profile available in the context model. To determine the best matching profile, the system permanently observes a set of defined context attributes. Users always have the option to outvote the system's decision and to manually switch to another profile.

As only one context profile can be active at one specific point in time, whatever people do only influences the user model partition associated to the currently active profile. For example, if the currently active profile is trip planning, the navigation behavior will have no effect on the user model partition associated with the profile project management.

3.2 Extracting enriched Information about Resources
To extract enriched information, which we need e.g. to recommend background information about the resources, we currently allow for the usage of three different mechanisms:

Automated Tagging. Here the system analyzes markup generated by the Portal to find occurrences of identifiable information pieces of certain types such as persons, locations, etc., and wraps these into semantic tags. We have integrated the UIMA framework \(^1\) and written customized analysis engines able to identify such information pieces.

Semi-automated Tagging If the system cannot unambiguously identify the type of an information piece it still allows users to mark it and tell the system of what type it is. We call this process semi-automated tagging. For instance, if we find a fragment 'Paris H. was sighted leaving a Hotel in Paris' it becomes difficult for the system to determine whether Paris is a name or a location. The user can then mark the corresponding information pieces and tell the system their type. The information pieces are then wrapped into a semantic tag exactly as outlined before.

Manual Annotating Moreover, our system allows semantically tagged information pieces to be annotated manually again. For example, if the name of three persons Alice, Bob, and Charly often appear somewhere in the Portal system, e.g. in blog- or wiki portlets, our system automatically determines these fragments to be of type person, wraps them into semantic tags and allows for advanced interaction with these information pieces. Our tag engine allows these enriched fragments to be annotated e.g. with the term project-x which indicates that all three persons are somehow related to this project. This means that the options for manual annotating allow for an even more fine-granular categorization of information pieces.

4 Exploiting the Models for Adaptation and Recommendation
Now that we have described which information about the Portal resources and users are available to our system, we can explain how this information is used to improve the user experience with the Portal. We propose methods to adapt the content, to recommend content, to offer additional information and to recommend experts. In the following, more details about the approach are given.

4.1 Adapting the Portal Structure
Within the context of this project we have come up with different solutions allowing for adaptation and recommendation of the Portal’s structure. Most of them focus exemplarily on the adaptation of the navigation.

Manual Adaptation. First, options to manually adapt the navigation have been introduced. Therefore we implemented specialized portlets that allow each single user to generate her own navigation matching her preferences best. The first portlet allows users to generate their own navigation by hiding irrelevant nodes (pages) and by reordering nodes being part of the navigation in order to reach relevant nodes more quickly. The second portlet allows users to record paths (i.e. sequences of pages) traveled often. These recordings can be recalled later and navigated through by just clicking previous and next links. The recordings can even be exchanged with other Portal users which allows experts to record common paths for their colleagues.

Automated Adaptation. Automated adaptation relieves users from generating an optimized navigation manually. We leverage our user models to understand users’ needs. We use a structure reordering algorithm to rearrange pages: more important nodes are promoted to better navigational positions, less important ones demoted or even hidden. Continuous adaptation, based on the most current user models available, guarantees that the navigation permanently fits the users needs as best as possible. As soon as users’ behaviors change their user model changes, too and hence the navigation provided.

Automated Recommendation. Especially users that navigate according to the aimed navigation paradigm [Robertson, 1997] will not like permanent adaptations because of their aggressiveness. Automatic provisioning of recommendations avoids the permanent restructuring of the navigation while still providing users with shortcuts. We blend-in recommendations into the Portal’s theme that provides users with reasonable shortcuts to relevant pages. These shortcuts are dynamically generated depending on the current navigational position. Our recommendation system applies a MinPath algorithm [Anderson et al., 2001]. We try to predict shortcuts to nodes that are far away from the current node but have a high probability to be navigated to. The probability itself is calculated based on Markov chains as described in [Anderson et al., 2002; Smyth and Cotter, 2003].

Context-adaptivity. As mentioned above, users may have several different context profiles. By switching to a different profile, the Portal will be adapted accordingly based on the information contained in that profile.

Tag-based Adaptation. Based on the users’ tagging behavior analysis described in the previous sections, we can create alternative navigation structures and page layouts, e.g., pages annotated more frequently can be placed at better positions, or portlets annotated with semantically similar tags could be grouped together. In addition, pages

---

1 \(\text{http://www.research.ibm.com/UIMA/}\)
can rearranged according to their semantic distances that, which ensures that semantically related content has a small click distance.

**Tag-Based Recommendation.** Besides adaptations, recommendations might be issued for tags and resources, as the similarity calculations provide values for both. Tag similarity allows us to recommend related tags, based on the currently selected one. E.g., a system might be, among others, comprised of the tags *IBM, WebSphere, Downloads*. Additionally, we can recommend related resources based on the tag similarity. E.g., if a user has selected a page entitled *Company News* and tagged it with the tags *IBM, News, WebSphere Portal*, we can recommend the page *WebSphere Portal News* tagged with tags *IBM, News, WebSphere Portal*.

Finally, to identify users being part of the community with a similar tagging behavior a tag-user matrix is created (comparable to the tag-resource matrix). Each column in this matrix reflects the tagging profile of a user. Calculating the semantic distance between two columns of this matrix reveals the similarity of two users in terms of their tagging history. Our work about expert user determination and implicit social network construction based on users tagging behavior is described in more detail in [Nauerz and Groh, 2008].

### 4.2 Adapting and Recommending Background Information and Related Content

#### Recommending Background Information

When reading web sites, users want background information at their fingertips. If they do not understand what an abbreviation or a term stands for, who a certain person actually is, or, where a certain city is actually located, they want to be able to retrieve this information as easily and quickly as possible. We provide an environment which unobtrusively enriches the information pieces to allow for such look-ups.

Fig. 2 shows our system in action: it illustrates how a fictitious person name (*John Doe*), a location (*Stuttgart*), and a currency have been identified within a text fragment residing in a portlet and are visualized to the user. Pop-ups provide the users with background information.

#### Recommending Related Content

Analyzing occurrences of semantically tagged information pieces also allows us to recommend related content. For instance, if the term *WebSphere Portal* is identified in a news portlet and hence semantically tagged as a product name our system would provide users with background information about WebSphere Portal probably by linking to the product site. But, within a Portal system, the same term might occur at many other places, e.g. in a wiki portlet where users have posted some best practices, tips and tricks when working with this product, in a blog where users have commented on the product and so forth. We track all occurrences and recommend (an appropriate subset) of them as related content as soon as the user interacts with one single occurrence.

This can even be taken one step further. As mentioned above, we allow users to annotate already semantically tagged information pieces. This way we can recommend related content not only by having identified "exactly matching" occurrences of semantically tagged information pieces, but also by having identified similarly annotated, but differently semantically tagged, information pieces. For example, if *Alice, Bob*, and *Charly* have been tagged as persons and a user annotated them with the term *project-x* to express their relationship to this project, this allow us to recommend other users of the community as related "content" as soon as one user is clicked, just because they all seem to be assigned to the same project.

Fig. 2 shows how we can recommend related information for the detected information pieces *Stuttgart* and *John Doe* (other people probably working in the same team, on the same project etc.).

### 4.3 Recommending Expertise

As said, user models also tell us about which pages and portlets a user is typically working with. The first assumption is that users working with certain pages and portlets more often have more expertise about how to use them than other users have. The second assumption is that users working with the same pages and portlets more often have a similar behavior and hence interests and preferences.

For example, if users *A, B, and C* often work with the pages and portlets underneath the page entitled *My News* we can, on the one hand assume that they have knowledge about how to deal with the pages and portlets provided here, and, on the other hand assume that they have similar interests as they do similar things. A user *D* accessing the same pages and portlets rarely can then be presented with *A, B, and C* as experts when dealing with the information and services provided. We have designed a specialized portlet that, shows the contacts added explicitly by the user, the contacts that system has determined to behave similarly, and the contacts currently performing similar actions with the Portal (e.g. viewing the same page or working with the same portlet).

### 5 Conclusion and Future Work

In this paper we have presented a solution for adapting and recommending content and expertise to satisfy Portal users needs and improve collaboration among them. We have shown means to collect the necessary information and to adapt the navigation structure, to recommend background information, related content and expertise.

All the approaches proposed in this paper have been implemented and integrated into IBM’s WebSphere Portal.

For initial evaluation purposes we have set up a demo system and performed some initial surveys. 100% of all participants (all computer scientists, male, 25-50 years old) regarded the system as useful. Of course, we plan to perform more systematic evaluations within the next months.

Future work includes the extension of our recommendation and adaptation techniques. We are currently enhancing our user model with the knowledge about the users’ interests and expertise. We plan to introduce a component that extracts machine-readable semantics from the content of pages that the user has accessed in order to identify the concepts that the user is interested in and has knowledge about. We are also currently working on implementation of an algorithm for automatic selection and composition of services that provide related content and background information. Furthermore, we are interested in more sophisticated visualization mechanisms. E.g., at the moment, we are evaluating how to embed user- and context adaptive (Voronoi) Treemaps to allow users to navigate through the entire information space.
IBM and WebSphere are trademarks of International Business Machines Corporation in the United States, other countries or both. Other company, product and service names may be trademarks or service marks of others.

References


