The VLDB Journal (2000) 9: 56–75

The VLDB Journal

Analysis of navigation behaviour in web sites integrating multiple information systems

Bettina Berendt[1], **Myra Spiliopoulou**[2]

[1] Institute of Pedagogy and Informatics, Faculty of Philosophy IV, Humboldt University Berlin, 10117 Berlin, Germany; e-mail: berendt@educat.hu-berlin.de
[2] Institute of Information Systems, Faculty of Economics, Humboldt University Berlin, 10178 Berlin, Germany; e-mail: myra@wiwi.hu-berlin.de

Edited by P. Atzeni and A.O. Mendelzon. Received June 21, 1999 / Accepted December 24, 1999

Abstract. The analysis of web usage has mostly focused on sites composed of conventional static pages. However, huge amounts of information available in the web come from databases or other data collections and are presented to the users in the form of dynamically generated pages. The query interfaces of such sites allow the specification of many search criteria. Their generated results support navigation to pages of results combining cross-linked data from many sources. For the analysis of visitor navigation behaviour in such web sites, we propose the web usage miner (WUM), which discovers navigation patterns subject to advanced statistical and structural constraints. Since our objective is the discovery of interesting navigation patterns, we do not focus on accesses to individual pages. Instead, we construct conceptual hierarchies that reflect the query capabilities used in the production of those pages. Our experiments with a real web site that integrates data from multiple databases, the German SchulWeb, demonstrate the appropriateness of WUM in discovering navigation patterns and show how those discoveries can help in assessing and improving the quality of the site.

Key words: Web usage mining – Data mining – Web query interfaces – Web databases – Query capabilities – Conceptual hierarchies

1 Introduction

The Web is often described as the ultimate medium for the distribution and interchange of goods, extending from products and services to the primary good of information. However, the placement of information on a web page alone does not guarantee that people will find their way to it, even if they are looking for it.

Like the design of other types of software, web site design should incorporate human-computer interaction principles to help visitors find their way to the information and other goods offered. The core problem is the gap between the still-dominant focus on the design of single pages versus the necessity of considering the design of the web site as a whole. A web site is not a mere collection of web pages rich in information or in product offers. It is a network of structurally or semantically interrelated nodes, usually built in a way that reflects the designers' intuition. If this diverges from the visitors' intuitions, neither carefully gathered content nor sophisticated web page design guarantee the successful distribution of information or goods. HCI design criteria to address this problem are only beginning to emerge [Fle99].

How can we assess the quality of a web site? In this study, we define "quality" as the *conformance of the web site's structure to the intuition of each group of visitors accessing the site*. The intuition of the visitors is indirectly reflected in their navigation behaviour, as represented in their browsing patterns. By comparing the typical patterns with the site usage expected by the site designer, we can examine the quality of the site and give concrete suggestions for its improvement.

The usage of web sites has been analysed ever since the web started. However, both the commercial statistical tools and the mining prototypes were designed with conventional sites in mind, i.e., sites composed of web pages with well-defined static content. The enormous amount of information stored in databases and archives is thus overlooked. This information cannot be reached via static web pages but via form-based pages generated by scripts. Such a form-based site is much more than a trivial front-end to a search engine: digital libraries enable bibliographical research into the sites of multiple publishers. Electronic shops offer online catalogues which are more sophisticated in design and linkage than an endless list of items. Large information servers, like those of the European Union, integrate *and link* material from multiple underlying servers, many of them incorporating databases or text archives.

For a form-based web site, especially one integrating multiple databases and archives, the notion of quality encompasses two issues. First, it concerns the support for navigation across the generated pages, a feature offered by sites retrieving documents already containing links to other documents, i.e., for example, in archives of XML and HTML

Correspondence to: Myra Spiliopoulou

files. Second, it concerns the quality of the query capabilities, i.e., of the search palette, offered for information retrieval. Different searching and browsing patterns may exist in the interactions with a given web site, simply because visitors have different needs and interests. Some users want to reach a known item as quickly as possible, while others may want to get an overall impression of the amount and nature of the items managed in the site [RM98]. The query capabilities should support this diversity.

However, exploration of the data provided by the logs of visitors' actions during a site's normal operation, may signal that some searches are suboptimal and that the search environment needs to be improved. For instance, repeated refinements of a query may indicate a search environment that is not intuitive for some users. Also, long lists of results may signal that sufficiently selective search options are lacking, or that they are not understood by everyone.

In the present study, we propose a complete environment for web usage analysis to pursue the goal of assessing the quality of a form-based site which accesses the information of multiple underlying database servers or archives. Our environment is based on the web utilization miner WUM [SF99, Spi99], which offers tools for the preparation of web usage data, for the discovery of navigation patterns reflecting site usage, and for the visualisation of the mining results.

WUM contains a data preparation module for cleaning the data and compressing them into an efficient input structure. It contains an innovative sequence miner for the discovery of interesting patterns instead of frequent sequences. It also features a particularly powerful principle of pattern modelling and presentation, which assists in the interpretation of the results. The WUM environment is complemented by the application expert's knowledge, reflected in the formulation of the mining problem, in the comparison between expected and actual user behaviour, in the interpretation of the results and in turning the results into practice.

In the next section, we discuss related work on web usage analysis with mining techniques. In Sect. 3, we first describe the framework of the problem by discussing the browsing and navigation functionality offered by modern form-based web sites operating on databases, along with the technical features which enable this functionality and impede traditional web usage analysis. We then discuss data preparation, giving particular emphasis to data cleaning and to the conceptual generalisation of individual web pages. Section 4 introduces the notion of navigation pattern as the target of the analysis, and it describes the mining mechanism for navigation pattern discovery and the visualisation mechanism for the presentation of the results. In Sect. 5, we test WUM in analysing the web usage of a real site and we discuss the results. Section 6 presents the conclusions of our study.

2 Research on web usage analysis

The interest in analysing the usage of a web site is probably as old as the web itself. Early tools focused on total numbers of accesses to a site and relative frequencies for different web pages. Newer generations of tools analysing web accesses include statistical results on usage of pages and small page sequences. While this is an invaluable help in the administrative task of balancing the workload of site servers, it does not give information on how visitors navigate the site (see [ZXH98] for an overview and critique). In order to gain insights into the ways visitors explore and perceive a web site as a whole, we need a more sophisticated framework of data preparation, statistical analysis and result interpretation, combining the knowledge of the analyst with advances in data mining.

2.1 Data preparation for web usage mining

Web usage data differ fundamentally from other datasets used in data mining [CMS99], and there are several problems that must be addressed in preparation for data mining. The main problems are the distinction between different visitors in the log files recording accesses to the site and the reconstruction of all the activities of a single visitor.

The Common Log Format, as well as its most widespread extensions, only records the visitor's host and user agent. This means that different visitors sharing the same host cannot be distinguished. If proxy servers are used, the problem becomes even more acute. Cookies or authentication mechanisms make the identification of a visitor possible, but are undesirable due to privacy concerns. With the exception of some applications like tele-teaching, certain online libraries, and some e-commerce sites, most sites are still free to unauthenticated access.

Assigning activities to distinct visitors. Cooley et al. [CMS99] propose heuristics for the attribution of requests to different visitors. Their first heuristic states that two accesses from the same host but with different browser versions, are initiated from different visitors. The rationale behind this rule is that a user rarely employs more than one browser when navigating in the web. If the agent is recorded in the web server log, this heuristic can be applied to group accesses by host+agent.

The second heuristic of Cooley et al., states that if a web page is requested and this page is not reachable from previously visited pages, then the request should be attributed to a different user [CMS99, CTS99]. For example, if pages A, B, C, D are requested from a host in that order, and if none of A, B, C contains a link to D, then the person requesting D is different from the person that requested A, B, C. The rationale behind this heuristic is that users usually follow links to reach a page and very rarely type URLs. While this is a correct observation, it is also true that users keep bookmarks and can use them to reach pages not connected via links. So, this heuristic might misclassify bookmark invocations as requests from distinct users. The heuristic is also limited in that it can only be used effectively if the site is a sparsely connected graph. Rather, if almost each page can be reached from already visited pages, as is the case in a dense graph, then the heuristic cannot help distinguish between users either.

Reconstructing all activities of a visitor. The second problem is the reconstruction of the activities of a visitor. According to Tauscher and Greenberg, more than 50% of a visitor's

accesses in a site are backward moves [TG97]. Frame-based pages and caches blur this fact. Cooley et al., use information about the web site topology to perform path completion [CMS99]. The problem is probably less acute for form-based sites, because the pages are generated and thus cannot be cached. Although a visitor can move backwards to find a previously visited query form, each new query formulation will refresh the cache.

Establishing sessions. After the identification of different visitors and the reconstruction of their paths, these paths must be split into sessions. A session is a sequence of page accesses performed by the user to accomplish a task. We can define a session in two ways: either as the sequence of pages invoked to fulfil a generic task like "visiting the site", or as a sequence of pages leading to the achievement of a specific goal like the "purchase of a product".

In [SF99], the first definition is used. Sessions are defined on the basis of their duration, i.e., by placing either an upper limit on the total session duration or by posing an upper boundary to the amount of time spent on a single page. If this boundary is exceeded, it is assumed that the user has abandoned the site.

Chen, Park and Yu adhere to the second definition. They specify that a session only consists of forward moves; if a user moves backwards, then she/he is pursuing a different task than before [CPY96]. However, Tauscher and Greenberg have shown that more than 50% of a user's moves are backward moves [TG97]. Indeed, users do move back and forth while exploring a site, without necessarily changing the task they pursue.

Cooley et al., also use a variation of the second definition. They define five types of web pages: (i) "head" pages are entry points for a site; (ii) "navigation" pages contain many links and little information; (iii) "content" pages contain a small number of links and are visited for their content; (iv) "look-up" pages have many incoming links, few outgoing ones and no significant content, such as pages used to provide a definition or acronym expansion; and (v) "personal" pages have very diverse characteristics and no significant traffic. According to this categorisation of pages, they allow a session to be an expert-defined combination of pages of these types [CMS99]. This method has a couple of drawbacks: First, it is not easy to categorise all pages of a site according to the above scheme. Second, the same page may be observed by a user as a look-up page, while another one finds its content significant.

2.2 Data mining on web data

The research on discovering knowledge from web usage mainly concentrates on identifying and grouping contents by relevance [CDAR97, PE98, PPR96, ZEMK97] and on analysing and assisting in the navigation within a web site [CPY96, SF99, Wex96, ZXH98].

Work on grouping contents by relevance points out that this task differs from conventional text mining because page links also bear semantic information. However, the fact that pages are linked does not necessarily imply similarity of contents. Taking these considerations into account, [CDAR97] propose the use of semantic taxonomies of document content. Alternatively, grouping may be based on the clustering of web pages traversed in succession by the site's visitors [ZEMK97, PE98]. This clustering can be used to automatically construct dynamic web pages that provide links to pages suggested as relevant by earlier visitors' movements [PE98]. This can be expected to be very useful in helping visitors find relevant material in a site. However, the problem of adjusting the site to the expectations of the user is simply shifted into designing each cluster and the connections among the clusters in a satisfactory way.

The tool proposed in [PPR96] uses text similarity, topological proximity, and frequency of access along individual links as indicators of semantical relevance. Although good results for the grouping of individual source-page & target-page pairs can be achieved, there is no straightforward way of generalising this approach to longer paths.

Instead of clustering pages accessed together, Wexelblatt records the path followed by each user, identifies the most frequent paths among them and uses them as a basis for recommendations for new visitors [Wex96]. The rationale behind this approach is that if many users follow the same path in their search for information, this path should be suggested to new, inexperienced users, to help them in their search. A similar approach is used in [JFM97]. The goal of these studies is the establishment of a recommendation system for all users rather than the improvement of the site by analysing their behaviour. While the two tasks can be very close, we should keep in mind that if a large number of visitors show preference for a path, this does not mean that *all* visitors are of the same mind. If the recommendation system persists on suggesting this path to new visitors, the path will always be accessed frequently, although some of the visitors may find it non-intuitive.

[ZXH98] employ OLAP technology for prediction, classification, and time-series analysis of web log data. They obtain interesting results on web traffic analysis and on the evolution of user behaviour (e.g., preferred pages) over time. However, the orthogonal issue of assessing the users' behaviour to detect and prevent disorientation by site redesign, is left open.

Büchner and Mulvenna also apply OLAP technology to analyse web usage [BM98]. In their study, the goal is on establishing a cube of web usage data, especially appropriate for e-commerce applications. The data of interest in this context include not only web logs, but also a concept hierarchy, background knowledge of the expert, as well as previously discovered results. The study reveals the importance of electronically capturing and exploiting data from multiple sources in order to perform web usage mining. However, the work presents no results on how those different information assets are combined during analysis.

The miner proposed in [CPY96] discovers statistically dominant paths using a methodology for the discovery of association rules. However, the assumptions made on building those paths are rather over-restrictive. For instance, visitors of a web page do not usually visit *all* children of this page, with the exception of certain application domains like electronically available course material.

2.3 Sequence mining

The generic paradigm of sequence mining supports the discovery of frequent paths composed of not necessarily adjacent pages [AS95, SA96, MT96, Wan97, Spi99]. In [AS95] the problem of sequence mining is modelled as follows. Given a collection of transactions ordered in time, where each transaction contains a set of items, the goal is to discover sequences of maximal length with support above a given threshold. A *sequence* is an ordered list of elements, an *element* being a set of items appearing together in a transaction. Elements need not be adjacent in time but their ordering in a sequence must not violate the time ordering of the supporting transactions. Thus, a sequence miner discovers ordered correlations that appear often enough to make themselves statistically remarkable. In [Wan97], the problem is similarly formulated for more general pattern structures.

In [AS95, SA96, MT96, Wan97], frequent patterns are built incrementally, by discovering frequent patterns of size 1 and extending them stepwise to patterns of larger sizes. Patterns that do not satisfy the frequency threshold are pruned out at each step. Depending on whether the pattern is a sequence or a more generic structure [Wan97], length or a more sophisticated measure is increased at each step.

The above methodology for discovering sequential patterns on the basis of a sole frequence threshold has certain disadvantages for web usage mining. Consider as example a web site with pages W ("Welcome"), A, B, C, D, E, and assume that there is a link from W to D. After preprocessing, the fictional log consists of the following sessions: WABC appearing 1000 times, WBDC appearing 100 times and WABDEC appearing 400 times. The analyst sets the frequency threshold to 25%. Then, according to sequence mining terminology, the sequence WD is "frequent", because it appears 400+100 times, i.e., in approximately 33% of the sessions. However, if we want to improve the site, the question we would prefer to answer is: are the users using the link from W to D, or should we improve the site to make this link more explicit? To figure out that the link is used only in 1 out of 5 cases, we need to either inspect the *original sessions* or to cross-check the frequent sequence WD against any other frequent sequences that involve W, D and one or more pages in-between. Since sequence mining based just on frequencies does not allow us to discern such differences, we need a more sophisticated mining paradigm.

The web usage miners MiDAS [BBA+99] and WUM [SF99, Spi99] are equipped with a mining language, with which the analyst can drive the miner in discovering only patterns that satisfy much more elaborate criteria than a frequency threshold. Both mining languages allow the specification of "templates" for patterns: only patterns matching the specified templates belong to the mining results. Templates can be constrained on their structure, their content and on statistical measures like frequency and confidence (see [ATS93]). The major difference between MiDAS and WUM is the representation of navigation patterns. In MiDAS, a navigation pattern is still a sequence conforming to a template. In WUM, a navigation pattern is a *directed acyclic graph* composed of a group of sequences that conform to a template. This contains the information needed to answer the question posed above: the usage or non-usage of which links is responsible for the frequency of sequences.

WUM's navigation patterns allow a higher flexibility in the modelling of usage patterns. The goal of this is to improve the analysis of the navigation behaviour of user groups. The model of navigation patterns and the miner that can discover such patterns are presented in Sect. 4.

Like the other miners described in the preceding paragraphs, WUM was originally designed with conventional web sites in mind. The primary goal has been to assess the quality of a site, in which semantically relevant web pages of static content are satisfactorily or inadequately connected to each other. A web site composed of form-based pages, automatically generated from multiple database servers, is fundamentally different from a conventional one. Concepts like topology, page content and linkage have different meanings. The demand for quality remains but now translates to studying and evaluating the navigation behaviour of a visitor across pages, the content of which are determined by the visitor her/himself.

2.4 Measuring the quality of a web site

Web page design can be conceived as a special case of interface design, a topic investigated extensively in the context of Human-Computer-Interaction. Ever since the web became a medium appropriate for commercial interaction, the response of users to commercial web sites has been the subject of investigation too.

Sullivan proposes a number of measures to assess the quality of individual pages [Sul97]: (i) quality of service; e.g., as response time; (ii) quality of navigation, navigation modes supported; and (iii) accessibility: whether a page's existence can be ascertained, and whether it can be found. However, it is not clear how to aggregate these measures to assess the quality of a whole site.

Berthon et al., suggest a methodology for measuring the success of a web site in turning visitors into customers [BPW96]. However, the problem of improving a web site to become more successful in this context, is not addressed.

In [Eig97], Eighmey presents a study for measuring the quality of commercial web sites. The experiments are based on questionnaires and on a representative sample of users that tested a number of especially prepared web sites. The factors considered were information utility of the presented contents, ease of use and attractiveness of the presentation metaphor. The study came to the conclusion that the success of the site mostly depends on the quality of the presentation metaphor and on the level to which users find the information interesting. Unfortunately, the study concentrates on the quality of the web pages within the site, keeping the browsing route within the site fixed. Hence, only the quality of the individual pages was measured.

The study of [Eig97] is typical of the methodology of quality control used in web marketing: evaluation of questionnaires from a sample of users against a list of quality criteria. This methodology has two disadvantages: (i) questionnaires can only be addressed to a group of users, not to the whole population; (ii) the evaluation measures the quality of a site but gives no hints on how this quality should be

improved. Our web usage mining approach alleviates those disadvantages by taking the whole population of users of a site into account and analysing their navigation patterns in a way that gives concrete clues to which parts of the site should be improved.

3 Pre-analysis

In a first step, we need to prepare the raw data that we get from the web server logs. In order to determine sequences of URL requests which represent the navigation behaviours of human users, we need to apply a number of steps including conventional data cleaning, determination of visitors' sessions, as well as the abstraction of script invocations to a manageable number of concepts. This last step is essential in the analysis of form-based web sites; it will therefore be discussed first.

Many modern web sites are form-based. They do not consist of a well-defined number of HTML pages with a fixed topology. Instead, they operate on *large databases or information systems* and allow *browsing* and *search* using a rich palette of search criteria. In response to entries in form fields or similar HTML input elements, HTML *pages are dynamically generated* to allow flexible, up-to-date access to the underlying information systems while at the same time providing short paths through the web site for each individual information request. Advanced form-based web sites also utilise the full flexibility of the web: they provide for navigation across *multiple servers* whose pages are *cross-linked*. Pages can be cross-linked via hyperlinks, or via queries to another server's database. These queries can be created in different ways, e.g., by the script generating a page, or by a hyperlink calling a script located at another server. The effect of cross-linking is that a larger scope of information becomes available to visitors, but the physical location of the online resources remains transparent.

An important type of form-based web sites is the online catalogue, which gives information on a 'stock' of individual entities, e.g., books.

3.1 Constructing conceptual hierarchies for queries

A form-based web site contains a large number of URLs. This is because the site is accessed by queries in which several parameters can be set. From the query results, pages are then generated dynamically. Each combination of query parameters results in a different query. Each result is formed by extracting information from one or more records in the multidatabase or archive, i.e., from a typically voluminous data collection. Thus, both the number of possible queries and the number of dynamically generated result pages are very large. This implies an enormous number of query-URL & result-URL combinations, most of which are invoked only a small number of times.

3.1.1 Classification of URLs

To investigate search *patterns*, the URLs corresponding to queries and query results have to be classified. The classification structure will depend on the structure and contents of the web site. In some cases, a single taxonomy of concepts, e.g., reflecting the page contents, will suffice. However, in most cases, the information offered by a large web site is too rich to be reflected in *one* conceptual hierarchy. Rather, a classification along multiple dimensions of a feature space is more appropriate.

But how can the URLs be classified? If we performed market basket analysis, we would classify the *contents* of the database [BL97]. Web sites used for product merchandising may already possess such conceptual hierarchies [Mar99], while document archivers often exploit thesauri over the documents' contents. However, this type of "content-based" conceptual hierarchy is not appropriate for our web site analysis for two reasons. First, we are interested in visitor behaviour, i.e., in what information the visitors intend and expect to see, rather than in semantically related database contents. Second, the URL classification should depict the semantics of the results *as they are presented in the dynamically generated pages*. If these pages are built by combining multiple records from a multidatabase, existing conceptual hierarchies for each underlying database cannot be directly exploited.

Therefore, we propose the classification of URLs into a *"service-based" conceptual hierarchy*, in which we model the query capabilities of the query processing *service* rather than the *contents* of the pages generated by this service. For this new type of conceptual hierarchy, we introduce a number of application-independent feature space dimensions, along which the concepts describing the application should be specified by the analyst.

Example 1. **School search in SchulWeb.** SchulWeb, the site analysed in the present paper (see Sect. 5), offers typical online catalogue functionality: browsing and searching for information on a selection of real-world entities. One important class of entities are schools. As in other online catalogues (e.g., Amazon, Alta-Vista), the SchulWeb pages relating to each type of entities can be classified into three groups: top-level pages like the home page of the site; list-of-individuals pages reflecting different types of lists of schools; and individual pages that describe one school. This classification reflects one feature space dimension, the *"level of detail"* in the modelling of the site's information.

The visitor can generate a large number of different lists of individuals, by selecting different query parameters and assigning values to them. For SchulWeb, the visitor can specify the federal state (within a pre-selected country), the school type (like primary school, comprehensive school, etc.), and/or a string to be matched against some textual property, e.g., the name of the school, of its webmaster or of the school's home town. The selection criteria reflect another feature space dimension, that of *"specifiable properties"*. The particular properties and their permissible combinations are themselves application-dependent, but the modelling dimension is not.

In Fig. 1, we show the classification of SchulWeb URLs used in our analysis. In the upper part of the figure, we have classified pages relating to schools along the dimension *level of detail*. In the lower part of the figure, we have created two conceptual hierarchies across the dimension *specifiable properties*: they model: (a) the specific search parameters

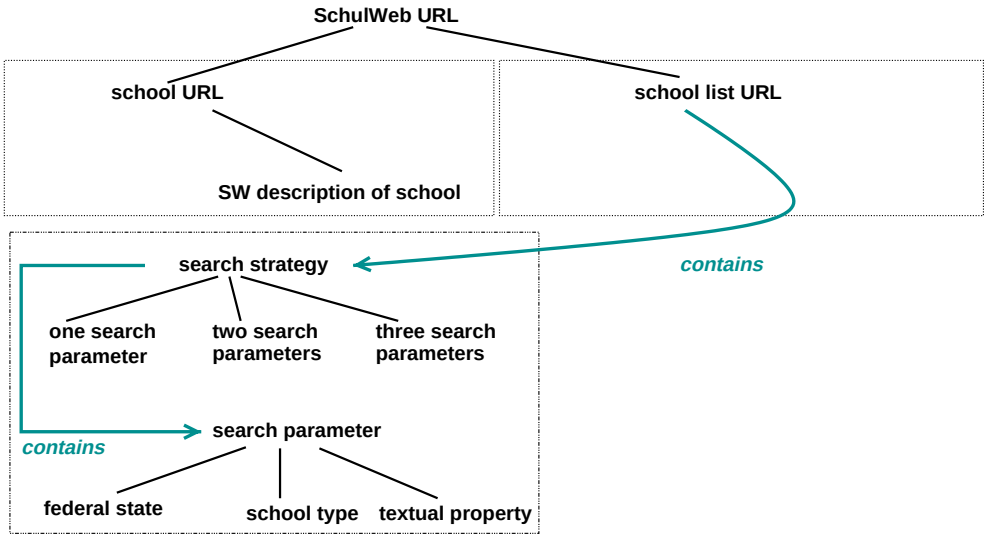

Fig. 1. Conceptual hierarchies in SchulWeb (excerpt)

and (b) the permissible combinations of parameters into a search strategy.

We can see that the service-based conceptual hierarchies of Fig. 1 are more appropriate for analysing the *search behaviour* of visitors than content-based taxonomies would be. Indeed, for our analysis it is important to know how many visitors started their search restricted to only the federal state, only the school type, or a combination of the two, how many visitors opted for search parameters that require text typing, and whether each of these groups of visitors were successful in finding individual schools. This is essential for improving the SchulWeb services. Whether the visitors asked for Bavarian primary schools or Saxonian comprehensive schools matters less, so both queries are mapped to the same composite concept SEITE1-LASALI-D. △

While the concepts depicted in Fig. 1 are peculiar to SchulWeb, the two feature space dimensions are present in each Web site of dynamically generated pages as part of the web-based querying paradigm. Our complete set of application-independent feature space dimensions for service-based conceptual hierarchies is presented next.

3.1.2 Feature space dimensions for service-based conceptual hierarchies

For the classification of URLs in service-based conceptual hierarchies, we propose the following feature space dimensions, across which the application-dependent concepts should be specified.

Entity class:
An entity is an object on which the site provides information. Entities can be documents from an archive, records from a database table or objects built by joining and projecting fields from multiple tables in the same or in different databases. Since a web site may contain different types of objects, we allow for different "entity classes". In market basket analysis, entity classes are expanded to content-based conceptual hierarchies [BL97].

Level of detail:
This dimension concerns the granularity of the representation. We define an ordering composed of "individual" (leaf), "list of individuals" and "top-level".

Specifiable properties:
This dimension expresses the query capabilities of the site in terms of search criteria and criteria combinations that are supported by the querying service.
The concepts defined along this dimension depend heavily on the goals of web site analysis. If the analyst is interested in knowing *which* fields are preferred by visitors, the names and further semantics of these fields should be specified, and probably also field conjunction and field disjunction. If the analyst is interested in *how many* fields are set by visitors during querying, the concepts should reflect the total number of fields, such as "one", "two", "more_than_three", "all".
The SchulWeb hierarchy of Fig. 1 was designed to analyse which fields were used during querying. The fact that SchulWeb only permits conjunctive queries is also reflected in the hierarchy.

Specifiable-property mode:
This dimension covers the different ways by which a value can be given for a specifiable property. It includes choice-button, pop-up field, simple fill-in field and regexp-fill-in field, in which regular expressions can be typed.

Further, application-specific dimensions may be added, like the origin of the data, whether multiple overlapping sources can be accessed or not, or the language of the interface if the site is multilingual.

In our current work, we focus on the first three types of application-independent classification. An important decision concerns the level of abstraction for the conceptual hierarchy in each dimension. For the purposes of navigation pattern discovery, we have opted for an "intermediate level" of concepts, namely the base or derived tables of the underlying databases (dimension: *entity class*) and the fields of these tables (dimension: *search parameter*), thus concen-

trating on the visitors' behaviour rather than on the actually retrieved data.

This "intermediate" level in the conceptual hierarchies is interesting because it gives hints concerning what functionality should be offered by the web site, and it is also the level where statistical distinctions are most likely to produce meaningful results. Analysis at finer levels will often produce frequencies too small to be analysed statistically, and analysis at the coarser level will not differentiate sufficiently. Although visitors interested in different schools or groups of schools, and so on, may show different behaviour patterns, we claim that a detailed analysis is only possible after gaining insights into the usage of the site at an abstract level. A refinement of the conceptual hierarchy of each entity class is a reasonable next step in the site's analysis.

In general, the construction of concept hierarchies over the pages of a web site is a task that cannot be fully automated. This has been shown in the discussion of the example above, and it will be taken up again in Sect. 5.2. However, mechanisms discovering the schema or parts of the schema over semistructured data [GW97, WL97] could be exploited to build at least a preliminary version of the leaf-level of a concept hierarchy and to obtain hints on possible generalisations of the leaf nodes to more generic concepts.

3.2 Data preparation and cleaning

Our web usage mining environment WUM contains a preprocessing module for the preparation of web usage data for the analysis. This module, `WUM_prep`, is responsible for the removal of access records which the analyst considers undesirable or irrelevant (usually images and other files included in the URL requested), for the exclusion of entries made by software agents rather than human visitors, and for the reconstruction of sessions.

We concentrate on files conforming to the W3C Common Log Format extended by the Agent Log and the Referer Log, although WUM supports some more data formats. This format records host name/IP number, date and time of request, request method (`GET / POST / HEAD`), requested URL, page code, number of bytes transferred, user agent, and referer URL.

3.2.1 Exclusion of robots

The most important step of data cleaning was the removal of robot accesses from the log data. We use the term 'robot' to refer to any programmable software agent that does not access a site interactively. These requests can mislead the analyst, because these sequences do not reflect the way human visitors navigate the site.

To exclude these accesses, we employed several heuristic methods based on indicators of non-human behaviour. These indicators are: (a) the repeated request for the same URL from the same host; (b) a time interval between requests too short to apprehend the contents of a page; and (c) a series of requests from one host all of whose referer URLs are empty. The referer URL of a request is empty if the URL was typed in, requested using a bookmark, or requested using a script.

3.2.2 Reconstruction of sessions

In Sect. 2.1, we have seen that in a first step of data preparation, we need to clarify our assumptions concerning the distinction between different visitors who may be sharing one host. Analysing sites that are free to unauthenticated access, we cannot rely on cookies or similar measures to uniquely identify visitors. Moreover, in our current analyses, we are faced with a large majority of visitors using the same user agents. We therefore do not distinguish visitors within the sequence of accesses from one host, except by splitting these sequences into sessions.

There are two fundamentally different methods of reconstructing sessions: by duration or by structure. The second method is appropriate for studying the semantic relationships between web pages, as perceived by the visitors. When investigating how users perceive the web site, this method is less appropriate. In particular, this method defines a session as a *logical* unit of work to reach a target. However, in our study of navigation behaviour, we are interested in finding whether there are logical units of work at all. To achieve this, all information about query reformulations, refinements and new query initiations, as well as all circular moves within the web site need to be retained as part of the visitor's session. Hence, the first method of time-based limitation of a session's duration is the only feasible method.

WUM offers two criteria for determining the duration of a session: an upper limit on the time spent visiting a page and an upper limit of the session duration as a whole. The first criterion is more appropriate for applications where visitors may spend a long time studying the contents of a page, such as tele-teaching. The second criterion is intended for applications where browsing through pages or data items in pages is a usual activity. This is the case when querying form-based sites. Hence, we used the second criterion in our experiments, as described in Sect. 5.

Within these sessions, we take a number of measures to reconstruct the activities of a visitor.

Two subsequent requests for the same URL are collapsed into one if the time between the requests did not exceed a threshold, e.g., 5 s. This threshold can be longer than that for robots because a person needs more time than a program to make a renewed request. Such repetitions typically result from impatience or a mistake (clicking twice), or from a cache first requesting (via `HEAD`) whether a cached page has changed and then, if it has changed, retrieving the new page (via `GET`). The time limit of 5 s is used because it is possible that a visitor requests a page and requests it again a while later, not doing anything in the meantime.

The high connectivity of the pages of a form-based site makes path completion difficult and often unnecessary, see Sect. 2.1. However, *URL completion* becomes necessary when requested URLs are only partially recorded in the log file. Again, this requires knowledge of the site. An example will be discussed in Sect. 5.3.

4 Navigation pattern discovery with WUM

The pre-analysis phase described in the previous section is part of our mining environment WUM. WUM (Web Utilization Miner) is a complete, GUI-based system offering

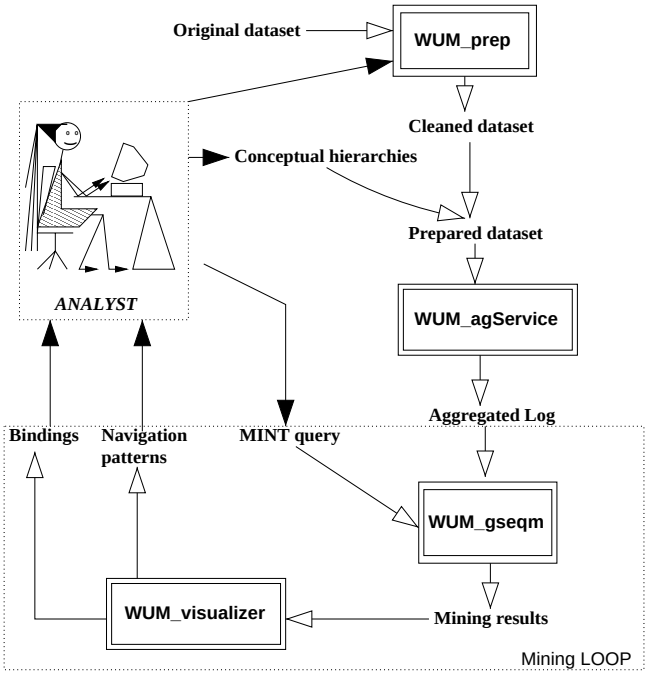

Fig. 2. The architecture of WUM

services for data preparation, advanced sequence mining to resolve the problems discussed in Sect. 2, and a visualisation module for the presentation and closer inspection of the results.

4.1 Architectural overview

The overall architecture of WUM is depicted in Fig. 2. The `WUM_prep` module is responsible for the pre-analysis phase, i.e., for cleaning the dataset and organising the accesses into user sessions, as described in the previous section. The `WUM_agService` module stores the prepared dataset into a permanent data structure, the "Aggregated Log", on which data mining is performed according to the instructions of the analyst. These instructions are expressed in the mining language MINT and forwarded to the mining core `WUM_gseqm`, which discovers patterns conforming to the instructions. The `WUM_visualizer` presents the results to the analyst in graphical form.

In the figure, white arrows indicate information input to or output from WUM, while black arrows indicate instructions from the analyst or results returned to her/him. The interactive and iterative nature of the mining process is depicted in the interrogation of the human with the "mining loop".

The goal of WUM is the discovery of navigation patterns. The dataset prepared by the end of the pre-analysis phase consists of sessions of page accesses. In the next subsection, we model navigation patterns on the basis of sessions and describe the notion of template as basis for pattern discovery. Conventional sequence miners as described in Sect. 2 cannot cope with these types of patterns, so that an innovative mining technique is necessary. Thus, in Sect. A, we briefly present the mining algorithm itself, accompanied by the mechanism for graph visualisation. The complete theory of navigation patterns can be found in [Spi99], while the algorithm is thoroughly described in [Spi99, SFW99].

4.2 Navigation sequences and navigation patterns

A session is a directed list of page accesses performed by a user during her/his visit in a site. A navigation pattern should be a structure that: (a) emphasises the common parts among the sessions; (b) does not purge the dissimilar parts; and (c) annotates both common and non-common parts with quantitative information, such as frequence of occurrence.

4.2.1 Sessions and sequences

We denote as P the set of web pages in the site. If the site is of dynamic nature, P is the set of all pages that can be generated. If concept hierarchies have been used, P is rather the set of concepts to which the actual URLs have been mapped.

Further, let $\mathscr{D}$ be the prepared dataset of sessions output by the pre-analysis. A session is actually a directed list of elements from P. In the general case, $\mathscr{D}$ is not a set but a multiset, since the same session might appear more than one time; this is particularly likely for very small sessions of users that enter the site, visit one or two prominent pages and leave again.

Definition 1. *Let $\mathscr{N}$ denote the set of positive integers (without zero). A "sequence" of length $n > 0$ is a vector $s \in P \times \mathscr{N}^n$ for which there exists a session $\varsigma \in \mathscr{D}$ such that:*

- *ς is comprised of n accesses to pages*
- *For each $i = 1, \ldots, n$, $s[i] = (\varsigma[i], j)$ where j is the occurrence number of $\varsigma[i]$ in the session.*

An element of a sequence is called a "page occurrence". □

In this definition, we map the pages of a session into elements of a sequence, whereby each element is a pair comprised of the page and a positive integer. This integer is the *occurrence* of the page in the session, taking the fact into account that a user may visit the same page more than once during a single session. (According to Tauscher and Greenberg, the probability of such a revisit is approximately 65% [TG97].)

Example 2. Let the set of pages for a given site be $P = \{a, b, c, d, e, f, g, h\}$, and let $ab, ac, abcde, bcbf, abdfhe$ be the sessions that appear in $\mathscr{D}$, where ab denotes an access to page a followed by an access to page b. We can see that many sessions begin with the same page a, some of them also have common prefixes, e.g., ab for $ab, abcde, abdfhe$, while others have different parts in common, e.g., $abcde$ and $bcbf$ that share bc.

According to the above definition, each session in $\mathscr{D}$ is mapped into a sequence as shown in Table 1.

We can see that in session 4, page b has been accessed twice. The first occurrence of this page in the respective sequence is annotated with the number 1, the second occurrence with the number 2. △

Table 1. Mapping the example sessions to sequences

No.	*session*	*sequence*
1)	ab	`(a,1)(b,1)`
2)	ac	`(a,1)(c,1)`
3)	$abcde$	`(a,1)(b,1)(c,1)(d,1)(e,1)`
4)	$bcbf$	`(b,1)(c,1)(b,2)(f,1)`
5)	$abdfhe$	`(a,1)(b,1)(d,1)(f,1)(h,1)(e,1)`

Let P be the set of pages in a site as denoted above. We denote as $U := P \times \mathcal{N}$ the set of page occurrences, over which the elements of sequences may range. Then, for each sequence s it holds that $s \in U^*$, where * is the Kleene star. Finally, we denote as $\mathscr{L}$ the "log of sequences", on which the sessions in the log $\mathscr{D}$ are mapped. Since $\mathscr{D}$ may contain duplicates, $\mathscr{L}$ is also a multiset.

4.2.2 Generalised sequences

A sequence depicts the behaviour of a single user, as a vector of adjacent page requests. In our analysis, we are rather interested in *patterns* that reflect the navigational behaviour of many users. Such patterns may be interesting because they are frequent, i.e., match many sequences, or for some other application-specific reason related to their statistics, structure or content. To discover such patterns, we first model them using the concept of "generalised sequence", defined below.

Outside the set of page occurrences U, we specify a "wildcard" $[low, high]$ that is matched by any sequence of elements that has length at least $low \geq 0$ and at most $high \geq low$. $high$ may take the special value $+\infty$ indicating a sequence of arbitrary length. Also, both low and $high$ may be equal to zero, denoting adjacent elements. In the following, we denote a wildcard with $\star$, if its range is not of interest.

Definition 2. *A "generalised sequence" or "g-sequence" g is a vector $g_1 \star g_2 \star \ldots \star g_n$, where $g_1, \ldots, g_n \in U$. The number of non-wildcard elements in g is the length of g, $length(g)$.* □

A g-sequence matches a sequence in the $\mathscr{L}$ if the sequence contains the non-wildcard elements of the g-sequence in that order and in distances within the boundaries specified by the wildcards.

Example 3. We assume the same sessions as in Example 2. The g-sequence `(a,1)`$\star$`(b,1)[2;4](e,1)` is matched by the 3rd and the 5th sequence above. △

Definition 3. *Let $\mathscr{L}$ be a sequence log over sequences from U^* and let g be a g-sequence over elements of U. The group of sequences from $\mathscr{L}$ that match g constitute the "navigation pattern of g" navp(g).* □

Thus, a navigation pattern is the uniquely defined group of sequences matching "its" g-sequence. Similarly to the sequence log $\mathscr{L}$, this group of sequences may contain duplicates. For pattern discovery, we are not interested solely in the contents of the patterns, but also in their statistic properties; the importance of a pattern depends also on whether it is frequent or rare.

We model the statistical properties of navigation patterns on the basis of the g-sequences describing them.

Definition 4. *Let $\mathscr{L}$ be a sequence log over sequences from U^* and let g be a g-sequence over elements of U. The $hits$ of g, $hits(g)$, is the number of sequences in $\mathscr{L}$ that are matched by g.* □

Using the notion of hits for a g-sequence, we introduce the notion of confidence of a g-sequence element against a previously occurring element *in the context of the g-sequence*, in a way generalising the notion of confidence for association rules introduced in [ATS93]. In particular:

Definition 5. *Let $\mathscr{L}$ be a sequence log over sequences from U^* and let $g = g_1 \star g_2 \star \ldots \star g_n$ be a g-sequence over elements of U. For each $i = 2, \ldots, n$ and for each $j < i$, the "confidence of g_i towards g_j within g" is the ratio of the number of sequences containing $g_1 \star \ldots g_{i-1} \star g_i$ to the number of sequences containing $g_1 \star \ldots g_j$:*

$$confidence(g_i, g_j, g) = \frac{hits(g_1 \star \ldots g_{i-1} \star g_i)}{hits(g_1 \star \ldots g_j)}$$

whereby the confidence of g_1 within g is:

$$confidence(g_1, e, g) = \frac{hits(g_1)}{|\mathscr{L}|}$$

where e denotes the empty sequence. □

The definition of the confidence of g_1 needs some explanation: in the general case, we defined the confidence of a prefix g' of g towards a prefix of g' itself. Since g_1 is the first element of g, the only prefix to it is the empty sequence e, which can be observed as part of every g-sequence. Then, the $confidence(g_1, g) \equiv confidence(g_1, e, g)$ is the ratio of the sequences that match g_1 to the sequences that match the empty sequence e. Since e trivially matches any sequence, the denominator of this ratio is the cardinality of the whole log $\mathscr{L}$.

Example 4. We consider the same sequence log as in Examples 2 and 3, but we now state explicitly how many times each sequence appears in $\mathscr{L}$.

The g-sequence g = `(a,1)`$\star$`(b,1)[2;4](e,1)` matches the 3rd and 5th sequence. The value of $hits(g)$ (Def. 4) is the number of appearances of these sequences in $\mathscr{L}$, i.e., 30 + 10 = 40.

To see how the confidence values of the elements in g (and in $navp(g)$) can be computed, we model $navp(g)$ as a tree structure: we merge all sequences matching g by common prefix. We then annotate each tree node with the number of sequences that contain the tree branch up to that node. The resulting tree is shown in Fig. 3.

At the root of the tree in this figure, `(a,1)` is annotated by 100, because it appears in 100 sequences: 30 times as the first page of `(a,1)(b,1)(c,1)(d,1)(e,1)`, 40 times as the first page of `(a,1)(b,1)`, 20 times in `(a,1)(c,1)` and 10 in `(a,1)(b,1)(d,1)(f,1)(h,1)(e,1)`. Remarkably, although only the 3rd and 5th sequence are matched by g, the statistics are computed by taking *all* sequences in the log into account. This reflects Def. 5, where confidence of a (part of a) g-sequence is defined with respect to the frequence of each prefix of this g-sequence (part).

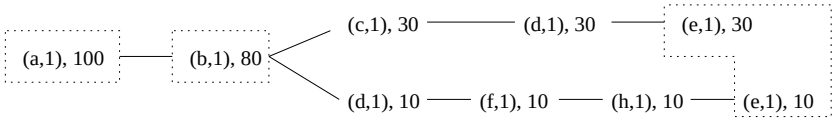

Fig. 3. The navigation pattern of `(a,1) ⋆ (b,1)[2;4](e,1)`

Similarly, to `(a,1)`, we compute the annotation to `(b,1)` by adding the number of appearances of the 1st, 2nd, 3rd and 5th sequence, i.e., of the sequences where `(b,1)` appears directly after `(a,1)`. This gives a value of 80. It should be stressed that taking the 4th sequence into account for this computation would violate the semantics of the g-sequence and introduce an error in its statistics: although `(b,1)` indeed appears 85 times in total, it appears only 80 times after `(a,1)`. △

The tree structure depicted in Fig. 3 is an "aggregate tree" [Spi99]. It merges sequences by common prefix and annotates them by the number of appearances of this prefix in the log of sequences. The log of sequences itself can also be materialised as a tree, using a dummy node as the root of all branches. In fact, this tree is the "Aggregated Log", the permanent data structure in which sequences are stored and on which data mining is applied.

Example 5. The Aggregated Log for the log of sequences $\mathscr{L}$ of Example 4 is shown in Fig. 4.

4.3 Discovering navigation patterns

We have defined a navigation pattern as a group of sequences that match a generalised sequence. A generalised sequence reflects the behaviour of some users, the pages they all accessed and the pages accessed only by some of them. The annotation on the nodes of a navigation pattern reflect the statistics of this group's behaviour.

A g-sequence describes one navigation pattern. During data mining, we are interested in all patterns that satisfy some properties. As already noted in Sect. 2, those properties should be more sophisticated than simple frequency of the pattern. In WUM, we use the mining language MINT to express constraints on these properties [SF99]. These constraints relate to the structure of the g-sequences and are expressed using "templates". Further constraints relate to statistic properties of the patterns, i.e., to the hits and confidence of their components, as defined in Sect. 4.2.2.

The definitions of hits and confidence generalise the definitions of statistics for frequent sequences, as proposed in [AS95]. Hence, g-sequences generalise frequent sequences in conventional mining, while their statistics generalise conventional statistics. We intend to exploit these statistics during pattern discovery to guide the mining process. This implies that we need a more powerful mechanism than those intended for conventional sequence mining. This mechanism is incorporated in `WUM_gseqm` [Spi99].

4.3.1 Templates

A "template" is a vector comprised of variables ranging over the domain U and of wildcards. A "mining query" is a template declaration accompanied by a conjunction of constraints on the permissible values of the template variables. A "binding" for a template is a g-sequence, the non-wildcard elements of which are bound to the template variables and the underlying group of sequences jointly satisfies the constraints on the template. This group of sequences constitutes a "navigation pattern", which together with the corresponding binding is a "solution" to the mining query.

Example 6. In `MINT`, the following clause declares a template with three variables and two wildcards:

```
NODE AS x y z,
TEMPLATE x * y [2;4] z AS t
```

The wildcards pose structural constraints: an arbitrary number of elements may appear between `x` and `y`, while the number of elements between `y` and `z` must be between 2 and 4.

If we issue the above query against the sequence log of Example 4, we can see that the g-sequence g is a solution to the query because it matches the template. Hence, the navigation pattern of Fig. 3 will appear in the result.

The g-sequence `(b,1)*(d,1)[2;4](e,1)` which matches the 5th sequence in the log is a further solution to the above query. △

The declaration of a template should be accompanied by the specification of at least one statistical threshold. Its value should be given by the analyst. Although rules of thumb exist, statistical thresholds depend on the idiosyncracies of the data in the site and change from mining session to mining session, as the analyst gains insights on the statistics of the data she/he analyses.

In `MINT`, the analyst may restrict the absolute number of hits for any variable of the template as well as the ratio of the hits values between *any two template variables*, not just the confidence of consecutive variables. The full syntax of `MINT` appears in [SF99].

Example 7. If the analyst is interested in g-sequences where the first element should appear at least 85 times and the confidence of the second element to the first should be no less than 80%, she can specify the following constraints:

```
WHERE x.support >= 85
AND ( y.support / x.support ) >= 0.8
```

Note that in `MINT` the function hits is named "support". This name should not be confused with the support ratio introduced in [ATS93].

The analyst can further specify that a g-sequence is of interest only if the number of sequences matching it are no less than 40% of the sequences matching only its first element. The complete mining query becomes:

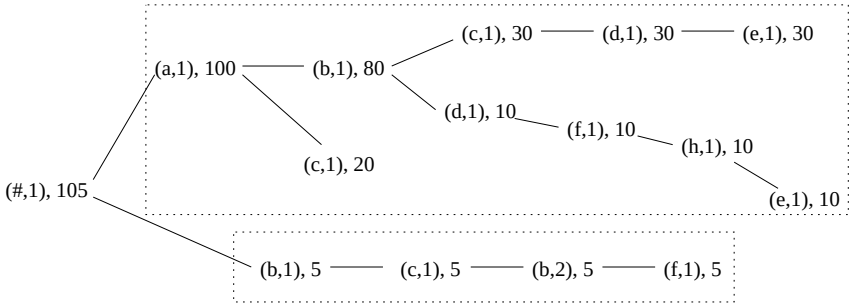

Fig. 4. An example Aggregated Log

```
SELECT t
FROM NODE AS x y z,
     TEMPLATE x * y *[2;4] z AS t
WHERE x.support >= 85
AND ( y.support / x.support ) >= 0.8
AND ( z.support / x.support ) >= 0.4
```

△

4.3.2 Discovering patterns for templates

The mining algorithm of WUM is presented in [Spi99, SFW99] and briefly discussed in the Appendix. Conceptually, a solution to a mining query is found by first generating all possible bindings for the template variables. Some g-sequences thus generated are not matched by any sequences in the log and are eliminated. For each remaining g-sequence $g_1 \star \ldots \star g_n$, we put all sequences matching the g-subsequence $g_1 \star \ldots \star g_i$ into a group G_i, for $i = 1, \ldots, n$. The cardinality of this group should satisfy the constraints of the corresponding template variable. If it does not, the g-sequence is not a solution.

Example 8. For the query of our running example, we assume the Aggregated Log shown in Fig. 4.

As already shown in Example 3 and 4, the g-sequence `(a,1)*(b,1)[2;4](e,1)` is a solution to this query.

G_1 is the group of sequences matched by the binding to the first variable, namely `(a,1)`. It contains all but the 4th sequence in the table of Example 4 and its cardinality is 30 + 40 + 20 + 10 = 100, as can be easily seen from Table 2.

Table 2. The contents of the example log of sequences

No.	*sequence*	*appearances*
1)	`(a,1)(b,1)`	40
2)	`(a,1)(c,1)`	20
3)	`(a,1)(b,1)(c,1)(d,1)(e,1)`	30
4)	`(b,1)(c,1)(b,2)(f,1)`	5
5)	`(a,1)(b,1)(d,1)(f,1)(h,1)(e,1)`	10

G_2 is the group of sequences matched by `(a,1)*(b,1)`, namely:

- the 1st sequence: `(a,)(b,1)`
- the 3rd: `(a,1)(b,1)(c,1)(d,1)(e,1)`
- the 5th: `(a,1)(b,1)(d,1)(f,1)(h,1)(e,1)`

The cardinality of G_2 is 30 + 40 + 10 = 80, i.e. 80% of the cardinality of G_1.

Finally, G_3 is the group of sequences matched by the whole binding, since the template has 3 variables in total. The sequences matched by `(a,1)*(b,1)[2;4](e,1)` are only the 3rd `(a,1)(b,1)(c,1)(d,1)(e,1)` and the 5th one `(a,1)(b,1)(d,1)(f,1)(h,1)(e,)`, as already shown in Example 3. From Table 2, we can see that the cardinality of G_3 is 30 + 10 = 40, i.e., 40% of the cardinality of G_1.

Thus, `(a,1)*(b,1)[2;4](e,1)` is indeed a solution to our example query. This solution is accompanied by the navigation pattern of Fig. 3. In that figure, the non-wildcard elements of the g-sequence are enclosed in boxes. They are exactly the pages to which the template variables have been bound. △

4.4 Pattern visualisation

`WUM_visualizer` is responsible for presenting the mining results in graphical form. The result of a mining query is a set of (g-sequence, navigation pattern) pairs. A navigation pattern is graphically depicted as aggregate tree, as described by the end of 4.2.2. Although the g-sequence describing a navigation pattern is present (and visually apparent) in the latter, `WUM_visualizer` also shows g-sequences separately to facilitate a quick overview of the statistics of the template bindings.

In Fig. 5, we depict part of a navigation pattern produced by a mining query issued during the analysis of the SchulWeb usage (see also Sect. 5). The g-sequence is depicted in a separate window, shown in the lower left corner of the figure.

5 WUM put into practice: experimenting with the SchulWeb site

We have experimented with `WUM` using the access logs of a real form-based web site operating on two servers. The objective of our experiments was to test the effectiveness of the miner for the analysis of navigation behaviour, in its task of accessing and improving the quality of the site's interface towards the needs of its visitors.

5.1 Overview of SchulWeb

The site used in our tests, SchulWeb[1], is a web site connecting schools in the web. SchulWeb accommodates the largest

[1] "School Web" at `http://www.schulweb.de`.

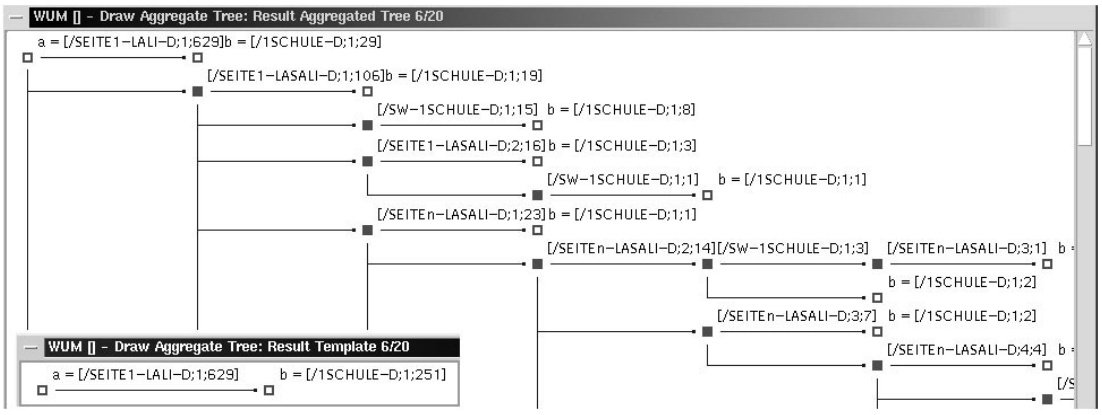

Fig. 5. A g-sequence and part of its navigation pattern

and most comprehensive database of German schools in the web, a database of German-language school magazines in the web, a collection of online resources and of communication services. It also offers access to a major database on educational online resources in Germany, maintained by the *German Educational Resources* GER[2]. Both the SchulWeb and the GER servers are developed and maintained at the Institute of Pedagogy and Informatics at Humboldt University Berlin.

SchulWeb and GER offer integrated access to the two underlying database servers. Access employs forms that query the tables in the databases and return the results in dynamically generated HTML pages. The generated pages often contain links to further resources that are either conventional pages, like the home pages of schools, or dynamically generated ones, like the teaching materials stored in the GER database.

Thus, SchulWeb operates as a site of dynamically generated and often cross-linked pages from multiple databases. In Fig. 6, we see the SchulWeb home page, from which queries can be posed by pop-up menus, choice buttons, fill-in fields and a clickable map.

SchulWeb offers many different services. The first author of this paper has, as domain expert, found through day-to-day work with the site, personal interaction with its users, and inspection of the absolute frequencies of page accesses that: (a) SchulWeb is regarded by many users as a 'large database of schools'; although (b) many visitors rather use SchulWeb as a platform for direct contact, e.g., among teachers in neighbouring schools, or indirect communication, e.g., by making teaching material available electronically. The first observation justifies our decision to select the online catalogue functionality from among the different services offered and to analyse its usage.

5.2 Conceptual hierarchies

The goal of our experimentation was the improvement of the SchulWeb site. As explained in Sect. 3, this implies the analysis of the visitors' behaviour, concentrating on *how* they navigate rather than on *what* they retrieve. This type of analysis requires an appropriate modelling of the dynamically generated URLs. We have proposed the notion of service-based conceptual hierarchy and introduced a number of application-independent feature space dimensions, along which concepts should be specified (Sect. 3.1.2). Part of the conceptual hierarchy of SchulWeb has been presented in the introductory Example 1.

The URLs of the whole SchulWeb are organised along the following feature space dimensions:

Entity class: school, school magazine, online resource
Level of detail: individual, description of individual, list of individuals, top-level
Specifiable properties reflecting the query interface of the web server, as discussed below.

Apart from these application-independent dimensions, SchulWeb has one more dimension, namely the "presentation language", since SchulWeb is a multilingual site. However, previous statistical analysis has shown that most visitors request that pages are generated in German. Hence, we have focused on entries for the presentation language German only.

Figure 7 illustrates the concepts along the dimensions: (i) entity class, on the horizontal axis; and (ii) level of detail, on the vertical axis. In the same figure, we also depict the site's quasi-topology by displaying hyperlinks as arrows. For example, the arrow from the concept `SW-1SCHOOL` to the concept `1RESOURCE` indicates that each URL describing a school contains a link to each 'resource' (teachware document on a specific subject) offered by this school. The topology also reflects the servers providing the data: URLs generated by the SchulWeb server are shown on a white background, pages of the GER server are depicted on a grey background and pages belonging to the sites of the schools are drawn on a grey background at the bottom left of the figure.

In Fig. 7, the concepts enclosed in ellipses are accessible via the SchulWeb main menu, which is contained in each SchulWeb page. Within SchulWeb and GER, the respective home page is accessible from each other page. The corresponding arrows are omitted to avoid graphical clutter.

In Fig. 7, we have provided concepts for all pages generated in the SchulWeb site. We further refine the concepts describing lists of individuals in two ways. First, we need to distinguish between short and long lists of individuals, since the latter are generated by insufficiently restrictive queries. Thus, in the feature space dimension "level of detail", we

[2] Deutscher Bildungsserver at `http://dbs.schule.de`.

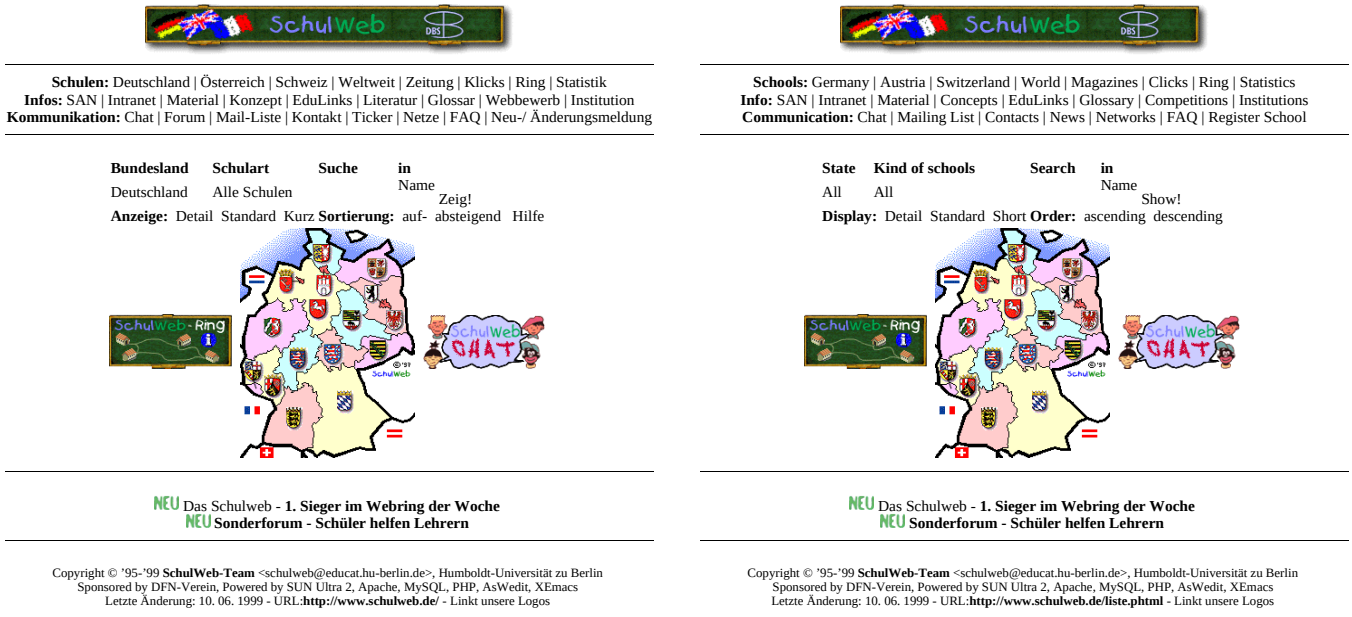

Fig. 6. The SchulWeb home page. Left: standard German version; right: English version

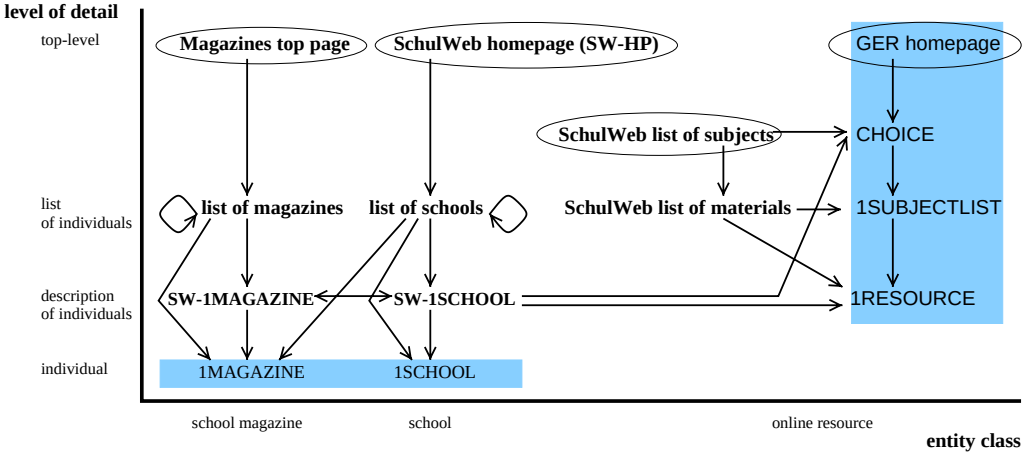

Fig. 7. Concepts in SchulWeb (arrows depict hyperlinks)

refine the lists of individuals by distinguishing between the first list page and a further list page. Second, we model the query strategies used by the visitors by refining the lists of individuals along the feature space dimension "specifiable properties".

Since our experiments concern the retrieval of schools, we concentrate on the concepts describing lists of schools only. The corresponding conceptual hierarchies along the "level of detail" dimension and the "specifiable properties" dimension are shown in the upper and lower part of Fig. 8, respectively. This figure extends Fig. 1.

In Fig. 8, we can see that a list of schools is mapped to the query invocation that produced it, expressed as a string composed of the following:

1. Level of detail: `LI` denotes a list of schools.
   According to our concept refinement along this dimension, `SEITE1` denotes the first page of results, `SEITEn` denotes a further page.
2. Specifiable properties: we model search strategies as combinations of search parameters.
   The search strategies are modelled as string combinations of the following: `LA` denotes a restriction on state; `SA` denotes a restriction on school type; `SU` denotes a restriction on a textual property by a search string. The specification of a country as one of `D, AU, CH, W` is mandatory.
   A query is formed as a conjunction of predicates on one or more of the specifiable properties, so that $2^3 - 1 = 7$ types of lists of schools can be constructed. The symbol `DREI` is used as an abbreviation for `LASASU`.

The abbreviations follow the German terms for the properties. For example, a query invocation returning the second page of a list of schools for a specified state in Germany is mapped to the composite concept `SEITEn-LALI-D`.

Complementing the description of navigation possibilities in Fig. 7, we note that each list page contains the query menu for the (re)formulation of queries. In terms of the conceptual hierarchy in Fig. 8, this means that each page of a list of schools contains a link to the first page of any other list of schools.

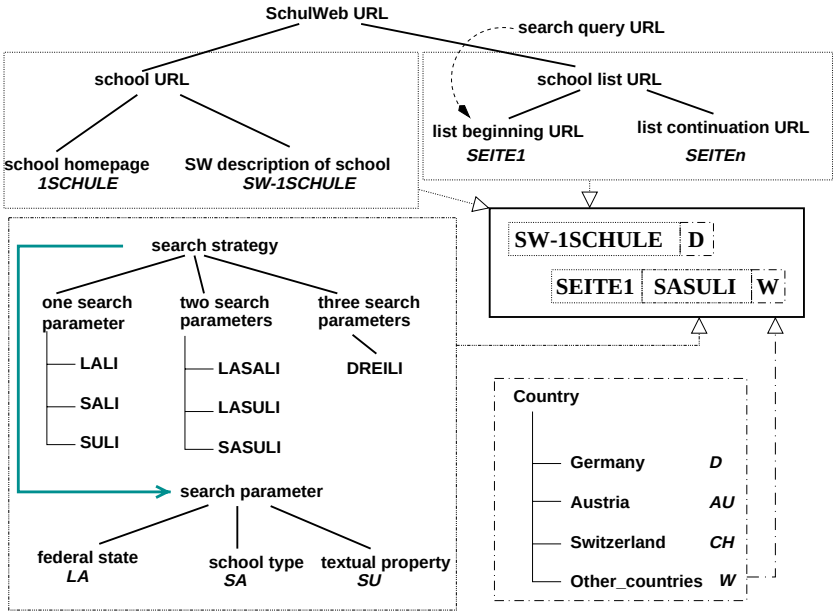

Fig. 8. Conceptual hierarchies in SchulWeb

5.3 The log used in the experiments

For our tests, we used a 24-hour log of visitors accessing SchulWeb from outside Humboldt University. The selected day was a Wednesday in April, outside the German school holidays. Since the data available in SchulWeb are not subject to weekly or monthly variations (as is the case for data related to University examinations), the selected log can be regarded as a representative sample.[3]

The log file was an integrated log file recording accesses to SchulWeb and GER. After removing accesses to images and robot entries, as described in Sect. 3.2, the cleaned log contained 32689 page accesses.

The cleaned log required one additional preparation step, because a `POST` request to page `liste.phtml` can stand for an access to the "first page" of *any* type of list. From the subsequent request to an "individual" or "description of an individual" page, or to a "further page", we could safely reconstruct the query that built the first page in 61.47% of the cases. This reduced the number of invocations to `liste.phtml` from 19.7% to 7.57%, thus removing a considerable factor of distortion from our test log.

By the end of the pre-analysis phase, the concrete query invocations were mapped to strings expressing concepts, as described in the previous section. The reconstruction of visitors' sessions used a heuristic value of 30 min as the maximum total duration.

5.4 Analysis

The present paper and, in particular, the analysis of web usage data, concentrates on only one aspect of usage, the online catalogue functionality. Our background knowledge (see above, Sect. 5.1) led us to expect that SchulWeb is often – though not always – used as an online catalogue of schools, but we had no knowledge of the paths traversed, i.e., of the search process, and therefore no expectations concerning the details of these search processes. In our queries to WUM, we have therefore concentrated on trying to find support for our expectation that school search is a typical usage pattern, and on trying to find out more about the details of these search processes.

We have used WUM to discover typical navigation patterns, to draw conclusions on the visitors' behaviour from them, and to investigate how these conclusions can improve the design of SchulWeb. In the tests, WUM did not only return frequent navigation patterns. The results also revealed the paths between pages accessed together, both frequent and rare. Some of those paths were expected, others were not. We analyse the dominant ones statistically, but our evaluation of the quality of SchulWeb's interface also includes an interpretation of the rare paths we discovered.

5.4.1 Templates and constraints

In the formulation of `MINT` queries, we have used our background knowledge on the purpose of the site and on its expected usage, namely posing queries to retrieve the data of individuals (schools, magazines and online resources). Our first mining session revealed that the web page most often accessed first was the SchulWeb home page for Germany `SW-HP-D`. This is also the top-level page for starting a search, as described in Sect. 5.2, so this result was expected. However, we encountered only 1507 hits for the first access to `SW-HP-D`, corresponding to 4% of the total number of accesses in our log. Since this was the most frequently accessed page, we concluded that our log consists of rather long paths. As we have seen in subsequent sessions, many of these paths contain repeated accesses to `SW-HP-D`.

After studying how SchulWeb is usually being entered, the second mining session started with discovering the navigation patterns starting at that page and having a confidence of at least 10%. Knowing that an individual can be reached

[3] See the web server statistics in `www.schulweb.de/statistik` and `dbs.schule.de/statistik`.

query (1)

```
select t
from node as a b, template a [1;10] b as t
where a.url = "SW-HP-D"
and a.occurence = 1
and ( b.support / a.support ) >= 0.1
```

query (2)

```
select t
from node as a b, template a b as t
where a.url = "SEITE1-LALI-D"
and a.occurence = 1
and ( b.support / a.support ) >= 0.05
```

query (3)

```
select t
from node as a b, template a [1;5] b as t
where a.url = "SEITE1-LALI-D"
and a.occurence = 1
and ( b.support / a.support ) >= 0.1
```

Fig. 9. The MINT queries of our experiments

in 2 steps after the home page, by posing an adequately focused query, and having concluded that our log contains many long paths, we have allowed navigation patterns to contain at least 1 and at most 10 intermediate steps after the top page. We thus specified query (1) of Fig. 9, which produces the same results as the more conventional mining query below but is faster:

```
SELECT t FROM NODE a b,
TEMPLATE a [1;10] b AS t
WHERE a.support > 1000
AND ( b.support / a.support ) >= 0.1
```

Query (1) returned 6 patterns, 2 returning to the top page, 2 reaching a list of individuals expressed as `liste.phtml` and 2 patterns reaching a specific individual, namely a school description `SW-1SCHULE-D` and a school home page `1SCHULE-D`. Since the ultimate function of SchulWeb is to return information on individuals, we focused on the last 2 patterns. We consider the first 4 patterns to be incomplete searches or explorations of the SchulWeb site to be studied in a separate mining session.

All patterns discovered by query (1) and by the subsequent mining queries only returned pages in Germany. This indicates that only requests for schools in Germany were frequent enough to satisfy the statistical thresholds incorporated in our mining queries. In the following summary and discussion of the results, we omit the suffix `-D` (for Germany) in all URLs.

We used `WUM_visualizer` to study the two navigation patterns returned by query (1) that reached an individual. The results of inspecting these two patterns are summarised in the upper part of Table 3. The two patterns are called D and I, respectively. As can be seen from the second table column, the first URL in both D and I is the German SchulWeb home page, appearing in 1907 sequences. This is the value of `a.support` for this binding of variable `a` in query (1).

The column **Goal** contains the URL to which the second template variable was bound. In pattern D, this page was `SW-1SCHULE-D`, while in I it was `1SCHULE-D`. We also list the number of sequences in which this URL was reached after `SW-HP-D` and the maximum number of steps undertaken in total. Since query (1) specifies at most 10 in-between steps, the total number of steps after `SW-HP-D` cannot exceed 11.

The middle column of Table 3 shows the URL requested immediately after the URL bound to the first template variable. For each pattern, we have identified the URL most frequently requested as follow-up. In both patterns, it was the page `SEITE1-LALI-D`, which corresponds to a list of schools retrieved by setting the specifiable property "state" only. We have extracted and summarised the sub-patterns containing `SEITE1-LALI-D` after `SW-HP-D`. These sub-patterns correspond to the g-sequences `SW-HP-D SEITE1-LALI-D [0;9] SW-1SCHULE-D` and `SW-HP-D SEITE1-LALI-D [0;9] 1SCHULE-D`, respectively. In Table 3, they are depicted as $D1$ and $I1$. By comparing the statistics of the goal URL between D and $D1$ and between I and $I1$, we can see that the confidence with which a school was reached after invoking `SEITE1-LALI-D` is much higher than the confidence in the original pattern.

The observations depicted in $D1$ and $I1$ led us to the investigation of navigation patterns that contained `SEITE1-LALI-D`. Query (2) returned the pages accessed in the immediately subsequent step, to test whether an individual was reached directly from this page with a confidence of at least 5%. As a complement to it, query (3) returned longer patterns with 1–5 intermediate steps. Because of the higher number of possible intermediate steps, we chose a higher confidence of 10%. Both queries are shown in Fig. 9.

The identification of the sub-patterns that were further investigated in queries (2) and (3) is an instance of the *data mining loop* made possible by WUM: the visualisation tool enables us to inspect not only the start and goal of frequent sequences, but also less frequent intermediate steps in these sequences. So, we find interesting subgroups and use them in the next iteration of the analysis.

Query (2) returned one pattern only, in which the SchulWeb description of a school was reached immediately after the query invocation with a confidence of 5.35%. This pattern, called Rs (Route $s$hort), is summarised in the lowest part of Table 3, together with the summary results for the pattern Rl (Route $l$ong) returned by Query (3).

5.4.2 Results and interpretation

In the interpretation of results, we made the following assumptions, again reflecting our knowledge of the purpose of the site and our expectation on how it is used.

- Search
  A sequence starting at the top page or at the first page of a list of individuals and ending at an individual represents a search for one or more individuals.
- Follow-up pages: search continued
  A query for a list of individuals with a specifiable property set retrieves the first page of the list. If a further page of the same list is requested, this reflects a browsing activity through the list, as a continuation of the same search.

Table 3. Global description of the analysed patterns

P.	Start		Follow-up request		Goal		Max.
	URL	no. of sequences	URL	no. of sequences	URL	no. of sequences	no. of steps
D	SW-HP	1907	(arbitrary)		SW-1SCHULE	304 15.94% of 1907	11
D1	SW-HP	1907	SEITE1-LALI	158 8.29% of 1907	SW-1SCHULE	67 42.4% of 158	11
I	SW-HP	1907	(arbitrary)		1SCHULE	468 24.54% of 1907	11
I1	SW-HP	1907	SEITE1-LALI	167 8.76% of 1907	1SCHULE	117 70.06% of 167	11
Rs	SEITE1-LALI	579	(arbitrary)		SW-1SCHULE	31 5.35% of 579	1
Rl	SEITE1-LALI	579	(arbitrary)		1SCHULE	134 23.14% of 579	6

– Follow-up pages: search refined
  A query for a list of individuals with a specifiable property set retrieves the first page of the list. If it is followed by a query for a list with the same and at least one more searchable property set, this reflects a refinement of the previous search.

We used those assumptions to construct Table 4 based on Table 3.

In the new table, we study the sub-patterns of the navigation patterns to identify search continuations, search refinements and initiations of new searches. The results show that dominant sub-patterns could be identified, which we interpret as follows.

Searches reaching a school are a dominant sub-pattern. Two of the 6 patterns found by query (1) are searches reaching an individual – a school or the description of a school. The high confidence of these patterns (24.54% for pattern *I* and 15.94% for pattern *D*) shows that they are indeed typical ways of using SchulWeb.

The remaining 4 patterns found in query (1), and not listed here, may describe incomplete searches that would lead to schools in more than 11 steps, especially the navigation patterns ending at `liste.phtml`. They may also describe explorations of SchulWeb, especially those ending again at SW-HP. They were therefore not investigated further.

"State" lists of schools are the most popular lists The inspection of the patterns' constituents showed that the most popular type of list was specified by the *specifiable property "state"* denoted by LALI. Requests for the first page of a LALI list accounted for 2.43% of all page hits, while requests for further LALI pages summed to 3.18% of total page hits. The first page of a LALI list was invoked immediately after the home page in more than 8% of the sequences. The confidence values for all other types of lists were much lower.

As can also be seen in Table 3, the confidence with which a school is reached in patterns *D1*, *I1*, i.e., after invoking a LALI list page, is higher than the respective confidence values over the whole patterns *D* and *I*, respectively.

Schools are rarely reached in short searches. It is possible to reach SW-1SCHULE or 1SCHULE from SEITE1-LALI in one step. So one question arising from the results of query (1) was whether sub-patterns *D1, I1* reached a school in the minimum number of steps, which is 1. This was to be compared to the general patterns *D* and *I*, which took up to 10 steps after the first request. The result of query (2), however, showed that this was *not* the dominant sub-pattern of *Rs*: Only 31 out of 579 (5.35%) reached SW-1SCHULE directly from SEITE1-LALI. The majority reached SW-1SCHULE in a higher number of steps, up to 6, as the result of query (3) show.

Long searches. Query (3) investigated how the longer searches proceeding via SEITE1-LALI reached a school.

The results show that only searches for a school itself (1SCHULE) were a dominant sub-pattern, whereas searches for the description of a school (SW-1SCHULE) were not. This pattern *Rl* had a confidence of 23.14%.

Moreover, the results of query (3) confirmed the most dominant sub-pattern of query (2), showing that after the specifiable property "state" is set, further pages are often requested: patterns with goal node SEITEn-LALI had a confidence of 17.62%.

A closer inspection of the sequences ultimately reaching SW-1SCHULE or 1SCHULE showed that the search for a school often proceeded via further pages of the same list, via a refined search employing state *and* type of school as search criteria, or via a new search. Details are given in Table 4. Here, the "no. of sequences" of "search continued", "search refined", and "new search" gives the number of sequences in which the respective URLs were called immediately after the node SEITE1-LALI. This can be interpreted as a sub-optimality of searches starting with SEITE1-LALI: while in principle, a school can be reached from the home page in 2 steps, searches via SEITE1-LALI often require a much higher number of steps.

It should be noted that the proportion of new searches may be overestimated relative to the proportion of refined searches. This is because occurrences of `liste.phtml` that could not be disambiguated can stand for any kind of new list, i.e., also for one of the list types classified as refined searches. However, it can be expected that this affects all

Table 4. Further actions in the analysed patterns

P.	Goal found (2 steps)	Search continued		Search refined		New search	
		URL	**no. of sequences**	**URL**	**no. of sequences**	**URL**	**no. of sequences**
D	2.83% of 1907	(n.a.)		(n.a.)		(n.a.)	
D1	20 12.66% of 158	SEITEn-LALI	48 30.38% of 158	SEITE1-LASALI	13 8.23% of 158	liste.phtml	25 15.82% of 158
				SEITE1-DREILI	5 3.16% of 158	SEITE1-LALI	4 2.53% of 158
					$\sum$ = 18 11.39% of 158		$\sum$ = 29 18.35% of 158
I	0.79 % of 1907	(n.a.)		(n.a.)		(n.a.)	
I1	11 6.59% of 167	SEITEn-LALI	55 32.93% of 167	SEITE1-LASALI	15 8.98% of 167	liste.phtml	29 17.37% of 167
				SEITE1-DREILI	5 2.99% of 167	SEITE1-LALI	7 4.19% of 167
					$\sum$ = 20 11.98% of 167		$\sum$ = 36 21.56% of 167
Rs	**goal found in next step:** 31 5.35% of 579	SEITEn-LALI	152 26.25% of 579	SEITE1-LASALI	36 6.22% of 579	liste.phtml	73 12.61% of 579
Rl	**goal found in next step:** 0 (23: SW-1SCHULE 3.97% of 579)	SEITEn-LALI	102 17.62% of 579	SEITE1-LASALI	31 5.35% of 579	liste.phtml	61 10.54% of 579
				SEITE1-DREILI	5 0.09% of 579	SEITE1-LALI	7 1.21% of 579
					$\sum$ = 35 6.04% of 579		$\sum$ = 68 11.74% of 579

investigated patterns equally and therefore only biases the overall results, but not the comparison.

5.4.3 Evaluation of the results

The results discovered in our web usage analysis with WUM have implications for the evaluation of the SchulWeb interface, and can serve to recommend changes to the interface. The following paragraphs summarise these implications. They draw on the values in Tables 3 and 4 and their interpretation. They also draw on observations of patterns which were too rare to be included in the statistical analysis. These will be discussed below.

Graphical interaction vs. textual interaction. It appears that although lists with the specifiable property "state" set ("state" lists) give rise to suboptimal searches, they are a highly popular choice from the home page of SchulWeb.

First, what could be a reason for the popularity of this property? "State" lists can be requested using the clickable map on the home page. This graphical or direct manipulation interface is doubtlessly very attractive even for users who have no specific search in mind. Also, even if requested via the choice-button instead of via the clickable map, "state" lists require the specification of only one criterion (i.e., the processing and use of only one choice-button) instead of two criteria. Since more than half of the queries of this type were started via the choice-button, we believe that this option is attractive.

Second, what could be a reason for the sub-optimality of these searches, reflected by search refinements, long browsing and new searches? This may be a consequence of the lists contents: lists constrained only by state seem to be too long, while lists retrieved when both state and type were set, are potentially more useful.

For example, parents or pupils searching for information on schools will often need information constrained by their area of residence, e.g., in order to decide which school they wish (their child) to attend. Similarly, teachers searching for activities and online resources offered by schools will require information constrained by the same parameters. Often, such searches will involve the restriction to a particular school type. In SchulWeb, these constraints can be satisfied by: (i) clicking at the desired state; (ii) choosing the school type; (iii) choosing "city" in the pop-up menu; and (iv) typing the name of the city. While visitors are immediately attracted to specify the desired state, they choose the school type mostly as part of search refinement and rarely ever specify a city name.

Still, these visitors who have to embark on long searches do so with a confidence which is higher than the average. And even those users who abandon the search after using the clickable map (i.e.,, after requesting the first page of a LALI) are attracted to SchulWeb and may return to its pages once they have a more specific search in mind, or they may use other SchulWeb offers.

Textual interaction: the property *search form field.* Only few users started their search employing the form field to type in a *property* value; their number was below 1% of all page hits. However, those users that did this had very successful searches (measured by the proportion reaching the goal, a school), and their searches were very efficient (measured by

the number of steps required to reach the goal). This may be a consequence of fewer users having specific school(s) in mind when starting a search, but it may also be a consequence of a sub-optimally designed interface.

In particular, the form field allows the specification of a (sub)string to be matched against school names. However, this default setting is termed "Search <string> in Name", as can be seen in Fig. 6. It is possible that the word "Name" is misunderstood and should be changed to "School name".

The reluctance with which users type a value in the available form field for property specification can explain the absence of city specifications that could constrain searches for lists of schools. Form fields that must be explicitly selected and require typing seem to be less popular. If this is so, the interface may need to be enriched with more clickable options, e.g., by having an explicit button "city" or by making cities clickable on the map.

Shortcuts in a meta data server. As we have seen, there are different ways of accessing information on a school from a list of schools: the school's home page may be accessed directly, or a SchulWeb description of the school may be accessed. From the latter, there is another link to the school itself. Inspection of patterns *D, D1* and *I, I1* shows that these three ways of obtaining information were all interesting to users: 15.94% of sequences starting at the home page eventually reach the description of a school, and 24.54% eventually reach a school itself. However, of the latter, only a fraction traverse SchulWeb's description of the school, i.e., use SchulWeb's service of displaying meta data on the school. The proportions were 39.91% (pattern *I*), 29.06% (*I1*), 32.56% (*I2*), and 31.34% (*Rl*). This result supports the dual access to information provided by SchulWeb: the concise, 'meta data server-type' description on the one hand, and the direct access to the school on the other hand.

6 Conclusions

In this study, we have proposed a comprehensive and effective environment for web usage analysis in web sites offering query-based access to underlying information systems. Such sites are composed of dynamically generated pages linking data from one or more data sources. Their effectiveness in meeting their visitors' needs lies in the conformance of the query capabilities they provide to the intuition of their visitors. For web usage mining, the challenge lies in discovering navigation patterns that help in assessing and in *improving* this effectiveness.

We have described the WUM environment, which contains methods for the preparation of web usage data and for the discovery of navigation patterns according to sophisticated statistical and structural constraints. To meet the challenge posed by the dynamic nature of query-based web sites operating on information systems, we have proposed the construction of conceptual hierarchies. Since our goal is the investigation of navigation behaviour rather than the discovery of relevant data objects, those conceptual hierarchies describe the query capabilities of the site at different levels of abstraction, rather than the data of the underlying sources.

The combination of WUM and conceptual hierarchies allows us to address a number of core issues in web usage mining. Using conceptual hierarchies, we describe a visitor's activities in a form-based site as a sequence of query invocations, in which query refinements, re-formulations and new searches can be detected. These sequences are combined by WUM to form navigation patterns that satisfy the statistical constraints specified by the analyst. WUM offers a user-friendly mining language, in which sophisticated constraints can be expressed, going far beyond the conventional notion of support threshold. Moreover, WUM discovers navigation patterns rather than individual sequences, and its graphical user interface permits the detailed inspection of these patterns to identify interesting sub-patterns, which are worth further investigation. This investigation is again performed by WUM, due to the larger palette of constraints supported by the mining language.

We have tested WUM on a real web site, SchulWeb, with the goal of assessing and improving the site's conformance to its users' intentions and expectations. The first results of our experiments show the appropriateness of WUM for this challenging task. We have gained insights regarding how the site is perceived and used by its visitors, and we have obtained concrete indicators on how the site's interface can be improved.

Of course, the interpretations presented in this paper are, technically speaking, only causal hypotheses: they have been derived from the analysis of field data, which are moreover mere observation data. The power of web usage mining is that, like other exploratory studies, it can lead to new and unexpected hypotheses and "provide a window on phenomena that are difficult to examine experimentally" [Ben94], p. 112. In a larger research context, this method can be complemented, and the causal hypotheses tested independently, by interview, quasi-experimental, and experimental methods. This will be the subject of further research.

The reader may have noticed that while the pre-analysis phase contains well-defined steps, the analysis and evaluation of the results are more ad hoc, relying heavily on the analyst's background knowledge and intuition. There are no generally applicable rules on how data mining should be performed. In web usage mining, we deal with this fact by offering a mining language by which the analyst has many degrees of freedom in expressing her knowledge to guide the miner. For the concrete problem of measuring and improving the success of a commercial web site, we are currently designing a step-by-step analysis method, first results of which are presented in [SFW99].

Further future work includes the extension of the data preparation methods. For example, by exploiting the load characteristics of the site, we will be able to derive more fine-tuned temporal limits for the construction of sessions. Moreover, we are interested in the incorporation of a mechanism for the construction of conceptual hierarchies as part of the data preparation mechanism of WUM. Moreover, we want to combine the advances of Human-Computer-Interaction on the subject of interface design with the mining capabilities of WUM, in order to better assist in the formulation of hypotheses about web usage patterns and in the interpretation of the discovered results.

Acknowledgements. We wish to thank all those people who have made the work presented in this paper possible: Carsten Pohle, Torsten Veit, and Karsten Winkler as contributors to WUM, and Stefan Dreßler, Peter Glöckner, Alexandra Grote, Ulrike Hammer, Frerk Meyer, Michael Oertel, and Silke Schmiedeberg as contributors to SchulWeb. (All names appear in alphabetical order.) We also thank Humboldt University Berlin and the DFN-Verein for providing the funding and infrastructure that have made this research possible. We are grateful to three anonymous reviewers for their comments on earlier versions of this paper.

References

[ATS93] Rakesh Agrawal, T. Imielinski, and Arun Swami. Mining association rules between sets of items in large databases. In *SIGMOD'93*, pages 207–216, Washington D.C., USA, May 1993.

[AS95] Rakesh Agrawal and Ramakrishnan Srikant. Mining sequential patterns. In *Proc. of Int. Conf. on Data Engineering*, Taipei, Taiwan, Mar. 1995.

[Ben94] John G. Benjafield. *Thinking Critically About Research Methods*. Allyn and Bacon, Needham Heights, MA, 1994.

[BL97] Michael J.A. Berry and Gordon Linoff. *Data Mining Techniques: For Marketing, Sales and Customer Support*. John Wiley & Sons, Inc., 1997.

[BPW96] Pierre Berthon, Leyland F. Pitt, and Richard T. Watson. The World Wide Web as an advertising medium. *Journal of Advertising Research*, 36(1):43–54, 1996.

[BBA+99] A. G. Büchner, M. Baumgarten, S. S. Anand, M. D. Mulvenna, and J. G. Hughes. Navigation pattern discovery from internet data. In *WEBKDD'99*, San Diego, CA, Aug. 1999.

[BM98] Alex G. Büchner and Maurice D. Mulvenna. Discovering internet marketing intelligence through online analytical web usage mining. *ACM SIGMOD RECORD*, pages 54–61, Dec. 1998.

[CDAR97] Soumen Chakrabarti, Byron Dom, Rakesh Agrawal, and Prabhakar Raghavan. Using taxonomy, discriminants, and signatures for navigating in text databases. In *VLDB'97*, pages 446–455, Athens, Greece, Aug. 1997.

[CPY96] Ming-Syan Chen, Jong Soo Park, and Philip S Yu. Data mining for path traversal patterns in a web environment. In *ICDCS*, pages 385–392, 1996.

[CMS99] Robert Cooley, Bamshad Mobasher, and Jaidep Srivastava. Data preparation for mining world wide web browsing patterns. *Journal of Knowledge and Information Systems*, 1(1), 1999.

[CTS99] Robert Cooley, Pang-Ning Tan, and Jaideep Srivastava. WebSIFT: The web site information filter system. In *[MS99]*, 1999.

[Eig97] John Eighmey. Profiling user responses to commercial web sites. *Journal of Advertising Research*, 37(2):59–66, May-June 1997.

[Fle99] Jennifer Fleming. *Web Navigation. Designing the User Experience.* O'Reilly, Sebastopol, CA, 1999.

[GW97] Roy Goldman and Jennifer Widom. DataGuides: Enabling query formulation and optimization in semistructured databases. In *VLDB'97*, pages 436–445, Athens, Greece, Aug. 1997.

[JFM97] T. Joachim, D. Freitag, and T. Mitchell. Webwatcher – a tour guide for the world wide web. In *Proc. of IJCAI'97*, 1997.

[MT96] Heikki Mannila and Hannu Toivonen. Discovering generalized episodes using minimal occurences. In *Proc. of 2nd Int. Conf. KDD'96*, pages 146–151, 1996.

[Mar99] David Martin. IBM SurfAid project: Transactive analysis and prediction. Invited Talk in [MS99], 1999. see also http://surfaid.dfw.ibm.com/.

[MS99] Brij Masand and Myra Spiliopoulou, editors. *KDD'99 Workshop on Web Usage Analysis and User Profiling WEBKDD'99*, San Diego, CA, Aug. 1999. ACM. Online archive of the extended abstracts at http://www.acm.org/sigkdd/proceedings/webkdd99/. Long version of the contributions in preparation for the Springer Verlag LNCS series.

[PE98] Mike Perkowitz and Oren Etzioni. Adaptive web pages: Automatically synthesizing web pages. In *Proc. of AAAI/IAAI'98*, pages 727–732, 1998.

[PPR96] Peter Pirolli, James Pitkow, and Ramana Rao. Silk from a sow's ear: Extracting usable structures from the web. In *CHI'96 (http://www.acm.org/sigchi/chi96/proceedings)*, Vancouver, Canada, April 1996.

[RM98] Louis Rosenfeld and Peter Morville. *Information Architecture for the World Wide Web*. O'Reilly, Sebastopol, CA, 1998.

[Spi99] Myra Spiliopoulou. The laborious way from data mining to web mining. *Int. Journal of Comp. Sys., Sci. & Eng., Special Issue on "Semantics of the Web"*, 14:113–126, Mar. 1999.

[SF99] Myra Spiliopoulou and Lukas C. Faulstich. WUM: A Tool for Web Utilization Analysis. In *extended version of Proc. EDBT Workshop WebDB'98*, LNCS 1590, pages 184–203. Springer Verlag, 1999.

[SFW99] Myra Spiliopoulou, Lukas C. Faulstich, and Karsten Winkler. A Data Miner analyzing the Navigational Behaviour of Web Users. In *Proc. of the Workshop on Machine Learning in User Modelling of the ACAI'99 Int. Conf.*, Creta, Greece, July 1999.

[SA96] Ramakrishnan Srikant and Rakesh Agrawal. Mining sequential patterns: Generalizations and performance improvements. In *EDBT*, Avignon, France, Mar. 1996.

[Sul97] Terry Sullivan. Reading reader reaction: A proposal for inferential analysis of web server log files. In *Proc. of the Web Conference'97*, 1997.

[TG97] Linda Tauscher and Saul Greenberg. Revisitation patterns in world wide web navigation. In *Proc. of Int. Conf. CHI'97*, Atlanta, Georgia, Mar. 1997.

[Wan97] Ke Wang. Discovering patterns from large and dynamic sequential data. *Intelligent Information Systems*, 9:8–33, 1997.

[WL97] Ke Wang and Huiqing Liu. Schema discovery for semistructured data. In *KDD'97*, pages 271–274, Newport Beach, CA, Aug. 1997. AAAI Press.

[Wex96] Alan Wexelblat. An environment for aiding information-browsing tasks. In *Proc. of AAAI Spring Symposium on Acquisition, Learning and Demonstration: Automating Tasks for Users*, Birmingham, UK, 1996. AAAI Press.

[ZXH98] Osmar Zaïane, Man Xin, and Jiawei Han. Discovering web access patterns and trends by applying OLAP and data mining technology on web logs. In *Advances in Digital Libraries*, pages 19–29, Santa Barbara, CA, Apr. 1998.

[ZEMK97] Oren Zamir, Oren Etzioni, Omid Madani, and Richard M. Karp. Fast and intuitive clustering of web documents. In *KDD'97*, pages 287–290, Newport Beach, CA, Aug. 1997. AAAI Press.

A The core of WUM for pattern discovery

Our *g-seq*uence discovery *m*iner `WUM_gseqm` operates on the Aggregated Log tree described in Sect. 4.1. The input to `WUM_gseqm` is a template t and a possibly empty list of predicates restricting the statistics and contents of the named variables in t. Its goal is the discovery of patterns satisfying the query. We distinguish three types of predicates, which we exploit to optimise execution:

- "Type-A predicates" restrict the permissible page occurrences, to which a page can be bound. They correspond to selection operations in conventional database queries.
- "Type B predicates" restrict the length of a path connecting two variables in the template.
- "Type C predicates" specify boundaries for the statistics of the group of sequences matching a g-sequence. They can only be evaluated after the construction of the group

Input: Template $< v_1, *, v_2, \ldots, v_k >$ and predicates of type A, B, C
Output: A set of navigation patterns.

1. *Generate the set of* All_gSequences *by traversing the Aggregated Log:*
   a) *For each order-preserving sequence of nodes* $< n_1, *, \ldots, *, n_k >$ *in a branch produce the g-sequence* $d =< d_1, *, \ldots, *, d_k >$, *where* $d_i = (n_i.page, n_i.occurrence)$.
   b) **if** d *is already in* All_gSequences, **then** *skip it.*
   c) **else if** *for all* $i = 1, \ldots, k$*:*
      i. *The web page referred to in* n_i *satisfies the type A predicates for variable* v_i.
      ii. *The position of* n_i *in the sequence is allowed by the template.*
      iii. *The occurrence number in* n_i *is permitted for* v_i.

      then *add* d *to* All_gSequences.
2. *Construct the navigation pattern for each g-sequence* d *in* All_gSequences*:*
   a) *Compare* d *with the g-(sub)sequences already in the set* Tested_gSequences *and test if it can be rejected without building the navigation pattern.*
   b) *If* d *is not rejected, construct the navigation pattern for it:*
      i. *Find all branches of the Aggregated Log that conform to* d.
      ii. *Merge at each element of* d.
      iii. *Compute the supports of the nodes produced by merging.*
      iv. *Test the C predicates against the navigation pattern.*
      v. *If* d *is rejected*
         – **then** *store the smallest prefix that caused the rejection in the set* Tested_gSequences, *marking it as* R*(ejected).*
         – **else** *store* d *in* Tested_gSequences, *marking it as* S*(uccessful).*
   c) *If* d *is not rejected,* **then** *output its navigation pattern.*

Fig. 10. The `WUM_gseqm` algorithm

of sequences and thus they contribute less to a priori reduction of the search space.

A simplified version of our mining algorithm in pseudocode is shown in Fig. 10. This algorithm is explained in [SFW99]. A faster but more resource consuming variation appears in [Spi99]. Type A predicates are exploited to early reject page occurrences that cannot be bound to the template variables. Type B and type C predicates are used during the generation of g-sequences (bindings): (a) to reject bindings; and (b) establish a list of rejected partial bindings, which is used to reject further bindings without testing. The corresponding heuristics are presented in [SFW99].

The complexity of `WUM_gseqm` is discussed in [Spi99]. Indexing techniques to further enhance performance are planned as future work. However, the relatively low complexity of conventional miners cannot be achieved trivially, because WUM retains information about complete navigation patterns. Since the parts of a navigation pattern that appear between two consecutive g-sequence elements need not satisfy the statistical constraints posed upon the elements, it is impossible to build navigation patterns incrementally. We consider this aspect of WUM's functionality indispensable; the complete navigation patterns are precious to the analyst of a web site. The human effort needed to reconstruct them in conventional sequence mining is considerable, and potentially much more expensive than the resource requirements of WUM.