Web usage mining, site semantics, and the support of navigation

Bettina Berendt

Humboldt University Berlin, Institute of Pedagogy and Informatics,
Geschwister-Scholl-Str. 7, D-10099 Berlin, Germany, berendt@educat.hu-berlin.de

Abstract

To satisfy potential customers of a web site and to lead them to the goods offered by the site, one should support them in the course of navigation they have embarked on. This paper investigates different patterns of search in online catalogues. It presents the tool STRATDYN, developed as an add-on module to the Web Usage Miner WUM (http://wum.wiwi.hu-berlin.de). WUM not only discovers frequent sequences, but it also allows the inspection of the different paths through the site. STRATDYN extends these capabilities: It tests differences between navigation patterns, described by a number of measures of success and strategy, for statistical significance. This can help to single out the relevant differences between visitors' behaviours, and it can determine whether a change in the site's design has had the desired effect. STRATDYN also exploits the site's semantics in the visualisation of results, displaying navigation patterns as alternative paths through information types. This helps to understand the web logs, and to communicate analysis results to non-experts. Analysing navigation in a heavily frequented site, the German 'School Web' (http://www.schulweb.de), we found significant differences between the search strategies employed, leading to proposals for the site's improvement. These highlight the importance of investigating measures not only of eventual success, but also of process, to help visitors navigate towards the site's offers.

Topic area: Data mining methodologies for different web data types

1 Introduction

The way visitors navigate a web site can be used to learn about their preferences and offer them an interface that is better adapted to these. These preferences will often depend on the task and behaviour at hand— each individual visitor should be offered the best interface *for the course of navigation he has just embarked on.*

Sequence mining is the branch of data mining investigating the temporal characteristics of web usage (e.g. [AS95, SA96, MT96, Wan97, BBA+99]). Most sequence miners concentrate on the discovery of frequent sequences. The Web Usage Miner WUM [Spi99, SF99] is special in that not only discovers sequences that are *frequent* and exceed a specified *confidence* threshold (50% of the visitors who looked at a page A later bought a product, and these constituted 20% of all visitors), but also allows the analyst to investigate the details of the paths these visitors took through the web site. This often leads to the discovery of possibly infrequent, but interesting behaviour, allowing one, for example, to track what happened to those visitors who did *not* eventually buy anything, and to find out possible causes for their leaving the site [SPF99].

However, WUM is limited in its ability to support statistical tests for significance, and it is—by design—independent of the site's semantics. The tool STRATDYN (STRATegy DYNamics), which can be used as an add-on module to WUM, was designed to overcome these two limitations. It has a more restrictive method of aggregating sequences, which allows χ^2 tests for the comparison of confidence values to be carried out. A consideration of the site's semantics helps in the automatic generation of queries for these tests. Knowledge about site semantics is usually present in an ongoing analysis of a site. This is also the basis for the generation of easy-to-understand graphics describing essential features of the way visitors use information. This contributes to the range of methods available for analysing web usage data, allowing for more meaningful results of data mining.

The paper is organised as follows: Section 2 briefly describes WUM and the knowledge discovered by it. Section 3 introduces STRATDYN, discusses its definition and use of site semantics, and shows how it complements WUM's existing capabilities. Section 4 gives an overview of a case study in which WUM and STRATDYN were used to analyse usage data of a large 'real-world' web site. Section 5 concludes the paper and discusses further work.

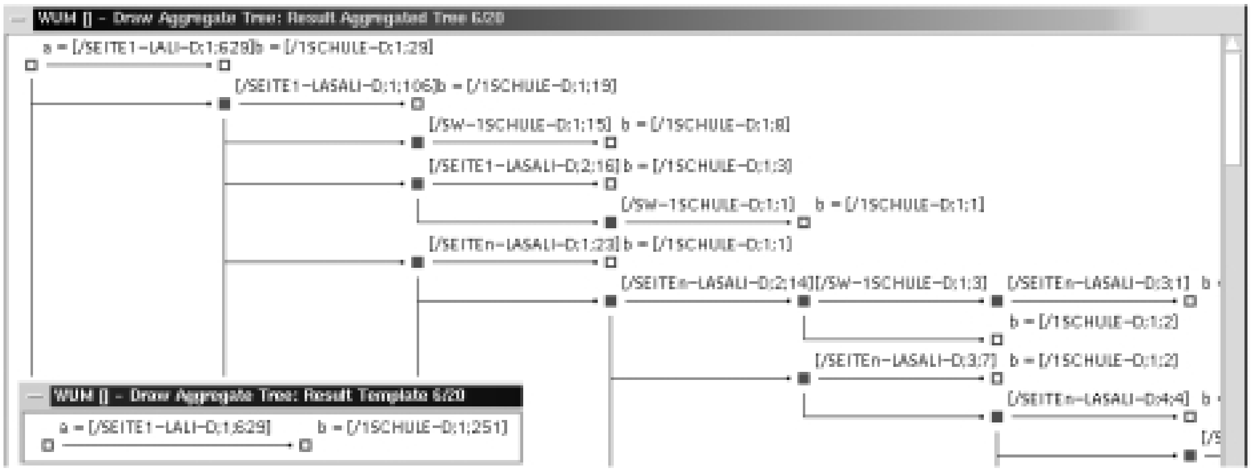
WUM [] - Draw Aggregate Tree: Result Aggregated Tree 6/20
a = [/SEITE1-LALI-D:1;629]b = [/1SCHULE-D:1;29]
[/SEITE1-LASALI-D:1;106]b = [/1SCHULE-D:1;19]
[/SW-1SCHULE-D:1;15] b = [/1SCHULE-D:1;8]
[/SEITE1-LASALI-D:2;16] b = [/1SCHULE-D:1;3]
[/SW-1SCHULE-D:1;1] b = [/1SCHULE-D:1;1]
[/SEITEn-LASALI-D:1;23] b = [/1SCHULE-D:1;1]
[/SEITEn-LASALI-D:2;14][/SW-1SCHULE-D:1;3] [/SEITEn-LASALI-D:3;1] b
b = [/1SCHULE-D:1;2]
[/SEITEn-LASALI-D:3;7] b = [/1SCHULE-D:1;2]
[/SEITEn-LASALI-D:4;4] b
[/S
WUM [] - Draw Aggregate Tree: Result Template 6/20
a = [/SEITE1-LALI-D:1;629] b = [/1SCHULE-D:1;251]

Figure 1: Part of a navigation pattern.

2 WUM: sequence mining and internal process

WUM (http://wum.wiwi.hu-berlin.de, for details, see [Spi99, SF99]) is a sequence miner whose knowledge discovery mechanism rests on three concepts. (1) The basic unit of analysis is a *sequence*: any subsequence (in the technical sense of a subset maintaining order) of a session found in the log file, with each URL annotated by its occurrence number in the session. I.e. if one URL is requested repeatedly during a session, it is assigned occurrence numbers 1, 2, ... So a sequence is an ordered list of page occurrences: [[URL,occurrence]]. (2) A *generalised sequence (g-sequence)* consists of page occurrences and constraints on the lengths of the path between any two subsequent page occurrences ($\star$: minimum and maximum number of nodes, or $*$ for arbitrary path length): an ordered list of [[URL,occurrence],$\star$]. (3) All the sequences that match a g-sequence are aggregated into a tree, the *navigation pattern* (for an example, see Fig.1 and the explanation below).

WUM employs an SQL-like query language. A *query* specifies constraints on a set of generalised sequences. As an example, consider the following Query 1:

```
select t
from node a b, template a * b
where a.url startswith ‘‘SEITE1-’’
and a.occurrence = 1
and b.url contains ‘‘1SCHULE’’
and b.occurrence = 1
```

This query returns navigation patterns that are constrained by the names of the start and goal pages ({a|b}.url), and the length of the path between them (here: arbitrary length *). The expression {a|b}.occurrence=1 restricts the patterns to those in which the start node is the *first* request (within the session) for the URL of node a (b).[1]

The constraints on the URLs are only partial (the URL's name startswith or contains a certain string, as opposed to being equal to that string, =). This means that the query can return several navigation patterns, one for each pair of URLs that can be bound to a and b. In general, this returns a *class* of navigation patterns (here: search for a school (...1SCHULE...) from the beginning of a list of schools (SEITE1...)).

Figure 1 shows part of a navigation pattern produced by this mining query [SB00]. The root node is the URL bound to a (here: SEITE1-LALI-D). The branches are different paths to the URL bound to b (here: 1SCHULE-D); that URL is also the leaf of each branch. Each URL on each branch is annotated by its occurrence and by the number of visitors that have reached this URL along this branch. For example, the second row of Fig. 1 shows that of the 629 visitors who requested SEITE1-LALI-D for the first time in this session, 106 requested the page SEITE1-LASALI-D next, and for the first time in this session. This row also shows that 19 of these 106 went directly to the goal node. The fourth row

[1] WUM queries can also contain constraints on node support and pattern confidence, but these are not relevant for the present argument, and are therefore omitted.

shows that a further 16 of these 106 requested another page of the same kind (SEITE1-LASALI-D with occurrence = 2). The second window at the bottom left of the figure shows the aggregate statistics of the pattern.

Each of the different paths to reach the goal page, and in particular those on which many visitors abandoned the search, may have its own special characteristics and require a particular kind of navigation support. In the analysis of such alternatives, problems, and ideas for page re-design, web usage mining and web design must work hand in hand, and the visualisation of the internal details of a navigation pattern is a valuable help in this cooperation.

For example, inspecting the pattern in Fig. 1 showed us that the most frequent next step is a 'refinement' of the search, a search query with an additional search criterion specified. This is indicated in the URL name: `...LASA...` instead of `...LA...`. This led us to also inspect the other patterns returned by the query for refinements. This interpretation of query results depends on an understanding of the pages' semantics. The visualisation module aids the interpretation (and generation of new hypotheses leading to new queries) by the human analyst.

However, WUM was constructed with a specific focus on the *discovery* of patterns. It was not specifically constructed for the *comparison* of such patterns. For example, one may obtain the result that $a\%$ of first-time visitors of a start page A reached a goal page C, while only $b\%, b < a$, of first-time visitors of a start page B reached C. Or one may obtain the result that $a\%$ of first-time visitors of a page A, but only $b\%$ of first-time visitors of B, refined their search in the next step. Is $b\%$ 'really' less than $a\%$ in a statistically significant sense? Such differences in proportions are commonly measured using Chi-square tests. But because of its focus on sequences, WUM results concerning differences in proportions cannot be tested in this way. This is because *all* subsequences of a session are sequences. The consideration of all of them is useful to show all the different ways of reaching a goal page from a start page. But different sets of sequences, e.g. those defining the different navigation patterns returned by a query, neither form a partition of the set of sessions nor of a subset of it, a necessary prerequisite for the application of Chi-Square tests. For example, a session that contains A, then B, and then C (all first-time visits), is counted towards two sets of sequences: those with start page A and goal page C, and those with start page B and goal page C.

STRATDYN addresses this problem by proposing a sequence/session classification algorithm that partitions the set of sessions. It improves on this generally applicable algorithm, and offers new ways of visualisation, by drawing on the site's semantics. These construct highly compact graphics, supporting the human analyst and web designer (or another non-expert) in the understanding and comparison of navigation patterns.

3 STRATDYN: significance tests, site semantics, and visualisation of strategy dynamics

STRATDYN operates on a file containing sessions. These must be derived from a web server log file according to the usual criteria (e.g. [CMS99, Spi99]). As an add-on module to WUM, STRATDYN operates on the file of sessions generated by WUM's data preparation module.

A core part of processing is the classification of sessions. This is described in section 3.2. It requires a session type hierarchy to be defined (section 3.1). This can be given by an explicitly formulated query, but it can also be automatically generated by a more abstract query specification, drawing on the site's semantics (section 3.3).

3.1 Session type hierarchies

The STRATDYN tool uses *sessions* as the basic unit of analysis. It classifies each session as belonging to exactly one *session type* from a given set of session types, and then counts the number of sessions belonging to each session type. It thereby ensures a partitioning of the set of sessions, such that tests for statistical significance can be performed. In the running example, a session containing A, then B, and then C, is only counted towards the set of sessions with start page A and goal page C.[2]

A *g-sequence* is basically defined as in WUM, with the following differences: No occurrence numbers are used in the current version of STRATDYN (implicitly, only first occurrences of a node are considered), and the URLs need not be explicitly given, but may be defined in relation to the preceding URL, which technically amounts to a disjunction of URLs that can match this part of the g-sequence (see section 3.3). Several sessions may match a g-sequence

[2] So STRATDYN in effect discovers the *maximal* sequences that match the query.

in the sense that they contain a subsequence that matches the g-sequence.

At present, STRATDYN only accepts *queries* that are a subset of all WUM queries. For an arbitrary number of nodes, constraints can be formulated on the URLs of nodes and the lengths of paths between them.

All the g-sequences that match a query are aggregated into a tree representation, the *session type hierarchy*. This extends the idea of a navigation pattern in two ways: (1) Two subsequent nodes need not be subsequent in the session, they can be separated by the number of nodes specified in the g-sequence. (2) In a navigation pattern, the sets of sequences belonging to the nodes succeeding this pattern's root node are all true subsets of the set of sequences belonging to the root node, and they are pairwise disjoint. So together with yet another group of visitors who did something else (exited or followed branches that never reached the goal node), they partition their parent node. This relation between a level (= number of steps from root) and its successor level holds for all succeeding levels. A session type hierarchy extends this idea of partitioning to the root nodes, so all sets of sessions at any one level are disjoint and jointly exhaustive.

At each level of the session type hierarchy, all children of a node "type τ" form a partition of the set of sessions classified as τ. The root node of this hierarchy is a generic type encompassing all sessions. For a given query, various subtypes of this "any session" type are considered: one or more "result" types and "others". Each "result" type corresponds to one of the navigation patterns that would be returned by the corresponding WUM query. This allows one to test a *class* of navigation patterns for differences. In addition, each of these subtypes can be further subdivided. So the tree as a whole represents all paths taken, albeit at different degrees of detail—all session continuations that are not of interest are truncated and declared as "other".
For example, Query 1 would produce the types [any], [any,SEITE1-LALI-D], [any,SEITE1-*] (for each other possible binding of `a`), [any,other],[any,SEITE1-LALI-D,1SCHULE-D],[any,SEITE1-LALI-D,other], [any, SEITE1-*,1SCHULE-D], [any,SEITE1-*,other], (and analogously for each other possible binding of `b`).

3.2 Session classification and χ^2 tests

In the second part of the algorithm, the session log file is read, and each session is classified at each level, yielding a classification for the session consisting of D level-type descriptors τ_d. D is the depth of the session type hierarchy, and $d = 1, ..., D$. The session type hierarchy is processed recursively:

```
(1)  classify (n,d)
(2)   // n:current node=subtype, d:current level
(3)   {counter(n)++;
(4)     // increment the counter of type n
(5)    if (end of path specification at level d)
(6)      {a=[];} // may be last node
(7)    else {min = min(d); max = max(d);
(8)     // path length n to subsequent URL
(9)         a=0;
(10)        for (i=0;i=min-1;i++)
(11)          {read request;
(12)           if (request==EOS) // session end
(13)                  {a=[other]; break;}}
(14)        i=min-1;
(15)         while (a=0) {
(16)   // only if session can still be classified
(17)          if (i>max) {a=[other];}
(18)                      // path length exceeded
(19)          else {read request;
(20)           if (request==EOS) {a=[end];}
(21)                      // subtype [...,end]
(22)           else {for each child m of n do
(23)                  {if (request matches URL(m))
(24)                 {a=append([type(m)]
(25)                           [classify(m,d+1)]);
(26)                      break;}}}}
(27)          i++; }}
(28)     // end:max path length reached,
(29)     // or session classified
(30)   return a; }
```

This algorithm reads each request at most once (it can terminate the classification of a session earlier, depending on the type hierarchy), see lines (11) and (19). Each request is tested at most $1+T$ times (lines (20) and (23)), where T is the number of children of each type and subtype applicable to the session. For a tree of depth D, the average $T = D\times$ the average branching factor of the tree (in our example of typical online catalogue search, $T = D \times 6$, with D between 1 and 5, see section 4). So for a log file containing L requests, the algorithm has the complexity $O(L \times T)$. For a given query, T is constant, making the algorithm linear in the size of the log.

The purpose of the classification of sessions is to find the values of *measures* for session types. While traversing the session type hierarchy, each session has been classified up to D times (fewer if an "other" node is encountered before the whole g-sequence could be matched), along one branch of the hierarchy. The nodes along this branch are the *level types* of the session (line (24) of the algorithm, e.g. "any", "SEITE1-LALI-D", ... in the example), and each ordered list of level types starting with the root node level type (e.g. [any], [any,SEITE1-LALI-D], ...) is a *type* of that session. Each descendant of a type node at a level d is one of its subtypes at a level $d' \geq d+1$. A measure describing a property of type τ is the frequency of some subtype of it, divided by the frequency of τ. The frequencies are measured by the `counter` in line (3) of the algorithm. Typical measures are the *popularity* within a parent type, the frequency of all subtypes divided by the frequency of their parent type, or various *conversion rates*, i.e. confidences of patterns given by the number of visitors of the goal page divided by the number of visitors of the start page [BPW96, SB00].

The frequencies of session types to be analysed further define a table with one row per analysed type $\tau^i, i = 1, ..., I$, and one column for each subtype $j = 1, ..., J$. Each cell (i, j) contains the absolute frequency of subtype ij. Cell $(i, (J+1))$ contains the absolute frequency of type i. Cell $((I+1), j)$ contains the absolute frequency of subtype j. Cell $((I+1), (J+1))$ contains the absolute frequency of the parent type of types $i = 1, ..., I$.

These frequencies are tested against the null hypothesis of equal distribution (if $J = 1$), or independence of the distributions of the types (rows) and subtypes (columns). The latter means that expected frequencies in each cell are defined by the product of the marginal frequencies. The table is partitioned in the required ways [BS66] to compare single cells (e.g. Is the conversion rate of page A higher than that of page B?, comparing types ij and kj), or groups of cells (e.g. compare the conversion rates of pages A1, A2,... with those of pages B1,B2,...). For each comparison, a χ^2 and a p indicating whether or not the difference is statistically significant, are computed.

In a typical data mining situation, one does not have prior hypotheses about the nature of the different navigation patterns. So the individual comparisons will typically be true post hoc analyses investigating possible differences between consecutive categories of session types. This means that alpha error corrections, e.g. Bonferoni corrections, should be performed [Bort93]. (In subsequent analyses, designed to test hypotheses gleaned from datasets used in a first analysis, the paired comparisons may test specific a priori hypotheses, which can obviate the need for alpha error corrections [Bort93].)

3.3 Site semantics

Site semantics denotes any kind of formal description of the 'meaning' of a site's different URLs. Various kinds of schemes for classifying a site's URLs have been proposed. These allow a larger number of visitor sessions or episodes to be identified as instances of one general pattern, as opposed to the very specific paths individual visitors take through the site. This helps towards such diverse goals of analysis as identifying association rules between purchases of goods, determining differences between site designers' goals and visitors' actual behaviour, identifying semantically meaningful navigation episodes, improving the interface, and characterising the workload of a site.

In a first step, the analyst must specify an ontology. This ontology may depend on the domain and typical page content, as in Craven et al.'s [CDF+00] example of computer science departments: these have faculty, staff, courses, ... Classification by page content is also typical in market basket analysis [BL97]. The ontology may also depend on page functions, expressed in structural features, as in Cooley et al.'s [CMS99] distinction between "head" pages serving as entry points for a site, "navigation" pages containing many links and little information, "content" pages containing a small number of links and designed to be visited for their content, "look-up" pages with many incoming links and few outgoing ones, usually providing a definition or acronym expansion, and "personal" pages with very diverse characteristics and no significant traffic. This classification is based on structural features, and it also relates to the kind of activity associated with the page. An ontology based on an elaborate domain event model is proposed by [MAF+99]. They distinguish typical activities performed in an e-commerce site: browse, search, select, add to cart, and pay. Once an ontology has been defined, page classification may be done (semi-)automatically, employing machine learning algorithms that operate on properties of the HTML code and its vocabulary. However, this approach generally has problems with the classification of dynamically generated pages [SCD+00].

In [BS00], we have introduced *service-based conceptual hierarchies*, and a feature space description, to describe the semantics of a form-based site. When mining a form-based site, a classification of URLs is necessary to detect navigation *patterns*, describing *groups* of visitors, because the number of parameters and parameter values that can be specified by a form is so large, and therefore the number of times each individual URL is requested so small, that no meaningful results would be obtained by mining the raw log data. We have proposed a classification by query rather than by content of the parametrised URLs because we were interested in visitor behaviour, i.e. in what information visitors intend and expect to see during their search, rather than in grouping pages and their content by relevance. Classification by query parameters rather than by query parameter values was done because it is more important to know whether and which menu options were chosen, or fields filled in, than knowing how often particular items were requested.

The ontology of information dimensions we have proposed in [BS00] can be used to model a whole class of sites, namely *online catalogues*. Typical online catalogues like online bookstores have pages that can be described along 3 dimensions: (1) *entity class* (e.g. books, records, video tapes, ...), (2) *level of detail*, typically *top-level* (main page or starting page for a specific entity class), *list (of individuals)*, and *individual*, and (3) *specifiable property*. The specifiable properties are usually search criteria. They can be described by their values (e.g. "author" or "title"), possibly in conjunction (e.g. "author+title"). A more abstract description only distinguishes *how many* properties are specified. In addition, where long results are split into several pages, *first pages* can be distinguished from *further pages*.

We have implemented a data pre-processing module that classifies web log data according to these conceptual hierarchies. The details of re-naming are of course dependent on the individual site and its naming scheme, but the task can be semi-automated by parsing URLs for specific parameters as search criteria or page number specifications, for file system path names indicating entity classes, etc.

The structure of common search interfaces is such that from a search initiated by the specification of n search criteria to obtain URL1, the *first page* of a *list* of individuals of some entity class, a visitor can take one of 7 kinds of next steps in requesting URL2. Each of these operations can be detected by parsing the two URLs. In the first five of the following cases, entity class(URL2)=entity class(URL1); in cases 2–5, the level of detail(URL2)=*list* :

1. **goal**: reach an individual page. This is the shortest search. level of detail(URL2)= *individual*.
2. **continuation**: request a *further page* of the same list. specifiable properties(URL2)= specifiable properties(URL1), page(URL2)= *further page*.
3. **repetition**: initiate a new search with the same search strategy, but probably different parameter values. URL2= URL1.
4. **refinement**: add further search criteria to narrow the search. specifiable properties(URL2)=specifiable properties(URL1) + some others, page(URL2) = *first page*.
5. **strategy change**: initiate a new search with yet another search strategy. page(URL2)= *first page*, but neither of type "repetition" nor "refinement".
6. **end** leave the site. URL1 is the last node of the session.
7. **other** do something else. This covers all remaining sessions.

This means that given a query to investigate searches, i.e. session types with the start being a first list page, STRATDYN, can (1) identify the distinct first list pages by matching to create the nodes at the first level, and then (2) generate the second level (the 7 next steps, or some grouping of them, for each node at level 1 define the session types at level 2). With I list types or *search strategies* and $J \leq 7$ *follow-up strategies*, this gives rise to an $I \times J$ table of frequencies as defined in section 3.1.

Generalisations of this approach to other ontologies are discussed in section 5.

This table is then subjected to a Chi-square analysis, testing for differences between the proportions of a given follow-up strategy in each of the I search strategies. Optionally, the program also identifies groupings of strategies which differ significantly in their properties for one measure.

But a *course* of navigation may not be exhaustively described by the first follow-up steps. For example, there may be a group of visitors who *continue* looking at the same list for many steps, while others immediately (and repeatedly) *refine*. Given the number of steps to be examined as input, STRATDYN recursively extends the type hierarchy defined above, defining $I \times J^{d-1}$ session types at level d of the hierarchy, $d \geq 1$. For these, an ever-increasing number of χ^2 tests can be performed. Alternatively, they can be further abstracted and visualised in a low-dimensional state space to aid the process of interpretation.

3.4 Stratograms: a new visualisation method for strategy dynamics

To understand search dynamics at an abstract level, STRATDYN employs a measure of *information concreteness* as one defining dimension of this state space. A second dimension is time, measured by the number of steps. The measure of information concreteness is derived from the site semantics in the following way: A list page with n specified parameters is assigned a value of n, and a page describing an individual is assigned a value of the maximum number of specifiable parameters plus 1.

This gives rise to *stratograms*.[3] Each transition from a page requested in step t with concreteness value i to a page requested in step $t+1$ with concreteness value j is shown in the stratograms by a

[3] inspired by Oberlander et al.'s "proofograms" [OCM+96]

line from (t,i) to $(t+1,j)$, with a slope of $(j-i)$. An exception are continuations and repetitions. These are shown as lines with a smaller slope: 0.1 upwards (repetitions), and 0.1 downwards. Circles located at (t,i) show the number of sessions that finished after step i. (So "other" continuations are not plotted, and refinements and strategy changes are described by their absolute degrees of information concreteness.) The lines' and circles' breadth increases with the proportion of base-level sessions that took the displayed path.

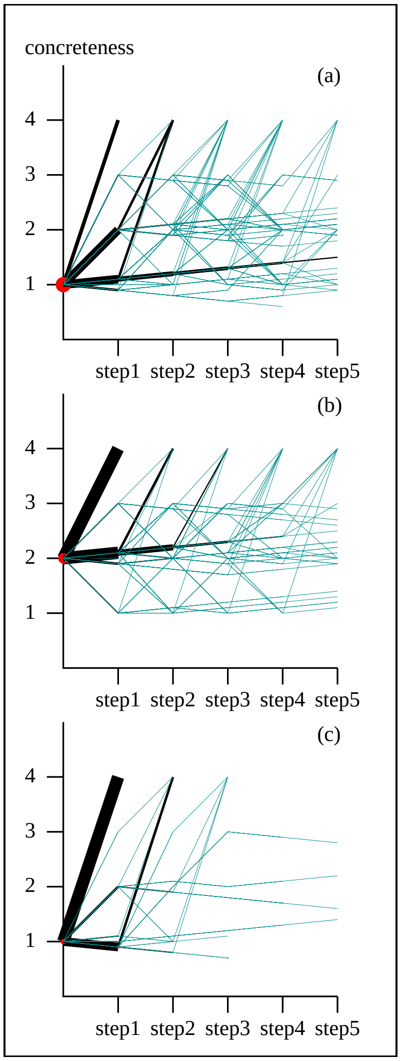

Figure 2: Stratograms.

Figure 2 shows typical stratograms for individual strategies (maximal $n = 3$, 5 follow-up steps plotted). The figure shows two different types: "horizontal" stratograms (a) and (b), and "vertical" ones (c). Strategies exhibiting horizontal stratograms are characterised by a long process of remaining at the same level of concreteness, while strategies with vertical stratograms change the degree of concreteness quickly, most often reaching the goal in the first step. (The visual impression of "horizontality/verticality" is of course measured more objectively by the frequencies of continuation and repetition vs. refinement and goal follow-ups.)

Stratograms describe web navigation in cognitive dimensions, and they complement WUM's visualisations by respecting two basic principles of the human reading of diagrams: the tendency to interpret magnitudes that are ordered in the diagram as ordered in reality [BRB98]—both x and y axis have meaning (unlike in Fig. 1, where the y axis has no meaning), and the search for an 'identity of place'—in the directed acyclic graph of Fig. 2, unlike in a tree, the "same" node has the same place, or at least the same y coordinate. For these reasons, stratograms are a good way of communicating web mining results to non-experts.

This can be interpreted as follows: visitors choosing strategies with horizontal stratograms (a) would rather reach their goal quickly, but are not aware of the existence of more restrictive search options, or (b) enjoy getting an overview of the range of offers of the site. Visitors choosing strategies with vertical stratograms, on the other hand, seem to be oriented mainly towards getting a specific information on an individual. This suggests offering support for visitors: For example, upon invocation of a horizontal strategy, additional help messages explaining the search interface, and/or additional information messages emphasising and linking to the range of offers of the site, may be added.

4 A case study

SchulWeb ("School Web", http://www.schulweb.de) is a form-based site offering, among other services, a database of more than 7000 schools searchable with a number of search criteria and input modes. The site offers typical online catalogue functionality: browsing and searching for information on a selection of real-world entities. These collections of information are the goods offered by the site, and the site's goal is to lead visitors quickly and efficiently to all the information they need. The data and their preparation are described in [BS00, SB00].

The site's semantics were analysed as described in section 3.3. In this site, the specifiable properties for the entity class "school" are *country*, (federal) *state* within that country, *school type*, and *textual property*, a typed-in search string to be matched against school name, city name, or webmaster's name. We concentrated on the entity class "school" and searches for schools in the country Germany, because these constitute the main use of the site. We analysed the follow-up steps as described in section 3.3, and in addition the *conversion efficiency over all paths* (how many of the sessions of type S reached a school URL eventually), *conversion efficiency over minimal paths* (how many of the sessions of type S reached a school URL in the next step) (see [SB00] for an investigation of these measures using WUM).

The popularity measure showed large differences within the 1449 school search sessions (out of a total of 6059 in the whole log), see Table 1. No state+string searches were found. The null hypothesis of equal distribution could be rejected ($\chi^2_5 = 667.17, p < .01$).[4]

Conversion efficiency over all paths was between 51% (state searches) and 58% (type searches), with no significant differences between strategies. However, the conversion efficiency over minimal paths differed, and also the general distribution of next steps. Table 2 shows the percentages. (By definition, refinement is impossible in state+type+string searches.) The null hypothesis of independent distributions could be rejected ($\chi^2_{30} = 402.35, p < .01$). Partitioning the table and computing χ^2 for each column separately showed that each measure was distributed non-equally. These results were significant at the .01 α level.

Strategy	Popularity
state (LALI in Fig.1)	37
type	7
string	7
state+type (LASALI in Fig.1)	27
type+string	9
state+type+string	11

Table 1: Popularity of search strategies (percent).

Further analysis of groupings showed that the input mode for the search criteria may account for these differences. Strategies which required typing a search string ("specification-based" strategies: string, type+string, state+type+string) exhibited significantly lower proportions of continuations and of refinements than those which required choosing from a menu or clickable map ("choice-based" strategies: state, type, state+type). The conversion efficiency over minimal paths was also higher for specification-based strategies. All results were significant at the .01 α level. Specification-based strategies had vertical stratograms (e.g. Fig. 2 (a): string search), while choice-based strategies had horizontal stratograms (e.g. Fig. 2 (b), (c): state and state+type searches). Differences for stratograms were significant except for "end of session" (significant at $\alpha = .05$ for "3-parameter search", and at $\alpha = .01$ for all other next steps).

Strategy	Sch.	Rep.	Cont.	Ref.	Strat.ch.
state	10	7	22	26	0
type	15	8	25	25	2
string	29	21	3	9	7
state+type	29	9	28	5	4
type+string	27	21	2	2	8
state+type +string	36	25	1	0	11

Table 2: Search strategies and distribution of next steps (percent): Sch(ool), Rep(etition), Cont(inuation), Ref(inement), Strat(egy) ch(ange).

5 Conclusions and outlook

The two tools described in this paper, STRATDYN as well as its foundation WUM, discover knowledge that not only concerns aggregate statistics of various patterns of web navigation, but also the paths taken through the site. They complement one another in that WUM is more focused on the discovery of patterns and generation of hypotheses, while STRATDYN is more focused on the testing of hypotheses. This allows more insights into the process of navigation, helping site designers to better support individual visitors by creating pages that adapt to visitors' chosen strategies.

As the case study example described in section 4 shows, the overall conversion rates of the different search strategies did not differ significantly. However, motivated by the discoveries in WUM's output (e.g. Fig. 1), we carried out statistical analysis of the different follow-up strategies (Table 2) and of the 5 next steps (Fig. 2). This showed that the course of navigation starting with a choice-based strategy differs significantly from that starting with a specification-based strategy. The popularity of state and state+type searches led us to conclude that a search by location is needed by many visitors.

[4] Statistical significance is only reported with respect to the common α levels of .05 and .01, although most differences had much lower p values. All reported multiple comparisons used Bonferoni corrections.

However, the search option *state* is not very restrictive, leading to long lists and long search episodes. Also, we know from e-mail contacts that many of the site's visitors are interested in information on schools in their present or future neighbourhood, i.e. usually within a specific *city*. To better support search by location, we proposed to change the site's design so that the possibility to search by typing in a city name (+ optionally choosing the school type) was highlighted.

The statistical tests employed to test difference within one log can also be applied in comparisons between different logs. This can be used to assess the effect of a change in design: Has this change led to a significant improvement of the page's conversion rates etc.? This kind of analysis is currently employed to evaluate the effect of the changes made to the SchulWeb site, with first results showing a marked increase in the use of the more efficient specification-based strategies [SP00].

Another possible use of the data processing methods presented here is prediction: Distinguishing different conversion rates in the session type hierarchy, we can obtain the frequencies with which the different goal pages are reached, given that the session started with session types $\tau^1, \tau^2, ...$ This knowledge could be used to predict the likeliest goal page given that a visitor has started a search episode with the sequence denoted by τ^i. This possibility will be explored in further work.

Further work also includes the extension of the modelling of site semantics, and the consequent automatic query generation, to sites other than online catalogues. The session type hierarchy classification as such can be used for any type of site. (Here, STRATDYN corresponds to WUM, whose query mechanism and the ensuing g-sequences are general-purpose, cf. the classification of different web usage miners in [SCD+00]. WUM has been successfully used in the analysis of different types of sites.) It is of course easier if some degree of automatisation can be used in the generation of the session type hierarchy. The example site type used in this paper, online catalogues, lend themselves rather easily to a classification of 'moves' from one page to another, and therefore to the automatic generation of the hierarchy.

A similar approach could be taken to any type of site described in semantic terms that allow a classification of 'moves' that gives rise to a meaningful session type hierarchy. For example, movement from "head" to "navigation" to "content" pages [CMS99] is similar to the movement from "top-level" to "list" to "individual" pages investigated in the present paper. This kind of classification also has the advantage that it lends itself rather easily to an interpretation as movement along an ordinally scaled dimension, and therefore to a visualisation like that of stratograms, where the order along each graphical dimension is meaningful.

Order is also present in temporal interpretations of event model ontologies. These interpretations impose a canonical order on the activities in an e-commerce site, cf. the process of book purchase illustrated by icons arranged from left to right in http://www.amazon.com. (The analysis of search in our case study can be regarded as a detailed investigation of the search+browse stage of such event models.)

STRATDYN's approach can also be generalised to sites with different ontologies, but it may lose some of its power if a spatial interpretation of page types and movements is not possible or wanted. For example, one could define types of movement in the example domain of [CDF+00]: Let "get-info-on-person" be a move from a faculty's home page to a personal page, and "get-more-info-on-person" be a move from a personal page to this person's courses or publications. One could then compare the different paths via the personal pages of professors versus assistants. "Get-more-info-on-person" could be interpreted as a refinement in the sense used here. However, a visualisation of these data should take into account that the dimension "type-of-person" is not necessarily ordinally scaled, affecting the permissible interpretation of "get-info-on-person". Common techniques to indicate a purely nominally scaled variable will need to be employed.

Another area of our research investigates the use of the different types of searches found by stratograms in user modelling. This builds on findings by [OCM+96], who could identify two different types of their "proofograms" with different cognitive styles of users.

The information gained from an analysis of usage paths could be associated with information on personal interests, demographics, etc., gained from the visitors themselves. Such information can serve as a further criterion of session type classification. We are currently exploring personalisation options for our web servers, which are needed to obtain such data on visitors.

Lastly, we are working on developing stratograms further. In their current form, their main goal is to portray the data in a highly concise format that can be perceived simultaneously, 'at one glance'. This complements the visualisation employed by WUM, whose emphasis lies on exhaustiveness. WUM's visualisation is textually annotated and allows for panning (if navigation patterns are large) and some degree of zooming necessary to read the textual information. It is known from the work of graphics designers as well as from neuropsychological findings that an 'image', a unit that can be perceived simultaneously, is limited to three or four 'visual variables', where position in x-y space is particularly important [Gree98]. The use of position (x,y), shape, and shading in the present stratograms already slightly exceeds three, the number of visual variables recommended for static, two-dimensional displays. However, the use of motion or animation can extend these possibilities [Gree98]. Interactivity certainly also extends them, by allowing attention to be focused on parts of the image, and further information to be perceived, joined to the relevant part of the image, and processed. Interactive versions of stratograms that realise this potential are currently being developed.

Acknowledgements

I wish to thank all members of the SchulWeb team, Myra Spiliopoulou, and Elke Brenstein for work, discussions, and comments that have made this research possible. Three anonymous referees of WebKDD'2000 have made valuable comments that have contributed a lot to the final version of the paper. I also wish to thank Humboldt University Berlin and the DFN-Verein for funding this research and SchulWeb.

References

[AS95] Agrawal, R. & Srikant, R. (1995). Mining sequential patterns. In *Proc. of Int. Conf. on Data Engineering*, Taipei, Taiwan, March 1995.

[BRB98] Berendt, B., Rauh, R. and Barkowsky, T. (1998). Spatial thinking with geographic maps: An empirical study. In H. Czap, H.-P. Ohly, & S. Pribbenow (Eds.), *Herausforderungen an die Wissensorganisation: Visualisierung, multimediale Dokumente, Internetstrukturen (Proc. ISKO'97)* (pp. 63-74). Würzburg, Germany: ERGON-Verlag.

[BS00] Berendt, B. & Spiliopoulou, M. (in press). Analysis of navigation behaviour in web sites integrating multiple information systems. *The VLDB Journal, 9,* 56–75.

[BL97] Berry, M.J.A. & Linoff, G. (1997). *Data Mining Techniques: For Marketing, Sales and Customer Support.* New York: John Wiley & Sons, Inc.

[BPW96] Berthon, P., Pitt, L.F. & Watson, R.T. (1996). The World Wide Web as an advertising medium. *Journal of Advertising Research, 36,* 43–54.

[Bort93] Bortz, J. (1993). *Statistik für Sozialwissenschaftler.* 4^{th}, revised edition. Berlin etc.: Springer.

[BS66] Bresnahan, J.L. & Shapiro, M.M. (1966). A general equation and technique for the exact partitioning of chi-square contingency tables. *Psychological Bulletin, 66,* 252–262.

[BBA+99] Büchner, A.G., Baumgarten, M., Anand, S.S., Mulvenna, M.D.& Hughes, J.G. (1999). Navigation pattern discovery from internet data. In *WEBKDD'99: KDD Workshop on Web Usage Analysis and User Profiling.* Berlin: Springer.

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

[CDF+00] Craven, M., DiPasquo, D., Freitag, D. & McCallum, A. (2000). Learning to Construct Knowledge Bases from the World Wide Web. *Artificial Intelligence, 118,* 69–113.

[Gree98] Green, M. (1998). *Toward a perceptual science of multidimensional data visualization: Bertin and beyond.*
http://www.ergogero.com/dataviz/dviz0.html.

[MT96] Mannila, H. & Toivonen, H. (1996). Discovering generalized episodes using minimal occurences. In *Proc. of 2nd Int. Conf. KDD'96* (pp. 146–151).

[MAF+99] Menasce, D., Almeida, V., Fonseca, R. & Mendes, M.A. (1999). A Methodology for Workload Characterization of E-commerce Sites In *Proc. ACM Conference on Electronic Commerce,* Denver, CO, November, 1999.

[OCM+96] Oberlander, J., Cox, R., Monaghan, P., Stenning, K. & Tobin, R. (1996). Individual differences in proof structures following multimodal logic teaching. In *Proc. COGSCI'96.* 201-206.

[Spi99] Spiliopoulou, M. (1999). The laborious way from data mining to web mining. *Int. Journal of Comp. Sys., Sci. & Eng., 14,* 113–126.

[SB00] Spiliopoulou, M. & Berendt, B. (to appear). Kontrolle der Präsentation und Vermarktung von Gütern im WWW anhand von Data-Mining-Techniken. In H. Hippner, U. Küsters & M. Meyer (Eds.), *Handbuch Data Mining im Marketing.* Vieweg.

[SF99] Spiliopoulou, M. & Faulstich, L.C. (1999). WUM: A Tool for Web Utilization Analysis. In *Extended version of Proc. EDBT Workshop WebDB'98* (pp. 184–203). Berlin etc.: Springer.

[SP00] Spiliopoulou, M. & Pohle, C. (to appear). Data Mining for Measuring and Improving the Success of Web Sites. *Journal of Data Mining and Knowledge Discovery, Special Issue on E-commerce.*

[SPF99] Spiliopoulou, M., Pohle, C. & Faulstich, L.C. (1999). Improving the Effectiveness of a web site with web usage mining. In *WEBKDD'99: KDD Workshop on Web Usage Analysis and User Profiling.* Berlin: Springer.

[SA96] Srikant, R. & Agrawal, R. (1996). Mining sequential patterns: Generalizations and performance improvements. In *EDBT*, Avignon, France, March 1996.

[SCD+00] Srivastava, J. Cooley, R., Deshpande, M. & Tan P.-N. (2000). Web Usage Mining: Discovery and Application of Usage Pattens from Web Data. *SIGKDD Explorations, 1.*

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