Creating User Profiles of Web Visitors using Zones, Weights and Actions

Natascha Hoebel
Database and Information Systems
Goethe – University Frankfurt, Germany
hoebel@dbis.cs.uni-frankfurt.de

Roberto V. Zicari
Database and Information Systems
Goethe – University Frankfurt, Germany
zicari@cs.uni-frankfurt.de

Abstract

In this paper, we report our experience in the implementation of a module for creating user profiles of Web visitors by using Zones, Weights and Actions. The module is part of Gugubarra 2.0, a tool for better understanding and management of communities of registered Web visitors, currently being developed by the database group at the Computer Science Institute of the Goethe University in Frankfurt. In addition we introduce a semantic concept for our approach.

1. Introduction

In this paper, we report our experience in the *implementation* of a module for creating user profiles. In addition we introduce a *semantic concept* for our approach.

The starting point of our project is the assumption that a community of users is registered on a Web site and that for each user a profile is built. A *non-obvious user profile (NOP)* is based on the actions and navigations the user performs on the Web site. In Gugubarra, we offer various settings that can be used to create and manage NOPs. By using these settings the Web site owner can focus on those aspects he wants to analyze.

The rest of the paper is structured as follows: In section 2 the concepts of Gugubarra are defined, i.e. the concept of zones. Therefore section 3 describes how zones are related to the NOP calculation. Further it discusses the Design Principles, including the *semantics of weights* and its impact. In section 4 is show how the concepts are used in the application. Section 5 describes the implementation of Gugubarra 2.0 with focus on the zone module. Related work is presented in section 6.

2. Basic concepts

In this section some definitions that will be used in the rest of the paper are given, i.e. four concepts to help creating profiles*: Topics, Weights, Zones, and Actions.*

2.1. Preliminary definitions

We consider a **Web site** (domain) as composed of different pages. A **Web page** is an HTML document, sent by a Web server, in response to a request of a Web browser. The request might be triggered by a visitor of the Web site.

2.2. Topics and Actions

A set of **topics** $Tp_1,..,Tp_n$ is defined globally with respect to a Web site. **Topics** are used to represent *the type of content* of the Web pages. The set of topics can change dynamically.

A set of **actions** $A_1,..,A_n$ is defined globally and represent the actions (e.g. *click a link*, *download a .pdf*) the user is allowed to do in the Web site, within each Web page. Each action is associated with an **action weight** aw_t. The action weight is a value between 0 and n. An action weight describes the *relevance* of an action as seen from the perspective of the owner of the Web site. The choice of which action weights to assign to the various actions is obviously dependent on the Web site owner's specific business models and goals. The set of actions and their corresponding action weights can change dynamically.

2.3. Content, Placement and Behavior

The following elements played a key role in the design of zones:

- The *content* of a Web site (published information).
- The *type of content*.
- The *placement* of the content on the Web site (i.e. where and how the information is presented to the user and how it is accessible).
- The *relative importance* of the content and placement of information in the Web site.
- The *behavior* of a Web site visitor (i.e. the user's decision process of which information to access).
- The *framing* effects [8], i.e. the effect that additional information presented to the user in the

same zone has on the user's choice of which information to select.

We combined all these elements when we introduced the concept of zones.

Zones allow a much finer level of granularity with respect to the capturing of the information a user of a Web site might be interested in.

2.4. Zones

A **zone** can be a set of (Web) pages, a set of parts within a page, a set of parts of several pages, or any combination thereof. Zones must be disjoint. The zones are used to represent the concept of a "location" on a Web site.

A zone has an associated **state**, which can be: *ON, OFF* or *OFF-ACTION-SENSITIVE*.

For simplicity, individual parts making up a zone, are denoted as **areas**. Therefore, a zone can also be defined as a set of areas.

For each zone, at creation time, a so called **zone topic list** is added. A zone topic list is defined as a set of 2-tuples (Tp_i, $v_j(Tp_i)$), where Tp_i is a topic defined in the zone, and $v_j(Tp_i)$ is a value between 0 and 1, which represents the weight of the topic, denoted as **topic weight**. The same topic can be defined in different zones. A zone topic list expresses the fact that the same topic can have different weights in different zones. $v(Tp_i,Z_q)$ is the weight of topic Tp_i within zone Z_q.

For each zone, the number of topics, the value of topic weights, and the number of areas composing the zone can change dynamically.

3. System Design

3.1. Choice I: Zones, Actions and User Profiles

In Gugubarra 2.0, we have improved the design of the system to calculate user profiles. We have taken into account not only the duration (*time*) spent by the user on a particular page of the Web site as in Gugubarra 1.0, but also which *actions* were performed by the user, and in which *zones* the actions were performed.

This section gives an explanation on *how zones, actions and user profiles* are related to each other.

For the calculation of user profiles it is taken into *how long* a user visited each zone/page and *which actions* he performed within the given *zone*. A *non-obvious profile (NOP)* is created for each registered visitor by calculating two internal profiles:

- a *Duration Profile DurP(i)*:

$$DurP(i) = \frac{\sum_j \left(duration(P_j) * v(Tp_i, P_j)\right)}{\sum_k duration(P_k)} \qquad \textbf{(1)}$$

, which takes into account the time spent by the user on each page, and

- an *Action Profile ActP(i)*:

$$ActP(i) = \frac{\sum_q \left(\sum_t aw_t \; * \; v(Tp_i, Z_q)\right)}{\sum_s aw_s} \qquad \textbf{(2)}$$

, which takes into account which actions were performed by the users and in which zones.

To compute *ActP(i)* for a topic Tp_i, zone Z_q is determined, where the action occurred. The associated topic weight $v(Tp_i, Z_q)$ is multiplied by the sum for all weights for all occurred actions in this zone. Finally is calculated the sum of all zones, where an action occurred and the zone topic list contains topic Tp_i, divided by the sum off all occurred action weights in session *s*. Note: This is done only for actions, which occur in zones with state ON or *OFF-ACTION-SENSITIVE.*

DurP(i) is computed in a similar way. Each visited page P_j, that contains the topic Tp_i is considered and the time the visitor spent on this page, *duration(Pj)*, is multiplied by its topic weight $v(Tp_i, P_j)$. These values are finally summed up and divided by the total session time, the user spent on the Web site. Note: This is done only for pages, including zones with state ON (see below).

In contrast to the previous implementation, which did not have zones, but only pages, in *Gugubarra 2.0* topics and their weights are associated to *zones* (not to pages as in Gugubarra 1.0). But since HTTP requests are always page based, we were facing the problem that we could determine the time duration the user has spent only on pages, not on zones (see log files section 0).

To take into account this limitation, when calculating *DurP(i)* the value $v(Tp_i, P_j)$ is used, which is a result of a *generated page topic list,* composed by the zone topic lists for each page. For this calculation only zones with state ON are taken into account.

In addition the Web site owner can decide how this topic list is generated by using a max, min or average rule:

- **MAX-Rule**: $x_i = \max(v(Tp_i, Z_q))\forall Z_q \in P_j$
- **MIN-Rule**: $x_i = \min(v(Tp_i, Z_q))\forall Z_q \in P_j$
- **AVG-Rule**: $x_i = avg(v(Tp_i, Z_q))\forall Z_q \in P_j$

In order to create user profiles which take into account *both* the time spent by the user and the actions, formula (3) is used to mix the two profiles together.

The parameters *a* and *b* of the formula are used to customize the ratio between *ActP(i)* and *DurP(i)* in accordance to the Web site owner's needs. More precise: *ActP(i)* and *DurP(i)* should *not substitute* each other. It should be understood as part of the strategy of the owner; He can decide which part is more important for his Web site and adjust the formula accordingly.

The resulting non-obvious user profile is a vector. The values of the vector are denoted by x_i.

$$x_i = a * ActP(i) + b * DurP(i), where\ (a + b) = 1 \qquad (3)$$

The NOP reflects how much time the user has spent on some pages and which zones the user has visited and which actions he has performed.

With the introduction of actions and zones, the calculation of user profiles is therefore more precise than in the previous implementation.

When a new session occurs a profile update can be performed in respect to the formula defined in Gugubarra 1.0 [10]:

$$x_i^{new} = \frac{scount * x_i^{old} + f * x_i^{session}}{scount + f} \qquad (4)$$

By choose of factor *f* as part of the strategy, the importance of new sessions in correlation to old sessions can be manipulated. *Scount* is the number of occurred sessions of the user.

3.2. Choice II: Semantics of Weights

The result of the described calculation is a NOP for each user. To understand how to interpret a NOP, it is necessary to understand how topic weights are calculated. Looking at three formulas for the NOP calculation (1), (2) and (3), the following parameters are used:

- the duration, the user has spent on pages,
- the action weights,
- the parameters *a* and *b* for the customization.

However the semantics of the topic weights isn't defined yet. This semantics of weights is an important design choice we were facing. We have chosen a parametric approach to the semantics definition. We recognized that the semantics of weights depends heavily on the context in which they are used. Different semantics implicate different meanings of the NOP. In other words, a change in the semantics of weights implies a change in the semantics of the NOP values.

Our overall concept is as follows:

- when launching Gugubarra 2.0 for a specific Web site, a specific semantics for topic weights have to be defined. This semantics is then valid for the whole utilization of the tool and cannot be changed.
- The interpretation (meaning) of NOPs is a function of the semantics chosen for topic weights.

The way we calculate user profiles works independently from the semantics of weights chosen, their interpretation instead, is dependent on the semantics.

Following subsections define three pre-define semantics that can be used to customize Gugubarra 2.0. Each of this semantics corresponds to a different scenario of usage of the tool.

3.2.1. Semantics 1: The Web owner controls the information and defines its relevance

Under this scenario, the owner of the Web site has full control of which content information is placed in the Web site. In this case, depending on a business model, goals and expectation the owner *places information* in such a way to reach a certain set of goals.

Examples of possible business goals are:

- Reaching a certain number of reads/downloads for a particular information.
- Reaching a certain number of sales related to a particular information.
- Reaching a certain percentage of registered users who click on a specific link.

The placement of the information in a Web site is an important factor which influences the behavior of the users.

An analogy can be drawn with a supermarket, where for each category of product specific *placement sections* in the supermarket are defined and labeled, e.g. one for fresh groceries, one for drinks, one for meat, etc. Zones capture this *placement* concept for a Web site.

Under this scenario, we define a semantics for the weight of a topic as the *measured importance (between 0 and 1)* for the owner of the Web site of the given topic (type of content).

The decision of the topic weight is directly related to the business model(s) the owner (or business manager) chooses for the Web site. This correlates to the semantics chosen for action weights.

As a practical consequence of using this semantics for topic weights, it is then possible to "classify" Web site visitors with respect to the prioritized set of topics pre-defined by the owner of the Web site.

A **topic weight** therefore describes with **Semantics 1** the relevance of the topic (for a placement) from the owner perspective. The placement and the relevance for the owner define the topic weight. This semantics result in the following **NOP interpretation**: The users NOPs reflect the relevance of the visitors with respect to the scale of relevance the owner has given to the topics covered in the Web site.

3.2.2. Semantics 2: The user discovers the content of a Web site manually on a high granularity level

In this scenario, we consider a Web site, for which we do not necessarily need control, nor have prior knowledge of its content. It can be done by the owner, a content manager or any other professional operator.

In this case, we use the concept of weights to express the manual process of discovering which information is present in the Web site. With this choice of semantics, topic weights are defined as the subjective relevance of the topic. With **Semantics 2** a **topic weight** describes the subjective relevance of the content (*value*) for the topic (*type of content*). This meaning of content relevance therefore defines the topic weight. This semantics result in the following **NOP interpretation**: For the user NOPs we now shift the prioritization of topics from a centralized (owner) top down decision as in Semantics 1 to a content relevance view. Users are then classified with respect to this new topic prioritization. The NOP will reflect the *supposed interest of the user*.

3.2.3. Semantics 3: Weights are defined as frequency of topics

The scenario here is the same for Semantics 2; we consider a Web site, for which we do not necessarily need control nor must have prior knowledge of its content. What is different is that topic weights are defined according to traditional approaches like feature vector and TF-IDF[12], [13]: In this case, the weight reflects the frequency of the topic (here in particular the *term* itself) in a zone.

The topic frequency can be defined in various ways with respect to different context. For example one can check the frequency of the words referring to all words in the zone, or all words in the page, or all words within the Web site, including all pages/documents. We will show in the following one possible solution how to calculate the topic weights with Semantics 3. One advantage of the calculation is that it will also self extract the topics!

We define the *term frequency within a zone* as follows:

$$TF(Tp_i, Z_q) = \frac{|Tp_{i,q}|}{\sum_t |Tp_{t,q}|} \qquad \textbf{(5)}$$

, where $|Tp_{i,q}|$ is the number of occurrences of the considered term (topic) Tp_i in zone Z_q and the denominator is the number of occurrences of all terms in zone Z_q.

To calculate the zone topic list, $TF(Tp_i, Z_q)$ is computed for all words in consideration of a stop list and stemming [11]. Then maximum the *ft* most frequent topics (*topics per zone threshold*) are selected. The owner can define *ft* as part of his strategy. As default we choose *ft*=5. The selected topics compose the zone topic list, each topic Tp_i has the topic weight defined by formula (6). The formula is a normalization and as result the sum of topic weights per zone is 1. It sets $TF(Tp_i, Z_q)$ in relation to the sum of $TF(Tp_i, Z_q)$ for all *ft*-chosen topics of zone *q*.

$$v(Tp_i, Z_q) = \frac{TF(Tp_i, Z_q)}{\sum_{ft} TF(Tp_{ft}, Z_q)} \qquad \textbf{(6)}$$

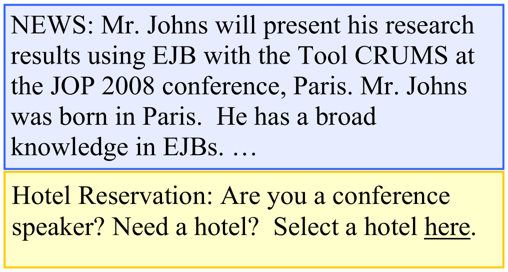
Using Semantics 3, the topic weights and even the topic themselves can be automatically or semi automatically extracted. A semiautomatic function to suggest a zone topic list from the content of *one* area has been implemented, so far. This function can be used by the owner at the creation time of a zone. He will receive a recommendation for a zone topic list, which he can accept, modify or reject.

Therefore with **Semantics 3** a **topic weight** describes the frequency of the topic in the zone. This semantics result in the following **NOP interpretation**: The user NOPs give an indication of which most occurred content the visitor has accessed in a Web site.

3.2.4. An example of different semantics

To use the tool, first a semantics for topic weights has to be defined. This semantics can be then "plugged" into the formulas (1), (2) and (3). Depending on the semantics chosen for weights, an interpretation of the NOPs has to be provided.

Here an example is given on how to create a user profile using the three semantics defined earlier. It is important to notice that any semantics can be defined, not just the three reported in this paper.



NEWS: Mr. Johns will present his research results using EJB with the Tool CRUMS at the JOP 2008 conference, Paris. Mr. Johns was born in Paris. He has a broad knowledge in EJBs. …

Hotel Reservation: Are you a conference speaker? Need a hotel? Select a hotel here.

Figure 1. Example content of a page

Suppose a Web page P_j is given, which includes two zones (Figure 1); Blue zone Z_1 which contains news on a speaker to a conference, and yellow zone Z_2 which contains some information on hotels and the possibility to perform an action, clicking a link to reserve a hotel.

To create a zone topic list, different cases can be used dependent on the semantics of topic weights chosen.

If *Semantics 1* is used for the topic weight, it therefore describes the relevance of the topic for the business goals of the owner of the Web site. Let's assume, because of his business goals, the Web site owner is interested in *speakers* and *conferences* and therefore associate high weights to them.

Although on his Web site he also offers the possibility to reserve a *hotel* this is not really relevant for him, and therefore associate a low weight to it. He might choose the following topics to describe the zones:

- Zone 1: {(Speakers, 0.9), (Conferences 2008, 1.0)},
- Zone 2: {(Hotel, 0.1)}.

Now, let's assume that a user click after 5 minutes on the link in Zone 2, which is pre-defined as the action "Click" that has a high action weight of 6. The NOP profile is calculated with formula (3), a=b=0.5, and the max rule (see section 3.1) for the generation of the page topic list. With Semantics 1 the following Profile results:

NOP 1 = { (Speakers, 0.45), (Conferences 2008, 0.5), (Hotel, 0.1)}.

If the owner uses *Semantics 2* instead, the topic weight describes the relevance of the content for the topic in the zone in the manual content discovery process. For example, this process could result in the following topics and weights for the two zones:

- Zone 1: {(Speakers, 0.3), (Java, 0.5), (Conferences 2008, 0.5)},
- Zone 2: {(Hotel, 0.8)}.

Now in this case the same user action will result in the following NOP:

NOP 2 = { (Speakers, 0.15), (Conferences 2008, 0.25), (Hotel, 0.8), (Java, 0.25)}.

If *Semantics 3* is chosen, a topic weight describes the frequency of the topic in the zone. Therefore the system extracts automatically the most e.g. ft=3 frequent words, that is "Mr. Johns", "EJB", "Paris" for Z_1 and "Hotel", "Conference", "Speaker" for Z_2. This result with formula (5) and (6) in the following zone topic list:

- Zone 1: {(Mr. Johns, 0.33), (EJB, 0.33), (Paris, 0.33)},
- Zone 2: {(Hotel, 0.61), (Conference, 0.2), (Speaker, 0.2)}.

What is apparent here is that the choice of the topic names is no more a subjective choice, but rather a result of word counting. Now in this case the same user action will result in the following NOP:

NOP 3 = {(Mr. Johns, 0.17), (EJB, 017), (Paris, 0.17) (Hotel, 0.61), (Conference, 0.2), (Speaker, 0.2)}

The example is only meant to show the influences of the three semantics on the NOP; not to show a realistic case. The calculation is the result of a ficticous 5 minute user action, and does not reflect a real NOP, that is calculated after many visits, actions, sessions, and months.

What is important though is to notice that weights and even the topic themselves depend on the semantics.

For example, the weight of topic hotel in NOP1 is very low, as hotel is of no interest to the owner. Whereas hotel has a high weight in NOP2, because the action took place in zone2 on the content hotel.

The semantics defined for weights are not an indication of the importance given to a topic by a *visitor* of the Web site. This user aspect plays as well an important role.

Therefore, in Gugubarra we allow users to give a **feedback**. This feedback mechanism defined in [10], [6], is triggered due to certain criteria, where upon the visitor is requested to give his feedback.

4. How to create and use Topics, Zones and Actions

Here a short snapshot is given, how topics, zones, and actions are created in a Gugubarra 2.0 application.

The owner creates globally topics and actions. He inserts names and optionally a description for a set of topics and actions. Further he associates global weights to each global action; dependent on the importance of an action.

The owner has two ways to create zones. On the one hand, he can define all zones globally. Therefore he inserts names and optional a description for a set of zones. For each zone he can choose a subset of the predefined global topics, and define topic weights. Then he loads a page that should be *zoned*, into the application. After marking an area, the owner can associate the area to a specific already global defined zone.

On the other hand, he can add the global zones, when he first needs them. Therefore, after marking an area, he can create a new zone instead of selecting an existing one.

The workflow and its implementation have one impact: A zone topic list can exist globally, without any associated page! Further, if all pages belonging to a zone are deleted, the zone topic list can still stay. It can be used to attach pages again. If the zone is no longer of interest to the owner, he can delete the global zone.

The precondition is an already existing Web site, composed of different pages. The result is that each page (or parts within) is associated to different zones.

The global actions are used to classify Hyperlinks. *Hyperlinks*, that are part of a zone area, can be associated to a global action. Thus, an action can occur in a zone, being performed by a Web visitor.

5. Implementation

We now focus on the implementation aspects of zones and their applicability. We present the general architecture, applicability and workflow of Gugubarra 2.0 and describe details of the implementation focusing on the zone module.

5.1. Architecture

In the age of Web 2.0 we hardly discover static Web pages. Today's Web sites consists particularly of dynamically generated Web pages that are built up at the moment of request.

To manage such dynamic Web pages usually a content management system (CMS) is used, that coordinates the workflows of a cooperative web-based working process and assists with the development of content.

When we designed the zone module we first thought of implementing zones as a plug-in module for a specific CMS. We abandoned this idea for the moment, since we wanted to ensure a system-independent functionality for an *experimental prototyping*. The application Gugubarra 2.0

including the zone module is now a .Net standalone application. Microsoft Visual Studio and C# were used for the development.

Gugubarra 2.0 works on the HTML pages transferred by the Web browser. Thus we can test it on all Web sites (whether static or dynamic), because we work on the output and don't need an access to the Web servers.

5.2. Applicability

Gugubarra 2.0 is applicable to both, a new Web site and to an existing one. We look at the two cases in the next subsections.

5.2.1. Creating Zones for a new Web site

When creating a new Web site from scratch the following preconditions for the zoning module need to be satisfied:

- The *last modification time* must be included into each Web page. For example using an invisible HTML comment.
- *Ids for the elements*: text, graphic, link must be inserted. For example using a *<div id=…>* tag.

The requirement for the whole application is obviously a logging system to create NOPs. From the log files sessions are identified and associated to registered users.

When mining the sessions, the information that is extracted is as follows: which pages each user visited, how long he stayed, and which actions he has performed.

5.2.2. Creating Zones for an existing Web site

If one intends to apply the Gugubarra tool to an existing Web site, there are two cases possible:

- If a *Content Management System* is used, then the content is normally already split in small pieces. In this case the administrator has a small effort including some lines of code for the time and identification.
- In case the Web site is composed of *static pages*, then the administrator will need some time to include the HTML tags. In the worst case each static page will have to be modified.

5.3. Workflow

The general workflow for Gugubarra 2.0 to create zones and NOPs, depicted in Figure 2, is as follows:

1. The Web server creates a specific HTML page.
2. The HTML page is loaded into the application.
3. Owner creates zones and saves them to the database.
4. Visitors navigate through the Web site.
5. Log files of the sessions are stored into the database.
6. The application reads the session logs and zones and
7. it calculates out of this information the NOPs.
8. Visitors give their feedback (details see [6]).
9. Measurement of e.g. difference NOP vs. feedback.

The details for each step are explained in section 0.

5.4. Code ramble

In the **first** process of the workflow the explicit URL created by the Web server for each page is needed. The user has to copy the URL corresponding to the page that should be zoned into the text field on the upper right side of the Gugubarra application (see GUI shown in Figure 3).

By clicking on the "GO" button the **second** process starts.

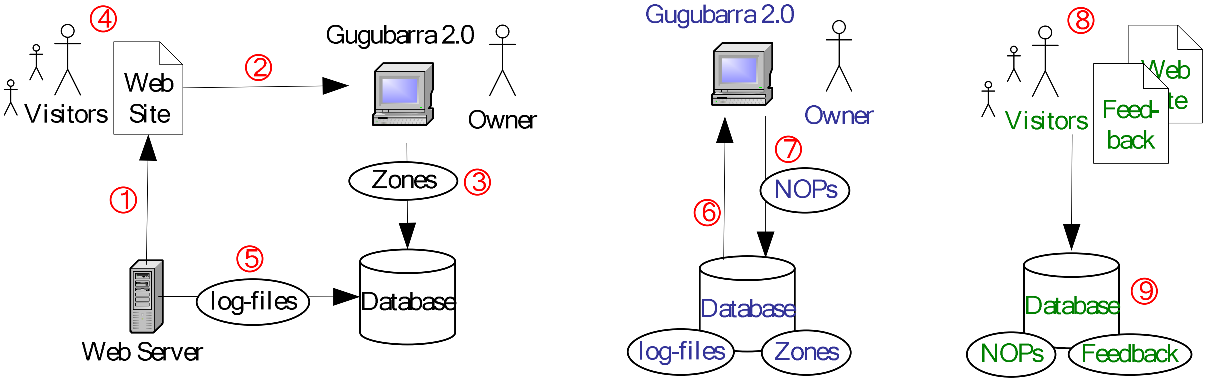


Figure 2. Workflow Gugubarra 2.0

The whole application is based on the C#-Class *System.Windows.Forms.Form*. It contains a window that visualizes the page of the URL. That window is an instance of the C#-Class *System.Windows.Forms.WebBrowser* and thus similar to a simple Web browser. The name of the object is in the following *pageWindow*.

With the subsequent code the page can be saved in the object *HTMLDoc*. Thus the HTML code is always accessible as an HTML document object. *this* refers to the application form itself.

```
HtmlDocument HTMLDoc =
  (HtmlDocument) this.pageWindow.Document;
```

Using the *HTMLDoc* object, the tags of the document can be parsed and navigated through. To implement the zones, *elements* including text, graphic, and links have to be *identified* (see 5.2.1). We therefore search tags including ids. Furthermore, we need in the current implementation the *time of the last modification of the page* for an easy way to recognize changes.

Note, these preconditions are necessary, as long as we work on the output of Web servers. Of course the Gugubarra zoning concept can be implemented without any difficulty as CMS plug-in. In this case, the source of content is directly accessible to include zone ids.

The following code is a snippet to find with a regular expression hyperlinks:

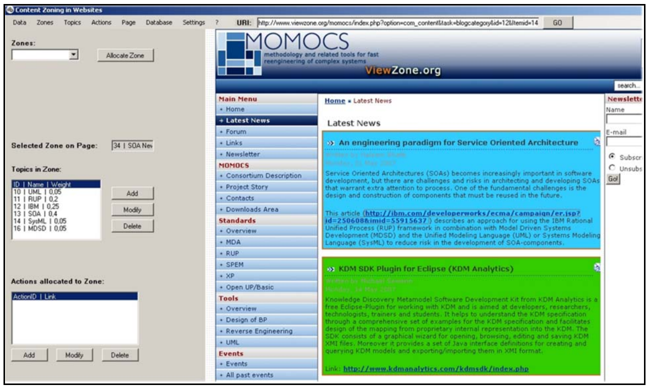
```
/* GET HTML CODE */
foreach (HtmlElement element in
HTMLDoc.GetElementsByTagName("HTML")){
    HTMLPageCode = element.OuterHtml; }

/* IDENTIFY LINKS */
Regex reg2 = new Regex("href\\s*=\\s*(?:\"(?<1>[^\"]*)\"|(?<1>\\S+))",
RegexOptions.IgnoreCase | RegexOptions.Compiled);
Match m2 = reg2.Match(HTMLPageCode);
```

Focusing on process **three**, we describe in the following how the owner can *create zones* and *save* them to the database (here DB2).

A position within the document is first selected; we then identify the enclosing element ids (articles, graphics, and hyperlinks) and mark them as a colored area. This is done by modifying the HTML code of the object *HTMLDoc*. Therefore we added an *event Handler* to each article, graphic or hyperlink element with the event *MouseDown* to recognize mouse clicks. With the method *getElementsByTagName(String tag).OuterHtml* we have access to the code of a specific tag and can add a color to mark an area. After marking an area, the user can associate the area to a specific zone in the left side of the application. He can choose an already global defined zone or create a new zone as described in section 4. The area adopts then the colour of the zone. Automatically all element ids belonging to the zone are stored into the database (referring to the zone id).

The owner can then proceed to select areas and assign them to zones. In Figure 3, for example, two separate zones are visualized as blue and green blocks respectively.



URI: http://www.viewzone.org/momocs/index.php?option=com_content&task=blogcategory&id=12&Itemid=14

Zones:

Selected Zone on Page: 34 | SOA Nev

Topics in Zone:

ID	Name	Weight
10	UML	0,05
11	RUP	0,2
12	IBM	0,25
13	SOA	0,4
14	SysML	0,05
16	MDSD	0,05

Actions allocated to Zone:

ActionID	Link

Latest News

» An engineering paradigm for Service Oriented Architecture

Service Oriented Architectures (SOAs) becomes increasingly important in software development, but there are challenges and risks in architecting and developing SOAs that warrant extra attention to process. One of the fundamental challenges is the design and construction of components that must be reused in the future.

This article (http://ibm.com/developerworks/ecma/campaign/er.jsp?id=250608&imid=55915637) describes an approach for using the IBM Rational Unified Process (RUP) framework in combination with Model Driven Systems Development (MDSD) and the Unified Modeling Language (UML) or Systems Modeling Language (SysML) to reduce risk in the development of SOA-components.

» KDM SDK Plugin for Eclipse (KDM Analytics)

Knowledge Discovery Metamodel Software Development Kit from KDM Analytics is a free Eclipse-Plugin for working with KDM and is aimed at developers, researchers, technologists, trainers and students. It helps to understand the KDM specification through a comprehensive set of examples for the KDM specification and facilitates design of the mapping from proprietary internal representation into the KDM. The SDK consists of a graphical wizard for opening, browsing, editing and saving KDM XMI files. Moreover it provides a set of Java interface definitions for creating and querying KDM models and exporting/importing them in XMI format.

Link: http://www.kdmanalytics.com/kdmsdk/index.php

Figure 3. GUI of the application

As content is dynamic, the application checks if all elements belonging to the zones of the current page still exists. If not the reference *element - zone* will be deleted. To implement a fast prototype, we have chosen to check changes only if changes have occurred. That's why we need the last modification time, as described in 5.2.1.

The three described processes are all related to how the owner can create zones. **Process four to seven** are the general workflow to calculate the non-obvious user profiles of the visitors:

Visitors navigate through the Web site. The Web server observes them and creates log files. From the log files we can identify sessions and associate them to registered users through a user id. This can be realized by e.g. including ids for sessions and users as part of the URL. Note, that this is not warranted by all Web sites. If session ids are hidden and not URL included, they must be logged by the portal in an additional logging process. Gugubarra 2.0 can process log files, which include entries similar to the following one, here taken from Apache Server:

```
89.172.168.227 - - [30/May/2007:08:55:03 +0200] "GET
/abilities/layout/style.css HTTP/1.1" 304 - www.viewzone.org
"http://www.viewzone.org/abilities/comments&suggestions/?tmid=4&UID
=65&SID=5d2651a7f132f1ee79bef720a9a1c630" "Mozilla/4.0
(compatible; MSIE 7.0; Windows NT 5.2; .NET CLR 1.1.4322;
InfoPath.1; .NET CLR 2.0.50727; .NET CLR 3.0.04506.30)" "-"
```

The log files can be imported and analyzed by Gugubarra. It collects from the log files the information which pages the user visited, how long he stayed, and which actions he has performed.

To calculates the NOPs, Gugubarra reads some data out of the database, like user data, zones of the visited pages, action weights, parameters for the specific strategy of the owner (*a*, *b*, rule, etc. see section 3) and the old NOP. When the new NOP is calculated with formulas (1) to (3), it will be saved and the old NOP is stored in a history database.

The NOP is in Gugubarra 2.0 not calculated after each session, because then we would need access to the Web Server or process the log files permanently. Instead the owner can choose when to update the non-obvious profiles. A NOP history is done for each time the profile is updated.

Process eight and nine play a role when a user feedback is given. Then some measurements can be taken into account like the difference between NOP and Feedback.

Finally, a few words about the **database schema**: The main table is *user* where each user has an entry including his *user id* and some personal information. The table *topic* with the primary key *topic id* holds all topics. The table *NOP* has a foreign key to the *user id* and to the *topic id*. Hence several entries in *NOP* define one user NOP. The number of entries depends on the number of topics, which the user had “touched”. “Touched a topic” means the user has visited a page that includes zones related to the topic. Therefore the user NOP is dynamic; topics can come and go. To keep the referential integrity, all entries in *NOP* referring to topic *i* should be deleted, when the target of the foreign key (*topic id i*, the global topic in table topic) is removed. Further it means if a new global topic is added, we don’t change the user NOP until the user has “touched” that new topic.

6. Related work

A unique feature of Gugubarra is the combination of the NOPs and the Feedback Profiles with custom rules. This is missing in systems such as the one of Acharyya et al. [1] or of D’Ambrosio et al. [2].

Most of the research done in the field of *Web user profiles* focuses on helping the visitor to search, recommend pages or to adapt systems. Instead *Gugubarra* should primary support the Web site owner with a flexible system; a strategy can be customized and several semantics can be plugged in, correlating to different business strategies. Further Gugubarra could support systems for behavioral targeting, e.g. by offering personalized e-services to the visitors.

With the introduction of zones and actions, we do not only rely on the recorded click stream as in [14].

Through our zoning concept, we can exclude the unnecessary content like possibly navigation menu, advertisement, contact information or copyrights. Thus it is similar to the aim of Gasparetti et al. [4], who analyze parts of the HTML of visited pages to identify user needs. However they have no possibility to customize a strategy and a specific semantics for different business goals.

Most approaches build user profiles using the terms from TF-IDF document vectors such as Mostafa et al. [9], as we do in Semantics 3. However our profile is more flexible and can be used for a wider purpose.

7. Acknowledgments

Our special thanks to Alexander Gerlach, who has implemented the zones module and to the rest of the Gugubarra Team: Clemens Schefels, Karsten Tolle, and Naveed Mustaq.

8. References

[1] S. Acharyya, J. Ghosh, *Context-Sensitive Modeling of Web-Surfing Behaviour Using Concept Trees*,.WEBKDD, 2003.

[2] B. D’Ambrosio, E. Altendorf, J. Jorgensen, *Probabilistic Relational Models of On-line User Behavior*, WEBKDD, 2003.

[3] DBIS J.W. Goethe University, *CeBIT Presentation for Gugubarra 1.0.*, web resource: http://www..dbis.informatik.uni-frankfurt.de/ news/?mode=art&l=e&aid=2&tmid=3&smid=8.

[4] F. Gasparetti, A. Micarelli, *Exploiting web browsing histories to identify user needs*, IUI, pp 325-328, 2007.

[5] N. Hoebel, S. Kaufmann, K. Tolle, R. V. Zicari, *The Design of Gugubarra 2.0: A Tool for Building and Managing Profiles of Web Users*, IEEE/WIC/ACM, International Conference on Web Intelligence, Intelligent Agent Technology and Data Mining, Hong Kong, 2006.

[6] N. Hoebel, S. Kaufmann, K. Tolle, R. V. Zicari, *The Gugubarra Project: Building and Evaluating User Profiles for Visitors of Web Sites*, IEEE Workshop on Hot Topics in Web Systems and Technologies, Boston, USA, 2006.

[7] N. Hoebel, R. V. Zicari, *On Clustering Visitors of a Web Site by Behavior and Interests*, Advances in Soft Computing, Springer, AWIC, June 27-29, Fontainebleau, France, 2007.

[8] D. Kahneman, A. Tversky, *Rational Choice and the Framing of Decisions*, Journal of Business, vol. 59, no. 4, 1986.

[9] J. Mostafa, S. Mukhopadhyay, W. Lam, M. Palakal, *A Multi-level Approach to Intelligent Information Filtering: Model, System & Evaluation*, ACM Transactions on Information Systems, 15(4), pp 368-399, 1997.

[10] N. Mushtaq, K. Tolle, P. Werner and R. V. Zicari, *Building and Evaluating Non-Obvious User Profiles for Visitors of Web Sites*, IEEE Conference on E-Commerce Technology (CEC 04), San Diego, California, USA, 2004.

[11] M.F. Porter, *An algorithm for suffix stripping*, Program, 14(3), pp 130-137, 1980.

[12] G. Salton, *The SMART retrieval system - experiments in automatic document processing*, Prentice Hall, Englewood Cliffs, New Jersey, 1971.

[13] G. Salton and C. Buckley, *Term weighting approaches in automatic text retrieval*, Technical Report 87-881, Computer Science Dept., Cornell University, USA, 1987.

[14] Qing Wang, Dwight J. Makaroff, H. Keith Edwards, *Characterizing Customer Groups for an E-Commerce Website*, Proceedings of EC '04, ACM Press, New York, 2004.