EnviroInfo 2010 (Cologne/Bonn)
Integration of Environmental Information in Europe
Copyright © Shaker Verlag 2010. ISBN: 978-3-8322-9458-8

Enhancing Sustainability of the Software Life Cycle via a Generic Knowledge Base

Julia Fischer, Stefan Naumann, Markus Dick
Trier University of Applied Sciences, Umwelt-Campus Birkenfeld
Campusallee, D-55768 Hoppstädten-Weiersbach, Germany
j.fischer@umwelt-campus.de
s.naumann@umwelt-campus.de
m.dick@umwelt-campus.de

Abstract

Information technology plays a double-edged part when facing the challenges of sustainable development: On the one hand IT can improve processes, dematerialize physical goods, etc. On the other hand, IT itself is consuming more and more energy and natural resources. In our paper, we present an approach for a generic knowledge database, which helps software developers, managers of data centres, users, and other actors within the hardware and software life cycle in producing, maintaining, and using more sustainable and "greener" software. In detail, we present a model for appropriate articles, and overall system architecture. With some user scenarios we demonstrate, how our system is working.

1. Introduction

Issues like global warming, climate change and sustainable development in all areas of the economy, industry, and the daily life become more and more important. Green IT can help to prevent waste by selling software as downloads (softwareload 2010) or reusable USB sticks with live recordings after concerts (Posdziech 2009). Additionally, more and more firms develop hardware in a resource saving way, by reducing the necessary production raw material and usage of safer chemicals. One pioneer of Green IT is Fujitsu, who started a recycling program already in the late 1980ies and developed the first Green PC. With their "proGREEN IT Mission1" in the year 2009, they started an initiative to analyze IT systems to inform the IT managers about energy consumption and potential savings (Fujitsu 2009).

Apple is another example firm, who started to develop Green IT and to inform users about the environmental footprint of their products and how the values of toxics, packaging, energy efficiency and recycling have changed over years (Apple 2010).

Information technology in the area of software engineering may help to decrease the energy consumption by up to 20%, but the savings depend on the different framework conditions. However, in the worst case, information technology may cause an additional impact on the environment of 30% (Hilty 2008, page 18).

The challenge for an environmentally friendly usage of ICT to reduce the carbon footprint and other environmental impacts has to be resolved. One contribution is to develop and use software itself in a resource-saving way (Sustainable IT), and another opportunity is to use the software to save energy (Sustainability through IT) (Naumann 2008).

In our paper, we present a model of a generic knowledge base that provides guidelines, tips, and hints to users, which help to reduce energy consumption during the whole software life cycle.

2. Related Work

Until now, there are many publications available regarding ICT and sustainability. Unfortunately these publications are focusing mainly on the hardware aspects of ICT by the means of Green IT. Just a few publications exist that focus especially on the development related software aspects of ICT. However, some publications are available that examine the impacts of software usage on sustainability.

Regarding the architecture of our knowledge base, Schreiber (2009) presents a database and a web based retrieval system for substantial scientific publications in the field of environmental sustainability. Filetti et al. (2008) present a web based environmental information system that combines environmental information, which will be otherwise offered in several short articles spread over the web. The articles are thematically ordered in a hierarchy and full text searchable.

The information stored in the knowledge base may be available in different levels of detail according to the expected audience. Hilty (2008) presented some recommendations for actors in decision making processes, including users and software producers, regarding sustainable development. Arndt et al. (2009) propose design principles elaborating on software architecture that potentially lead to more sustainable information and communication systems. Abenius (2009) suggests a categorization of green software and examined how the amount of HTTP requests and their size can be minimized by configuring a web server properly. Amsel and Tomlinson (2010) present a software tool that estimates energy consumption of deployed software to support users in making informed decisions about the software they use. Dick et al. (2010) present principles and suggestions for web developers, administrators and users that enable them to develop, operate, and use websites in an environmentally sustainable way.

3. Problem Description

As far as we know, there is no generic platform or knowledge base with a collection of guidelines, tips, and hints in the area of sustainable information technology. One can find much information material on Green IT with different tips to issues of reducing energy consumption and sustainability in the area of ICT hardware, like Behrendt and Erdmann (2009) and websites with general rules, like Yahoo! (2010). Another option for general software hints is the MSDN help desk function (Microsoft Developer Network 2010), which is e.g. included in the integrated development environment “Microsoft Visual Studio”.

Hence, we propose a knowledge base that has the objective to support people in optimizing their software products, so that negative or positive impacts, which result from the entire lifecycle of the software product, are mitigated or amplified. This lifecycle is inspired by Life Cycle Thinking, which is commonly known from eco-design principles and can be curtly described in terms like “from cradle to grave” (Tischner et al. 2000, pages 13-14). It consists of the phases: development, acquisition, distribution, deployment, usage, maintenance, deactivation, and disposal. The lifecycle phase from which the highest impacts on sustainability are expected is the usage phase. Hence, it is necessary to provide users with knowledge that enables them to optimize these impacts. Nevertheless, it is necessary to provide knowledge to actors involved in the other phases of the lifecycle, which supports them in anticipating the impacts expected from the usage phase and which supports them in optimizing the software product. The knowledge is provided in articles, which can be e.g. rules of thumb, hints, best practices, suggestions, recommendations, guidelines, or checklists. These articles are published in several skill levels, to consider the expertise of the targeted actors.

4. Requirements

As discussed in the problem description, the system provides support to resolve different problems, to submit requests, and to supply the user with articles.

Our article system is designed especially for the three actor roles: developer, administrator, and user. The actor roles are derived from the roles of the software life cycle and can be expanded to more detailed roles

like e.g. web developer, database administrator or consultant. Each of these groups has different demands, so they can search for articles in corresponding categories. The knowledge base should be generic and extensible. Later, it should be possible for the users to write their own articles and to rate and edit the whole community's articles.
Some usage scenarios were developed in order to get an impression of the necessary requirements, which the model has to fulfill. The scenarios are separated into two categories: the user and the author perspective. The scenarios from the user perspective show the different types of user roles, from the web- and software developer up to the administrator, advisor, and end user. Furthermore, they depict how the groups interact with the system and what their expectations are. From the author perspective is described, how the system will be filled with advice. The following sections describe some usage scenarios and the resultant requirements for each case in a concrete way.

4.1 Usage Scenario 1: A Developer's Perspective

Mister Smith is software developer and wants to have articles for the efficient usage of his software. He works at a company that develops software solutions for business customers in the area of logistics. He wants to use the article system daily, to get help on specific topics. During his search, he wants to differentiate between various categories, like the different phases of the software life cycle. He especially needs some functional blocks to build logistic methods to develop own software in a resource saving way. Additionally he expects suggestions to related problems.
Requirements: The article system has to provide the search in different categories and phases of the software life cycle. The search does not only take place with the exact search entry but with related words as well.

4.2 Usage Scenario 2: A Consultant's Perspective

Miss Stone works as a consultant in several different software projects with varying responsibilities. She wants to adapt her consultation work to the different scopes of duties. That is why she sets up more than one user profile. The search can be enclosed to a selected area and will be realized more efficiently. Furthermore she expects the possibility to save articles in her different profiles, to have them available in a faster way.
Requirements: Different user profiles are necessary and articles have to be stored in the individual profiles.

4.3 Usage Scenario 3: An End User's Perspective

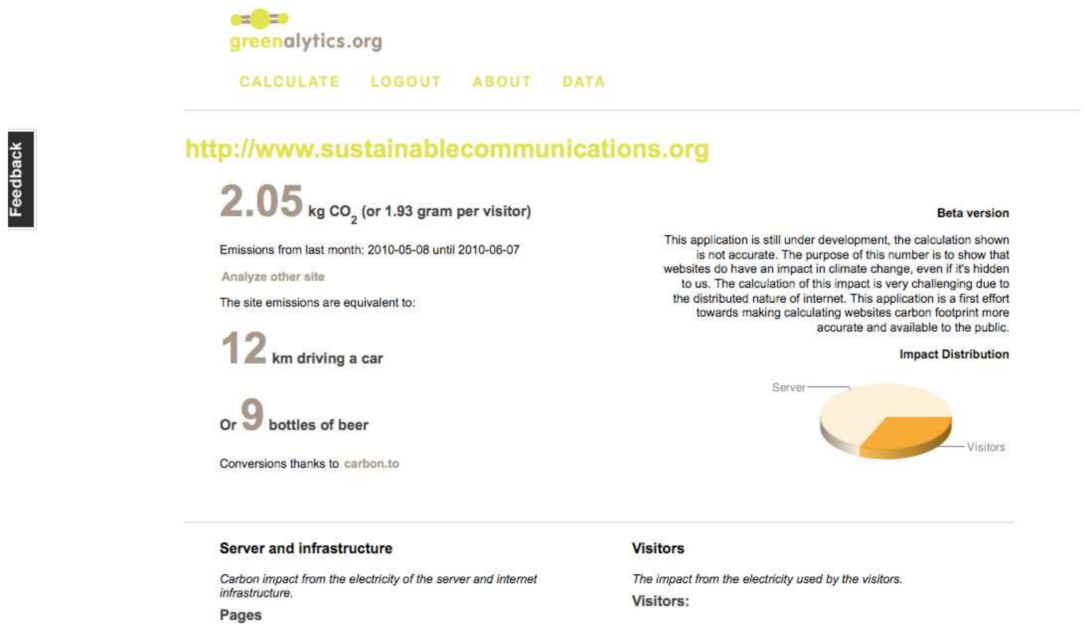
Sandra Miller studies environmental biology at the University of Bremen. She often uses the Internet for researches for term papers and seminar theses. She wants to use the article system, to get help on methods of a resource saving search in the Internet. Sandra expects different categories, which she can choose and the opportunity to select different skill levels according to her expertise.
Requirements: The article system has to provide different categories and to allow a variety of skill levels from the article reports.

4.4 User Story 4: An Author's Perspective

Ann Peach works at a research institution for software systems in the special field of Green IT. During her work, she has collected many different tips and hints referring to sustainability and information technology. She wants to publish these in an article in an appropriate way. Miss Peach expects a corresponding form to enter and edit her tips.

Requirements: There has to be a possibility to publish articles via a simple form.

5. Model Overview



http://www.sustainablecommunications.org

2.05 kg CO_2 (or 1.93 gram per visitor)

Emissions from last month: 2010-05-08 until 2010-06-07

Analyze other site

The site emissions are equivalent to:

12 km driving a car

Or 9 bottles of beer

Conversions thanks to carbon.to

Beta version

This application is still under development, the calculation shown is not accurate. The purpose of this number is to show that websites do have an impact in climate change, even if it's hidden to us. The calculation of this impact is very challenging due to the distributed nature of internet. This application is a first effort towards making calculating websites carbon footprint more accurate and available to the public.

Impact Distribution

Server and infrastructure

Carbon impact from the electricity of the server and internet infrastructure.

Pages

Visitors

The impact from the electricity used by the visitors.

Visitors:

Figure 1: Model of the Article System

The main components of the model (see figure 1) are a database for managing articles, a search engine, a news service, and the different user roles. There are different sorts of structures like thesauri, indices, and keyword lists, to organize the entries in the database. Thus, it is possible to search for similar and related keywords to get more results.

The information is structured in an independent data exchange format. In the overview of the model, one gets a first impression of the data structure by means of a domain data model, which is described in more detail in section 7. The consumers are differentiated in registered and unregistered users. Registered users have more options to interact with the system. They can search via the Internet browser or an application in an independent development environment. All users have two different possibilities for their search entry. They can use the standard search with a simple search field, where they can combine their search words with words like "and", "or", and "not" or the advanced search with e.g. different categories and the favorite language. The usage of the stored profiles enables registered users to use a faster and more accurate search.

The profiles are necessary for users, who want to search regularly for a special problem considering the software lifecycle or dependent systems. With the profile, they save the time filling out the attributes in the advanced search. The mentioned usage scenarios from section 3 have shown that the use of the article system differentiates according to the user and the expected result.

Furthermore, registered users can take the role of an author. They can write own articles to fill the database. Before the article is usable for other users, it is reviewed by the authority board to guarantee high quality and correct information. Additionally, they can edit and expand articles from other users or add translations to enhance the usability of the system. There is also a rating system to illustrate the suitability

for a special problem. Besides the authors, there is another possibility for inputting new articles. Maybe later, a crawler will be able to search for articles on other web sites.
The news service offers a RSS feed and a mail service. The consumer will be informed about latest entries or news considering the article system. Moreover, they can contact other users to ask questions to get more detailed information, or to solve problems when using an article.

6. System Architecture

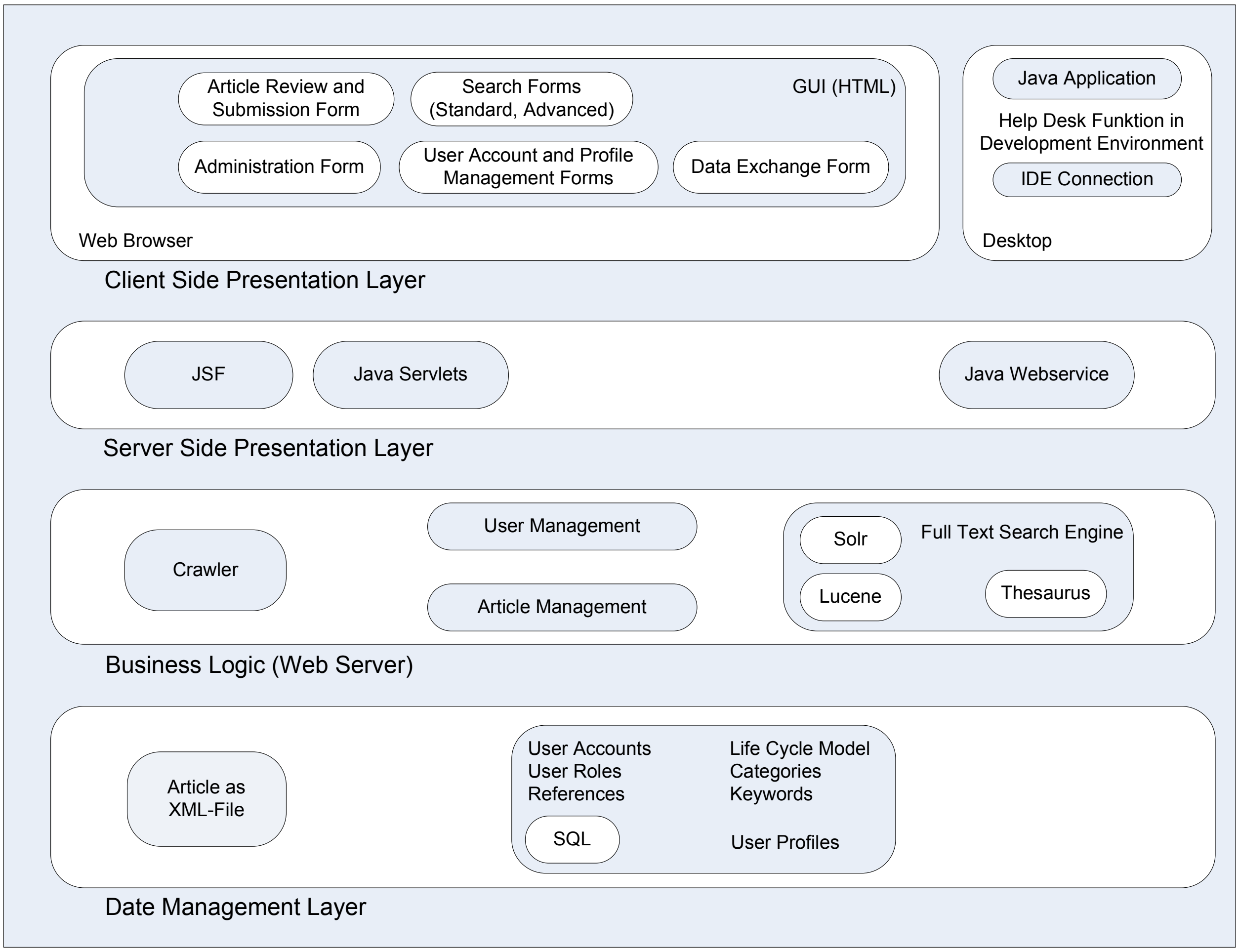


Figure 2: System Architecture

The system architecture (see figure 2) is planned as a four-tier architecture consisting of a client side presentation layer, a server side presentation layer, the business logic and a data management layer. The first layer includes two parts according to the presentation. On the one hand, users can search for articles via web browsers and on the other hand via plug-ins for integrated development environments.
The graphical user interface of the system consists of the different forms depending on the usage. The standard search form only has a simple search field. The advanced search enables the selection of different categories and user roles. The article review and submission form is especially for the authors and the authority board members. The authors send their written article to the authority board to let them approve of it. Afterwards, they get the article back either with a new status, like e.g. published, accepted, or dismissed, together with correction instructions. With the administration forms, the administrator can manage users, articles, access control rules, and maintenance of the knowledge base system. The user account and

profile management forms give the users the opportunity to login and register in the system and to create and edit their profiles. The data exchange form is necessary for the storage and upload of articles. The users can choose between different formats e.g. PDF, plain text or XML.

The server side presentation layer includes JavaServer Faces to build the server-side user interfaces, Java Servlets and a Java Web service to realize the usage of the system via the IDE.

A crawler, to search for articles on other websites, the full text search engine, the user management, and the article management are parts of the business logic. The full text search will be realized with parts of the open source project Lucene Java, a Java library to create and search through indices, and Solr, a search server for full text search. Both technologies will be complemented by a thesaurus to enable the search for related problems, similar spellings, and different words with the same meaning, like homepage, website, and web page.

The last layer is responsible for the data management and consists of information in the data format XML, a relational database, which includes the profiles of the different users, user accounts, user roles, literature references, keywords, categories, and the life cycle model.

7. Structure of the Knowledge Base Article

Articles will be structured with XML and consist of title, reference, keywords, authors, categories, life cycle phases, user roles, and actual article content. Furthermore, the articles have attributes like ID, publish and edit date, skill level, and rating. The ID will be generated via Universal Unique Identifier (UUID) (Leach et al. 2005) to prevent double IDs and to ease the coordination of the articles without a centralized authority.

If other users make changes in the article, the edit date will be updated and the author element gets a new entry. The skill level is important for giving the user an impression of the way the article is described according to the content structure e.g. as step by step instruction or only as general overview. The users have the possibility to rate an article, to let the others know how helpful it was.

One article can exist in different translations, which are all stored in one file. Depending on the favorite language and the search entry of the user, one gets the matching results. The elements category, life cycle phase, user role, reference, and keyword have connections to the database. The elements themselves in the XML-file only include the IDs. There exist tables with the belonging entries of the life cycle phases like development, usage, and maintenance or of the categories like server, cache, and image. The reference includes the overview of the referenced literature and sources, which were used to write the articles. The table of keywords implies a collection of the different keywords from the existing articles. The authors can choose them for their articles and input new ones to expand the collection. In the UML-diagram (see figure 3) below, all elements are listed.

We decided to use XML, to have an independent data exchange format to ease the portability and adaptability of articles. XML helps us to structure and extend our data to have a machine-readable foundation for our knowledge base.

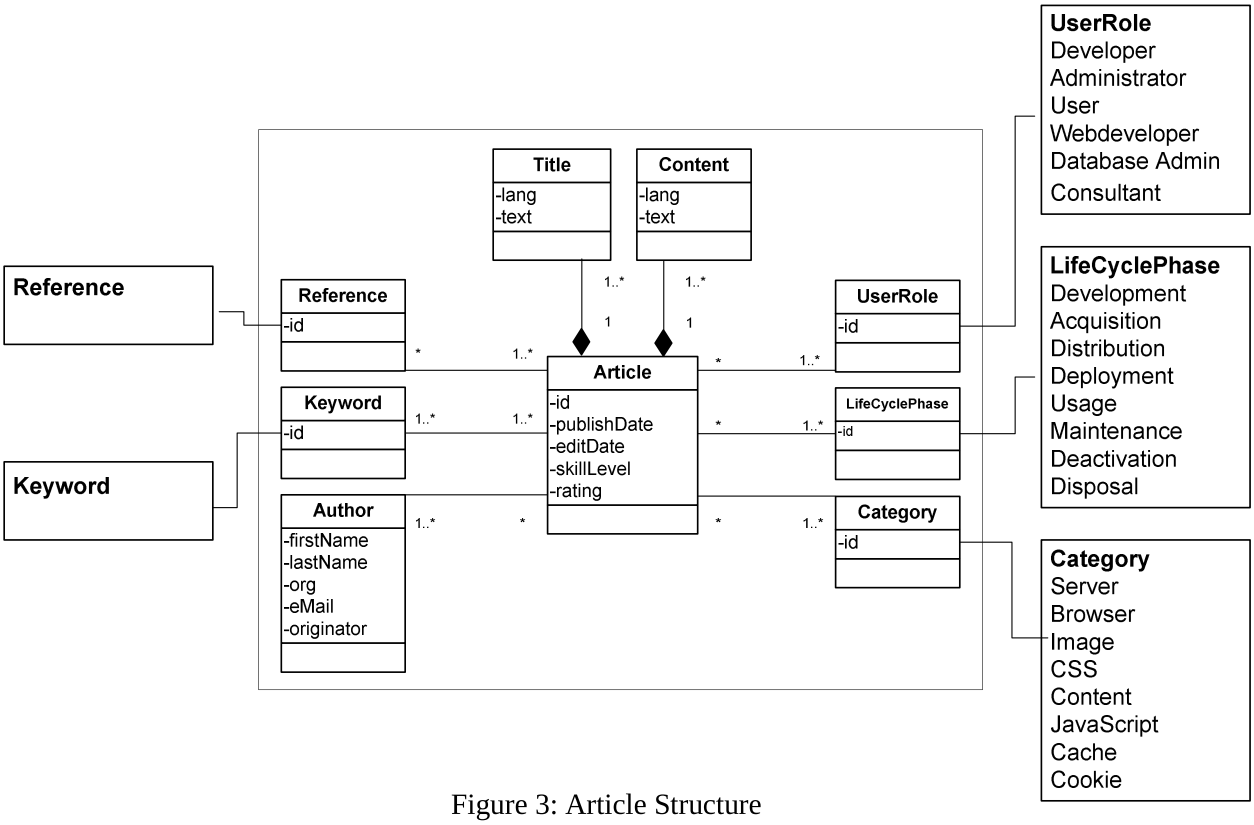


Figure 3: Article Structure

8. Exemplary Articles

The principles mentioned in Dick et al. (2010) are one basis for the development of the generic knowledge base. In order to get a better understanding of the articles and their structure, two of them will be presented in short in the corresponding format in this section. The first article is especially for web developers who can minimize and optimize their source code with simple transformations of color presentations and deleting of unnecessary white space and blank lines.

The second article deals with the configuration of the browser and is useful for end users. They can adjust settings in the browser to reduce the network traffic by using advertisement blockers and supporting special compressions in the browser.

8.1 Example Article 1: Optimization of CSS

Article	id	001
	publishDate	2010-03-25
	rating	very useful
	skillLevel	general
Title	Optimization of CSS	
Content	With minimizing and optimizing CSS it is possible to reduce HTTP requests. There are tools like CSSTidy, which can optimize CSS code in different ways. One mentioned technique is to use shorter definitions for colors: {color:#ffcc00;} can be written as {color:#fc0;}	

<table>
<tr><td></td><td>Another way to reduce the file size is to delete unnecessary whitespaces, strings and blank lines. The strength of the compression depends on how readable the code has to be after the compression. If readability is not important, the whole code can be written in one line.</td></tr>
<tr><td>UserRole</td><td>Web Developer</td></tr>
<tr><td>LifeCycle-Phase</td><td>Development</td></tr>
<tr><td>Category</td><td>Source Code</td></tr>
<tr><td>Author</td><td><table>
<tr><td>firstName</td><td>Arno</td></tr>
<tr><td>lastName</td><td>Nym</td></tr>
<tr><td>org</td><td>Institute for Software Systems</td></tr>
<tr><td>eMail</td><td>greensoft@umwelt-campus.de</td></tr>
<tr><td>originator</td><td>yes</td></tr>
</table></td></tr>
<tr><td>Keyword</td><td>CSS, Color, HTTP requests, Compression</td></tr>
<tr><td>Reference</td><td>King, A., 2008. Website Optimization. 1st ed. Sebastopol: O’Reilly Media</td></tr>
</table>

8.2 Example Article 2: Configuration of the Web Browser

<table>
<tr><td>Article</td><td><table>
<tr><td>id</td><td>002</td></tr>
<tr><td>publishDate</td><td>2010-05-17</td></tr>
<tr><td>rating</td><td>useful</td></tr>
<tr><td>skillLevel</td><td>very general</td></tr>
</table></td></tr>
<tr><td>Title</td><td>Configuration of the Web Browser</td></tr>
<tr><td>Content</td><td>Web users can contribute a large part to the reduction of network traffic and power consumption by configuring the caching capabilities of their web browsers in a way that far future expiry dates can take effect, or by installing web browsers that are fully compatible with GZIP based HTTP compression.

- configure large caches in Web browsers - do not clear caches during browser shutdown
- use advertisement blockers to block - use browsers supporting GZIP compression
advertisement images and Flash</td></tr>
<tr><td>UserRole</td><td>End User</td></tr>
<tr><td>LifeCycle-Phase</td><td>Usage</td></tr>
<tr><td>Category</td><td>Cache, Browser</td></tr>
<tr><td>Author</td><td><table>
<tr><td>firstName</td><td>John</td></tr>
<tr><td>lastName</td><td>Doe</td></tr>
<tr><td>org</td><td>Institute for Software Systems</td></tr>
<tr><td>eMail</td><td>greensoft@umwelt-campus.de</td></tr>
<tr><td>originator</td><td>yes</td></tr>
</table></td></tr>
<tr><td>Keyword</td><td>Network Traffic, Browser, Advertisement Blocker, Cache</td></tr>
<tr><td>Reference</td><td>King, A., 2008. Website Optimization. 1st ed. Sebastopol: O’Reilly Media</td></tr>
</table>

9. Conclusion and Outlook

Summing up, the development of software in a resource-saving way is an important field of research and helps to reduce the energy consumption and the environmental impact. The knowledge base of guidelines, tips, and hints is a good opportunity to use and to expand the potentials of Green IT.
Up to now, we are at the end of the conception and development phase of the model. We detailed and broadened our model with usage scenarios, use cases and a structured overview model. Our next steps are to create screen flows for the user interface and to start a prototypical implementation. The subsequent user acceptance test will follow. For the future, we plan to evaluate the platform to find out, which articles are particularly efficient.

10. Acknowledgement

Our paper evolved from the research and development project “Green Software Engineering” (GREENSOFT) sponsored by the German Federal Ministry of Education and Research under reference number 17N1209.

11. Literature

Apple (2010), Internet: The story behind Apple’s environmental footprint, http://www.apple.com/environment/complete-lifecycle/#facilities, last access 28.06.2010.

Abenius, S. (2009): Green IT & Green Software – Time and Energy Savings Using Existing Tools. In: Wohlgemut, V., Page, B.,Voigt, K. (eds.): EnviroInfo 2009 : Environmental Informatics and Industrial Environmental Protection: Concepts, Methods and Tools, 23rd International Conference on Informatics for Environmental Protection - Volume 1: Conference Sessions. Herzogenrath : Shaker. Pages 57 – 66.

Amsel, N., Tomlinson, B.: Green Tracker: a tool for estimating the energy consumption of software. In ACM (ed.), CHI EA '10: Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems. New York: ACM. Pages 3337–3342.

Arndt, H.-K., Lau, S., Strehl, A. (2009): Sustainability of Information and Communication Systems (ICS). In: Wohlgemut, V., Page, B.,Voigt, K. (eds.): EnviroInfo 2009 : Environmental Informatics and Industrial Environmental Protection: Concepts, Methods and Tools, 23rd International Conference on Informatics for Environmental Protection - Volume 1: Conference Sessions. Herzogenrath : Shaker. Pages 67 – 74,

Dick, M., Naumann, S., Held, A. (2010): Green Web Engineering. A Set of Principles to Support the Development and Operation of “Green” Websites and their Utilization during a Website’s Life Cycle. Filipe, Joaquim; Cordeiro, José (eds.): *WEBIST 2010*: Proceedings of the 6th International Conference on Web Information Systems and Technologies, April 7 - 10, 2010, Valencia, Spain, Volume 1. Setúbal: INSTICC Press, pages 48 - 55.

Filetti, M., Mulisch, H.-M., Gnauck, A. (2008): Environmental Data Online – Design of a Web Based Environmental Information System for Public Services. In: Möller, A., Page, B., Schreiber, M. (eds.): Environmental informatics and industrial ecology : proceedings of the 22nd International Conference Environmental Informatics - Informatics for Environmental Protection, Sustainable Development and Risk Management, September 10 - 12, 2008, Leuphana University Lueneburg, Germany / EnviroInfo 2008, 22nd International Conference on Informatics for Environmental Protection. Aachen : Shaker. Pages 390 – 395.

Fujitsu (2009), Internet: ProGREEN IT Mission, http://www.progreenit-mission.de/, last access 20.06.2010.

Hilty, L. M. (2008): Information Technology and Sustainability – Essays on the Relationship between Information Technology and Sustainable Development, Books on Demand GmbH, Norderstedt, page 18.

Leach, P., Mealling, M., Salz, R. (2005), Internet: A Universally Unique IDentifier (UUID) URN Namespace, RFC 4122, The Internet Society, July 2005, http://tools.ietf.org/html/rfc4122, last access 25.06.2010

Microsoft Developer Network, Internet: http://msdn.microsoft.com/, last access 25.04.2010.

Naumann, S. (2008): Sustainable Informatics – A new Subfield of Applied Informatics? In: Möller, A., Page, B., Schreiber, M. (eds.): Environmental informatics and industrial ecology : proceedings of the 22nd International Conference Environmental Informatics - Informatics for Environmental Protection, Sustainable Development and Risk Management, September 10 - 12, 2008, Leuphana University Lueneburg, Germany / EnviroInfo 2008, 22nd International Conference on Informatics for Environmental Protection. Aachen : Shaker. Pages 384-389.

Posdziech, Michael (2009), Internet: Concert Stick: Das Konzert zum Mitnehmen, http://www.onlinekosten.de/news/artikel/34270/0/Concert-Stick-Das-Konzert-zum-Mitnehmen, last access 17.06.2010.

Schreiber, M. (2009): EnviroInfoLit – A Literature Database for Environmental Informatics. In: Wohlgemut, V., Page, B.,Voigt, K. (eds.): EnviroInfo 2009 : Environmental Informatics and Industrial Environmental Protection: Concepts, Methods and Tools, 23rd International Conference on Informatics for Environmental Protection - Volume 2: Conference Workshops and ICT-ENSURE Tracks. Herzogenrath : Shaker. Pages 201 – 204.

Softwareload, Internet: http://www.softwareload.de/, last access 20.06.2010.

Umweltbundesamt Berlin (ed.), Tischner, U., Dietz, B. (2000): How to Do Ecodesign?: A Guide for Environmentally and Economically Sound Design, Verlag form GmbH, Frankfurt am Main.

Umweltbundesamt (ed.), Behrendt, S., Erdmann, L. (2009): Computer, Internet und Co. - Geld sparen und Klima schützen, http://www.umweltdaten.de/publikationen/fpdf-l/3725.pdf, last access 28.04.2010.

Yahoo! Developer Network (ed.), Internet: Best Practices for Speeding Up Your Web site, http://developer.yahoo.com/performance/rules.html#postload, last access 28.04.2010.