

Migrating to Modern Web Technologies - Experiences in Re-Designing a Substance Retrieval Systems

Matthias Menger¹, Petra Jahnke¹, Josef Eiblmaier²

Abstract

The successfully running search & retrieval applications for the substance database GSBL have been coming to age and need to reflect more the new needs of users dealing with modern networks and infrastructures. In its first version, the substance database GSBL used to be a classical desktop application. Performant access (the combination of throughput, response time, availability, and functionalities) was restricted to a narrow group of users having a desktop (fat) client installed. This allows a very performing client application with a wide range of functionalities which make this search & retrieval system unique in its class of substance information systems. Search and retrieval of substructures in more than 200.000 chemicals within a fraction of a second has been just one essential feature.

With the growing of wider and better networks, more users and the approach to use software as a service, the obstacles of a pure client application become obvious. Innovative Web technologies and the availability of ever better performing networks allow to provide information to broad public more easily, faster and more convenient. Functionalities for search & retrieval do not need a fat client including installation and administration burdens. Just using your browser to get the same qualified information from everywhere you have access to the network was the aim of the re-design.

Therefore, the cutting-edge technologies like Apache STRUTS, Ajax and JSTL were applied to create an easy to handle and efficient web application named 'GSBL Web application 3.0'. The following paper will describe the choice of technologies which fulfil best the needs of current and potential GSBL-users. This re-design allows continuing the successfully running service for environmental relevant information for chemical substances. The combination of continuously updated, comprehensive and reliable environmental data of the Federal Environment Agency (Umweltbundesamt) with InfoChem's expertise in substance and structure handling make this application is an indispensable resource for public environmental authorities, first action forces, and governmental departments in the legislative process.

1. History and Background of GSBL

The GSBL is a joint substances data pool of the Federal State and its Länder and has been established in 1994. This central database contains important, useful and reliable information on chemical substances, their impact on health and environment, and their relation to current legislative acts.

From public administrations over first actions forces a wide range of Institutions are using the data to prevent or react on incidents or handling of hazardous chemical substances. The central data pool is managed at the Federal Environment Agency 'Umweltbundesamt' with its headquarter in Dessau. But having data and information available without a retrieval and presentation layer, which allows the potential user to browse and search these information in a convenient (state-of-the art) way, does not make any sense. Therefore, a complex software suite with a range of tools for administration, search and retrieval has been developed over the last decade.

¹ Federal Environment Agency, Dep IV2.1 Information Systems Chemical Safety, Wörlitzer Platz 1, D-06844 Dessau
e-mail: petra.jahnke@uba.de, matthias.menger@uba.de, Internet: <http://www.umweltbundesamt.de>

² InfoChem GmbH, Landsberger Str. 408, D-81241 München,
e-mail: je@infochem.de, Internet: <http://www.infochem.de>

2. The 'old' GSBL-Applications

2.1 Experiences

The discovery and presentation of the data and information is achieved by using a modular built information system which offers, besides complex registration and administration tools, a very performing search & retrieval software. Two applications had been developed over the years to fulfil client's needs of those days. The result has been one pure client application with a huge package of functionalities (all based on most important user requirements!) and later on a thin web client application with less functionality. Both client applications have been using the same server software to perform a very fast search and retrieval of data.

The pure or 'fat' client application with an installation routine on each user PC offers many and complex functionalities to discover the best information out of the data pool.

In contrast, the web client has been limited to a range of 'most used' functionalities and discards professional search functions which have been rarely or just by a few users either used. Nevertheless, important features like searching for chemical sub-structures or multiple database fields have been seen as essential and have been realised. Development technologies and software have been based on the lowest common denominator of the first user groups recruiting from the public sector and the known restrictions of those days. The result has been a CGI- based Web application programmed in C/C++.

2.2 Shortcomings

Time alone has been contributed to shortcomings of the applications. Current information technologies offer much simpler possibilities to provide the same (or even more) functionalities in a search and retrieval system. The server side C++-modules and also the CGI technology has become obsolete meanwhile. Moreover not being multithreaded has become a significant obstruction for performance and usability. But this was not the only reason to redesign the application. In combination with the increase of users and thus the necessity to handle parallel sessions on the search & retrieval server leads to a non acceptable loss of performance and stableness of the whole application. Security and administration issues had to be addressed in a more and more inconvenient way. Customers using the pure client application were in need of frequent updates, although the application meanwhile has been dispersed over hundreds of PCs – not regarding the redundant installed server application in different institutions all over Germany. This burden of administration has consumed a lot of time and efforts to keep 'everybody' up-to-date not even knowing if everybody has got the latest version.

Another important reason for redesign has been the far to complex user guidance and the realisation of the GUI (graphical user interface) with plain html which has been lead to restrictions and unacceptable 'clicking' session to get the desired result.

3. New Requirements - User Needs for a Modern GSBL-Application

Unfortunately user requirements and user's environment is changing more and more frequently. The first web client has already been a reaction to the changing requirements, infrastructures, and the awareness that not all requirements of the first days have been so 'important' as they seemed to be. With these current IT infrastructure and commodities user behaviour and their requirements has changed in a way that it was no longer possible to offer a performing tool to discover the useful information in a way users are used to. The technologies used for the first approach have been coming to their limits regarding performance, multiple users and process handling, user guidance, stableness, and security.

The availability and accessibility of wider and better networks, e.g. bandwidth, performance, in most institutions (even in the public sector) leads to more and more 'software as a service' understanding and

thus concepts of providing more and more pure web-based software. The new approach should provide an easy access from anywhere, just using an internet browser connected to the network (internet or intranet) with a clear user guidance and easy to understand search features. The need to install an additional 'fat' client to make more useful(!) functionalities available should be avoided and so reduce administration efforts, i.e. version handling, and costs.

With modern software technologies have becoming standards in nowadays solutions and infrastructures, security policies of IT departments are already reflecting possible concerns without reducing features offered by these modern technologies.

The new approach of GSBL allows user (group) specific settings defined in different 'look and feel' profiles. This should lead to an increased product acceptance and operator convenience. Moreover, it offers a good extensibility for future enhancements, which are sure to come.

4. Technologies used for the Redesign of GSBL-Web 3.0

Taking into account the above mentioned, implementing a web based application was the consequent decision. Moreover, state of the art, performant and maintainable technologies had to be applied, which was the reason to choose an web container that implements the Servlet and Java Server Pages (JSP) specifications. Due to financial restrictions the employment of the freely available Servlet engine *Apache Tomcat* was decided. Besides the support of integrated management of user sessions the usage of Java technology (JavaBeans, JSP, Expression Language, Tag Libraries, JSTL), web application frameworks like Apache STRUTS and the employment of modern software architectures/paradigms (MVC - Model View - Controller) is provided by this software package.

To accelerate the development process and to apply a standardized request- and response handling, it was decided to employ the web application framework STRUTS.

With using STRUTS, a huge set of comprehensive Java libraries is already available and many requirements on the server side software are already met by the framework.

Since the application needs frequent screen updates (search forms, hit list and dossier display, tree view, hit list navigation), a method reducing the server load and speeding up the user interface had to be found. The development concept *AJAX* was chosen to meet this requirements. Because it is cross-browser/ cross-platform compatible and uses open standards, *AJAX* has proven to be the optimal means to lower the server load and to increase screen display speed.

Core server side modules of the application (*RetrievalServer* and *AccessServer*) have proven effective and performant so that these parts could be reused in the new application. The communication protocol between the web application and this components is TCP/IP because it is a standardized protocol, platform independent and it allows to distribute components on separate platforms. XML as a customisable, flexible format is used as language for data exchange not least because many software libraries - especially in Java - are already available for creation, reading, handling and processing this format. Furthermore data contained in XML can be mapped well to a Java object model.

One requirement was the server side generation of hit list and dossier reports in RichText- and MS Excel format. The use of free available server side Java libraries to generate RichText and Excel documents suggested itself. For the generation of RichText documents the Java library *iText* is used. MS Excel spreadsheets are generated using the Library *JExcelAPI*. The advantages are obvious: These libraries are free available, well documented, stable and platform independent.

The screenshot shows the GSBL Webapplication 3.0 interface. The browser window title is "GSBL Webapplikation 3.0 - IE 6 (Microsoft Internet Explorer)". The address bar contains the URL: `http://gsbl.infochem.de/gsblweb30/login.do;jsessionid=4E51A59CCA86159AFBC6B73E35C994DE`. The page header features the GSBL logo and the text "Umwelt Bundes Amt GSBL". Below the header, there are navigation options like "Abmelden", "Hilfe", and "Impressum". The main content area is titled "GSBL062THES Datenbank > Liste1 > p-anisidine". A search bar and navigation links are visible. The left sidebar lists various categories such as "IDENTMERKMALE", "RECHTSEIGENSCHAFTEN", "STOFFEIGENSCHAFTEN: UMGANG / VERHALTEN", etc. The main panel displays "ALLE_MERKMALE" for "p-anisidine". It includes a "Struktur" section with a chemical structure diagram of p-anisidine (COc1ccc(N)cc1). Below the structure, there is a table titled "Registriername (Großbuchstaben) (1 - 11 von 11)".

	GSBL-RN	Merkmalsnummer	Registriername (Großbuchstaben)	GSBL-RN
1 von 11	17	2966	P-ANISIDINE	
2 von 11	17	2967	P-ANISIDIN	

Figure 1: Screen Shot of GSBL 3.0
Source: Federal Environment Agency

5. First Experiences with GSBL-Web 3.0

The first test results show that the efforts of re-designing the GSBL Web application has paid-off in terms of performance and usability. The GSBL-Web 3.0 is currently under testing but already at this stage it has been achieved to establish a central application server for (at least) the cooperation partner to gain from the synergies of a web-based application. This and the fact that it is a thin-client application running via the network will minimise the burden of administration. Every user group may provide its specific profile to fit best their needs without lacking important functionalities as it used to be with the previous version.

It seems that the purpose of the GSBL pure client application is becoming obsolete and it is going to be faded out in the near future.

Bibliography

Eiblmaier, Dr. Josef, InfoChem GmbH (2005): Analyse des UBA GSBL-WebClients und Vorschlag fuer eine Neu-Implementierung, (unpublished)

Jahnke, Petra, (Umweltbundesamt), Ruhdorfer, Dr. Jakob, Eiblmaier, Dr. Josef (InfoChem GmbH), (2007): Software Design Description UBA-GSBL Webapplikation 3.0, Vers.2.2, (unpublished)