D3:
Developing an online workspace platform
for asynchronous collaboration

Željko Klindžic
Žarko Vuckovic
Jasmin Klindžic

Abstract

The WWW (W3) has enormous possibilities, but for realization of effective collaboration solutions certain future innovations of W3 are needed. Implementation of additional software to W3 server is one possible way to extend its existing capabilities. This paper describes experimental work that used standard W3 server, HTTP protocol, HTML and CGI modules to develop a simple platform for asynchronous online collaboration. Easy to reconfigure, modular platform simply named "Workspace" enables communication, document exchange and online tool sharing among of group of users, all inside virtual rooms of shared services. Content of a particular room, displayed to user, depends on access scheme defined on service basis. Set of standard rooms (communication, resources, tools and administration room) can be reconfigured, new rooms and services can be added and user rights changed by editing access and deny lists inside each room. Services are modules for specific tasks as administration, shared resource management, internal communication, publishing tools or just information displays. WS was tested at portals SB OnLine, Pozega-online, Osijek-online and several other web projects. Experience gained by WS usage shows that even simple software solutions can be used before future W3 innovation.

Introduction

In its current form, the World Wide Web (W3) provides a simple and effective means for users to search, browse and retrieve information, as well as to make information of their own available for others. The utility of this approach, whereby providers design and retain control over the form and content of their information, is evident from the growth of the W3 to become the primary means of accessing information over the Internet.

This approach, however, does not provide support for more collaborative information sharing, where widely-dispersed working-groups can jointly author, comment and annotate documents, negotiating and developing the form and content of the information as part of a group, rather than an individual, activity. (1)

The work on a project, where active collaboration between a group of associates, often dislocated in space and time, is necessary, requires collaboration system that will be realized as private communication space with shared information, resources and online tools. Based on

potential of Internet and W3, number of online collaboration systems is developed that we can classify in four categories, depending on the extent to which they depart from existing W3 standards (1):

1. Purely W3-based: uses standard W3 clients, comply with HTML and HTTP standards, and only extends server functionality using the CGI interface.

2. Customized servers: require special-purpose servers, to provide behavior beyond the possibilities offered by CGI. Such systems support standard W3 clients and protocols, but the enhancements may reduce the portability.

3. Customized clients: requires particular or modified clients, or non-standard client

4. Web-related: Such systems may provide a W3 interface, but support only limited interaction using the HTTP protocol.

This paper analyses a development of our know-how and presents our purely W3-based system for asynchronous collaboration using CGI program modules on a standard W3 server.

Problem identification

During several years of online community web-portal development and work on different web projects, through which we have tried to enable multi-user collaboration in publishing and managing of online resources and services, we have encountered a problem of different software applications and clients installed on our collaborator's local PCs (accompanied by different levels of computer literacy and skills of our collaborators), multi-user active access to the server content and similar. The stable and efficient collaboration was hard to establish, because although all points were interconnected by Internet, the problems emerged in almost every physical point of collaboration.

Conclusion: Communication infrastructure and incompatible points of collaboration do exist, but organized collaboration space is still unavailable, hence the results are weak or non-existent. (Fig.1. Case 1)

"You want to give users and groups the ability to work together in teams, and give them tools that are simple and easy to use. You need to make it easy, simple to use, straightforward." says David Hobbes. (2)

To achieve better results, we have installed online tools on web servers - program modules for work simplification and reduction of need for local, platform-specific clients and local software. For example, HTTP upload modules replaced FTP clients, and server-side modules (web forms for database entry; web content generation with database and template usage) replaced local web page design tools. We have developed program modules for services like W3 board, classified ads, address-book and several automatic publishing systems, all of that can be administered online. After the replacement of some classic services and local clients and tools with W3 server extensions, the only tool considered necessary by collaborators was standard, unmodified W3 browser, and according to their reactions, the collaboration was simplified. Volume and frequency of document exchange and publication have also been increased and new collaborators have joined much faster than before.

Conclusion: The points of collaboration became more simple and compatible. With server-side software implementation, the virtual collaboration space is formed, and the results are getting better. (Fig.1. Case 2)

With the increased number of users and more and more different, mutually independent, online tools on servers, together with the increase in number of servers equipped with similar modules, administration of user groups, maintenance and development of additional software modules became more and more complicated. Because of the need for user authorization on different addresses to access different online server-side tools and the differences in module interfaces have rendered collaboration more and more uncomfortable.

Conclusion: Efficiency is going down because the virtual collaboration space became fragmented and without a clear layout. (Fig.1. Case 2)

Case-study analysis has brought us to the conclusion that there should be a server-side platform, some kind of basic shell that will simplify implementation of online collaboration tools. Basic shell has to solve access and security scheme and enable connection of functional modules – online services (tools) into organized entity. That basic shell has to simplify administration of the relations between space of users and space of online services, and the standardization of the

each service interface, together with the clear layout of the virtual collaboration space should render the operations on the user's side much easier than they were before.

Analysis of these problems and available solutions (1) has led us to the universal platform model that realizes the collaboration space with modular organization consisting of subspaces of shared services. Service can be realized with particular program module i.e. Perl CGI or PHP module or similar, communicating with the user through W3 server and dynamically generated web interface. In server-side operations these modules are using data files and MySQL or some other online database. After previously created online collaboration tools were redesigned and standardized they were added to the basic shell as services and prototype named "Workspace" was born.

After "Workspace" was implement on variety of servers, a number of positive effects were noticed. Together with the simplification in administration and more stable work, now we had more time for further development of specific services. Most obvious benefits for the collaborators were, beside the standardization and simplification of each tools' interface, a clear and logical layout of the collaboration space organized as 'rooms' of shared services.

Conclusion: Organized virtual collaboration space realized through W3 server-side collaboration system creates a stable and powerful base for online collaboration. (Fig.1. Case 3)
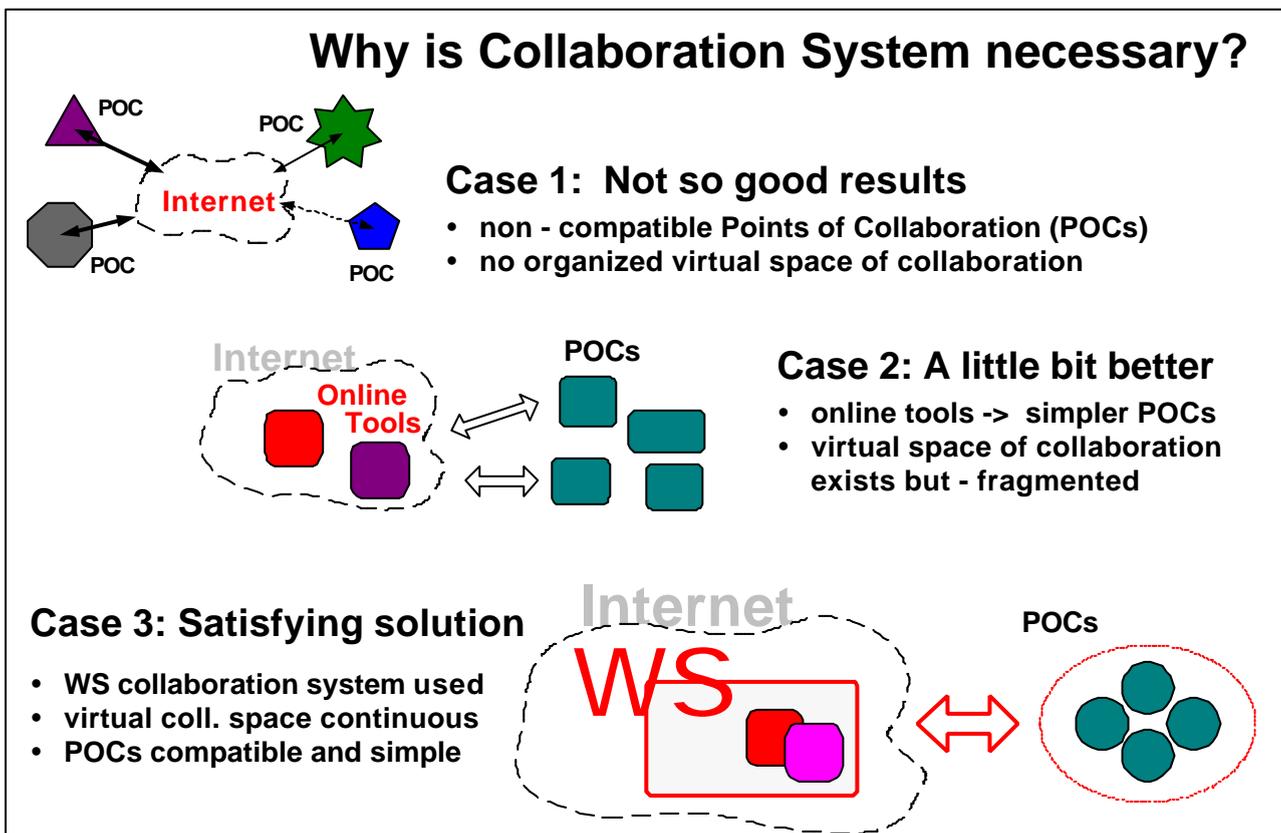
Why is Collaboration System necessary?

Case 1: Not so good results
- **non - compatible Points of Collaboration (POCs)**
- **no organized virtual space of collaboration**

Case 2: A little bit better
- **online tools -> simpler POCs**
- **virtual space of collaboration exists but - fragmented**

Case 3: Satisfying solution
- **WS collaboration system used**
- **virtual coll. space continuous**
- **POCs compatible and simple**

Fig.1. Illustration of problem identification phases

Realization of a universal basic collaboration shell

Universal virtual collaboration shell, simply named **Workspace** was realized through Perl modules on CGI-enabled W3 server using text files for configuration data storage. Dynamically generated web-based user interface uses only standard-compliant HTML and occasionally - JS (JavaScript) code. This so-called purely W3 based solution (1), is applicable on any standard CGI-enabled W3 servers, and standard, unmodified W3 browser is all that is needed at the user's side. The universal shell can further be improved by adding program modules used for task specific services (i.e. by adding specific online collaboration tools) to produce complete collaboration system.

Virtual collaboration space is organized as set of topic-related sub-spaces – rooms of services. Configuration data of the virtual workspace, subspace content, user information and subspace service access list are stored in configuration tables, thus enabling the simple configuration of the virtual space and administration of the relation(s) among user set and service set. Workspace wasn't constructed as a tailorable system (5), because the primary goal was the simplicity of construction. User interface is very simple, enabling user to travel through set of subspaces, see list of available services and activate particular service.

Each subspace has an independent list of services, access and deny list, and in them, users are treated either as members of certain groups (administrators, registered users, members of topic-related groups and unidentified users – guests) or individually. (Some discussion about authorization and access control in CSCW can be seen in ref. 3)

Subspaces can be added like modules by adding items in the configuration table of the main navigation menu, which is followed by the preparation of the introductory text, service table and definition of the users access right. The creation of each new subspace also involves adding a new section to the online operating manual. These operations can be done by users having level of WS administrator, but also by others if the administrator allows them to do so by placing their ID in the access table of administrative tools.

In addition to the informative and passive sections (display-only type service), the most important sections in the subspace content are active services (online tools). While the first type is used for

data and information presentation only, and do not change the server content, the second type is used for change or creation of the server content (writing of messages on W3 board, news creation, database update, upload of documents to archives, workspace administration or preparation and publication of internally or publicly visible web documents).

Adding services to the Workspace shell

A new service is added to a platform as a program module executable on server upon call from WS-shell by simply stating its URL and passing set of parameters. It can be written as a CGI Perl module, PHP script or similar, by obeying communication rules (syntax) of WS shell calls.

Called module should be able to accept entry parameters (the key parameter being **the ticket** – user session identifier) from the WS shell. Module must be able to find and read tables with currently active (and all registered) users for ticket validity check and additional user information retrieval. When the work is finished, module must return the user to the WS while presenting his/her ticket, which means that during the whole procedure it has to keep both the user's ticket and record of the URL needed for return to the WS shell. In case of procedure error or invalid ticket detection, module must enable automatic return to WS shell.

Online tool can have its own configuration and administrative functions available to the WS master administrator. A good illustration of that is a tool for collaboration in preparing and publishing news: WS administrator in service access table defines a group of collaborators that will be able to start this service and use that tool, which gives them the lowest status - "author". Inside the online tool section, administrator defines who is the editor-in-chief. Later on, editor-in-chief can use internal administration functions to "promote" some of the authors into "editor" status. Authors can only prepare their own articles by editing text and adding pictures in the content database (without the "publish" function available, their articles are invisible to the general public), while editors can work with all articles and publish the content made by authors (or use some internal administrative and configuration functions).

Usage: A voyage through the *Workspace*

If a user wants to enter Workspace, first he/she has to authorize himself/herself (Login operation) at the authorization page (ticket office, session marker generator, cookie generator). After the initial identification, the user is given a randomly generated data sequence (**ticket**, cookie, session marker) that also has a time duration and relation to the user's computer (IP address) added. Ticket and other user data are then stored in the currently active users table. Later on, when moving around the WS, the user is identified by his/her ticket.

Inside the *Workspace*, the user can move through rooms of services by using the main navigation menu, and at the end, he/she can leave WS by using Logout operation from the main menu. The

ticket (session marker) is destroyed and entry in the currently active (online) users table is erased.

Standard subspaces of the Workspace are **Lobby** or **Welcome room** with *Workspace* news & help desk, **Coffee & post office** - communication room with internal communication services, **Library** – room to with services to use shared resources **Factory** – online tools room with services for active collaboration in resource management, publishing etc, and finally **WS Office** - administration room with WS configuration and management services.

Each subspace shows the user a dynamically generated web document with main navigation menu, introductory text, link to the manual and service table (links for activating and short description of the service). Inside a subspace, the user can see and use only those services that he/she has been registered for in access list, and not registered in the deny list (negative rights principle used for fine tuning of access scheme, ref. 3). During his/her work it possible to get a context-sensitive help, and helpdesk feature is realized as an e-mail support. With "Hello, who is online?" function, which uses the currently active (online) user list, it is possible to see who is currently inside the WS and perhaps invite him/her to a synchronous communication (JavaChat) in the communication room.

Fig.2. Illustration of user's interface – at the top is the main menu, below is a content of Online Tools Room

Some standard *Workspace* services

Main WS administrative services have tools acting at Workspace or room level. At *Workspace* level is possible to edit main parameters, table of rooms, table of users and analyze logs. At room level, admin tool enables editing of introductory text, help text, table of services, access and deny lists. Additionally, admin can use service for writing information to be displayed in welcome room as *Workspace* News.

Inside communication room standard services are W3 board (discussion), members directory (contact data, important addresses etc), internal person-to-person messaging system (internal message exchange without a need for e-mail). JavaChat room can be added as a kind of synchronous communication service.

Shared resources room, or Library, typically has services to use archive of documents with catalogue organized by topic. User can upload and download files of different type using W3 browser only. Other typical service is WebDirectory used as an internal archive of global resources addresses in which WS user can add links.

Tools room has services for managing other services and shared resources (for moderators, team leaders, editors etc) and online services with production tools for information resources building, production of web publications in remote collaboration and similar. Service can be used to manage or publish inside Workspace only or something visible in outside world. For example, we can join team of authors inside WS and give them access to online service for preparing and publishing daily news at web portal. Some of them can act as writers, authors, or editors, while all of them access online tools trough Workspace services for online collaboration in publicly visible content production.

Some management service examples are W3 Board or WebDirectory admin. With these services authorized user can moderate discussion, approve new or edit old entries in Web directory. Examples of publishing services are Instant News Manager (single-column news) and Column Master (publishing system useful for FAQ mastering, multi-columns news etc..).

Live testing: The examples of *Worspace* usage

Putokazi ravnice (Web portals & online communities) SB OnLine, [www.sbonline.net], Pozega-Online, [www.wolf.hr], Osijek-Online, [www.osijek-online.com].

> The collaboration of editorial teams in web-portal content realization. Publication of web content through collaboration of the community members, construction and administration of the public information resources by number of entities.

CTR d.o.o, Local development agency, Slavonski Brod, [www.ctr.hr]

> The collaboration of the development team, administration of the online services for County Brodsko-Posavska entrepreneurs and small business

CDL, Intes.Hr, Slavonski Brod, www.intes.hr/cdl

> Experimental collaboration space for teachers and students

Matija Antun Relkovic High School, Slavonski Brod, skola-mar.avalon.hr
White Stork Online Archive

Online collaboration of three European schools (Spain, Romania, Croatia) in "Europe At School 2001" competition. Communication, document archives, publishing of multilingual EkoMAR zine on weekly basis in remote collaboration.

e-Zbornica

Experimental project of high-school teachers collaboration. Communication, document archive for education material exchange, and the development of the self-learning resources.

Conclusion

Together with the growing number of Internet users, the average level of understanding the increasingly complex software applications and tools is getting lower, which results in disproportionably fast growth of the number of users online, as opposed to the rather slow and gradual development of the efficient practical appliance of the Internet. If we want to enhance the efficiency of the Internet, with special focus on the development of online collaboration, W3 as a biggest Internet potential has to be changed and upgraded in capabilities, and at the same time making the necessary simplification changes on user's side. One of the available possibilities, before W3 improvement, is the development of software for W3 server capabilities extension, and simple solutions, like described in this paper, are following that guideline.

References:

(1.) Richard Bentley, Thilo Horstmann, Klaas Sikkel, Jonathan Trevor:
Supporting Collaborative Information Sharing with the World Wide Web: The BSCW Shared Workspace System
http://bscw.gmd.de/Papers/WWW4-Boston/WWW4-Boston.html

(2.) Joe Paone:
Collaboration: Bringing Everyone Together, Microtimes, Issue 221, May 14, 2001.
http://www.microtimes.com/221/cvstypaone221a.html

(3.) Klaas Sikkel:
A Group-based Authorization Model for Cooperative Systems,
in: Proceedings European Conference on Computer-Supported Cooperative Work (ECSCW'97),
Lancaster, September 1997, © Kluwer, Dordrecht.

(4.) Thilo Horstmann and Richard Bentley:
Distributed Authoring on the Web with the BSCW Shared Workspace system
in: ACM Standards View, Spring 1997, ACM Press.
http://bscw.gmd.de/Papers/ACM-StandardsView/Author.html

(5.) Appelt, W., Hinrichs, E., Woetzel, G.:
Effectiveness and Efficiency: The Need for Tailorable User Interfaces on the Web, in Proceedings of the 7th International World Wide Web Conference, Brisbane, April 1998.
http://bscw.gmd.de/Papers/WWW7-Brisbane/index.html

(6.) **http://www.w3.org/Collaboration/Workshop/**
Workshop on WWW and Collaboration, September 11-12, 1995, Cambridge, MA