Hello AMMRL'ers,

My apologize for the delay in getting this summary out. I have been busy learning and working on this issue, and can now discuss what I am doing more concretely than possible even a short time ago.

First, let me state once again what a great group this is! A number of our members are willing to offer assistance on this issue, some by providing complete solutions, some offering limited tailoring, some offering source code, etc.

There are many ways to implement calendaring systems via the web. Eleven solutions, all different, are listed below! The last describes how I am proceeding; I have surprised myself by choosing to not follow in the footsteps of others. A combination of local training and programming assistance, and our facility's complex requirements for sign-ups lead me astray.

All responses to my initial email are summarized below. Please contact the individual directly for questions about their respective systems. I provide some detail about how I am proceeding at the end.

> At 02:34 PM 7/30/2003 -0500, you wrote:
> Hi all:
> >
> > We are finally looking into getting web-based scheduling implemented in our facility. It appears that we need more flexibility than FACES provides. We need to implement a well-defined set of sign-up rules that, as one example, will allow students the flexibility of deleting other users' sign-ups (a student can sign-up "illegally:" another student can remove that sign-up, but only if they will use the time; this rule does increase usage on the spectrometers). Another example is adding the capability of charging a fee--smaller than the actual usage fee--for sign-ups that are removed too close to the use time.
> >
> > We are looking into the Calcium program as one possibility for doing what we need.
> >
> > Summaries of scheduling software that I can find on the AMMRL archives are now four years old or older.
> >
> > If you are using some other kind of software for scheduling, and it is available for use to others, I would appreciate hearing back from you. I will provide a summary of the responses.
> >
> > Cheers,
> > Charlie

**1** Jeff Simpson <jeff.simpson@unh.edu> has written a FORTRAN-based calendaring system called res1. It is currently in use at UMass/Amherst, MIT, and UNH. Jeff is interested in refining res1 so it may better serve others. It runs stand-alone on Sun, SGI, or RedHat platforms. res1 is not a point-and-click system, but that keeps the bandwidth extremely small by today's standards.

**2** Dave Scott <scott@iastate.edu> has home-grown perl scripts comprising a system viewable at: http://olf.chem.iastate.edu/NMR/pages/schedules/frame.html. Dave is willing to provide source code (which he describes as pre-beta :). This is a very tailorable, straight-forward system that runs under IRIX and Linux.
**3** Pierre Audet <Pierre.Audet@chm.ulaval.ca> has written tcl/tk scripts running on an Apache server to manage their reservation list. The code generates its own database, but currently does not provide accounting features. The user calendar tools are simple to use, and nicely laid out. Pierre likely is willing to share his code with others interested in doing their own modifications.

**4** Ronald F. Federspiel <federspi@chuma1.cas.usf.edu> has a system based on MS Outlook. Students can view instrument schedules via this system, and make new reservations (appointments). Ronald says the system has worked well for them.

**5** Shaoxiong Wu <swu@emory.edu> is using a system based on a Win2000 server, Cold Fusion, and an Oracle database. This is a very configurable, "modern" system. The first page can be viewed at http://www.emory.edu/NMR/. Shaoxiong hired a programmer to write their system.

**6** Bruce Lix <bruce.lix@ualberta.ca> has a system based on php, an open-source technology similar to what I've decided to use (see 11 below). Schedules can be viewed at http://www.nanuc.ca/nmr/schedules.php, but because of NANUC's charter, time has to be requested via a separate form. The calendars and time requests are therefore not dynamically linked.

**7** NMRFAM also has complete scheduling (Sundial) and time request (Sand) modules in their Sesame Project. These javascript-based modules are very configurable. Contact Zsolt Zolnai <zsolt@nmrfam.wisc.edu> for more information.

**8** The FACES system from CCRC is a system available for general use for scheduling, and has an advantage that CCRC provides the server (and thus, security) for the system. See http://www.ccrc.uga.edu/web/facilities/facilityframe.html for more information.

**9** The Calcium software (see http://www.brownbearsw.com/) is being used in various places, including Northwestern. Contact Yuyang Wu <wu@chem.northwestern.edu> for questions about their implementation.

**10** Chris Rithner <cdr@colostate.edu> informed me about a system available from John Hazelrigg <hazelrigg@uh.edu>.

```
* Program:  nmr_resv.c  -  UH Chem. Dept. version 1.2
*       resv_01.c  -  Distribution copy of nmr_resv.c
```

**11** Last, and certainly at this time, least, is the system I am developing. I decided, following discussions with many of the kind folks listed above (thanks in particular to Jeff, Dave, Pierre and Chris in really going out their way to assist), sifting through their source code, playing with FACES and Calcium, etc., and thinking hard about the rather weird and more-complex-than-common sign-up requirements in our lab, that I needed to write my own calendaring system. A class I took two weeks ago here at the UW in Active Server Pages (ASP) got me jump-started, and the instructor has been invaluable as a resource.

ASP uses any of a number of programming languages, but most commonly Visual Basic, written within and around HTML code to make dynamic web pages. ASP links to MS Access or Oracle databases via SQL. I've found that ASP is simple to learn, and am making good progress in writing an application that is quite general, and has excellent flexibility. Security features appear to be strong. ASP is very common on all brands of web servers, easing server issues. I will use our departmental server (no changes needed, and no problems with its administrator), thus avoiding having to install and administer a server myself. In addition, it is simple to setup a desktop PC as a test-bed, allowing development/trouble-shooting outside the primary server.

The primary negative with ASP is that it is Microsoft's way of doing dynamic data-driven web pages. PHP is an open source alternative that I would have preferred if our computer department had provided training and support for it rather, or in addition to, ASP. ColdFusion is a solid, Macromedia-provided, alternative to ASP and PHP.

When I finish, I will update this group again as to how it all _really_ went: seems easy so far, but I'm probably only ~10% of the way to having a complete application. Continued discussion from any of you on this topic is likely to be welcomed by AMMRL as interest is clearly high within the community.

Cheers,
Charlie