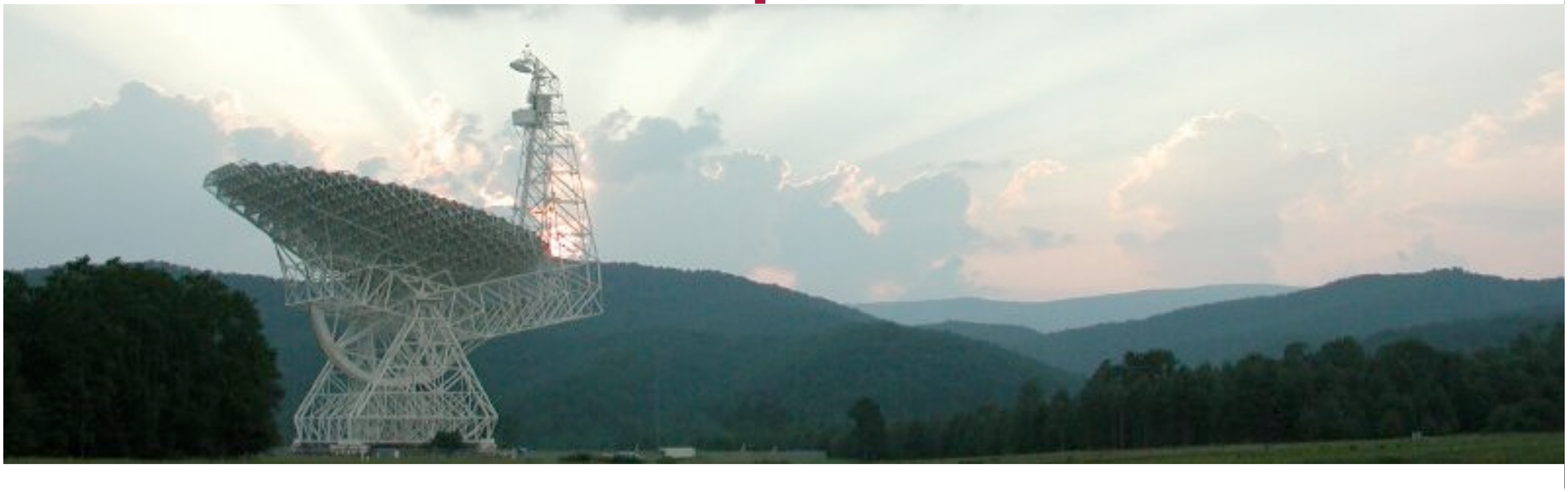
A Case Study in Web 2.0 Application Development



Paul R. Marganian

Software Engineer – National Radio Astronomy Observatory, Green Bank

Atacama Large Millimeter/submillimeter Array
Expanded Very Large Array
Robert C. Byrd Green Bank Telescope
Very Long Baseline Array

NRAO

Authors:

Paul Marganian

Mark Clark

Amy Shelton

Mike McCarty

Eric Sessoms

Atacama Large Millimeter/submillimeter Array
Expanded Very Large Array
Robert C. Byrd Green Bank Telescope
Very Long Baseline Array

NRAO

Introduction

- Discussing our recent experiences creating a Web 2.0 application: the new Dynamic Scheduling System (DSS) for the Green Bank Telescope (GBT).
- Addresses ADASS focus topic: Commonly available development environments & tools
- What is Web 2.0?
  - Wikipedia: "is commonly associated with web development and web design that facilitates interactive information sharing, interoperability, user-centered design and collaboration on the World Wide Web."
  - Desktop quality applications for the Web Browser.

Overview

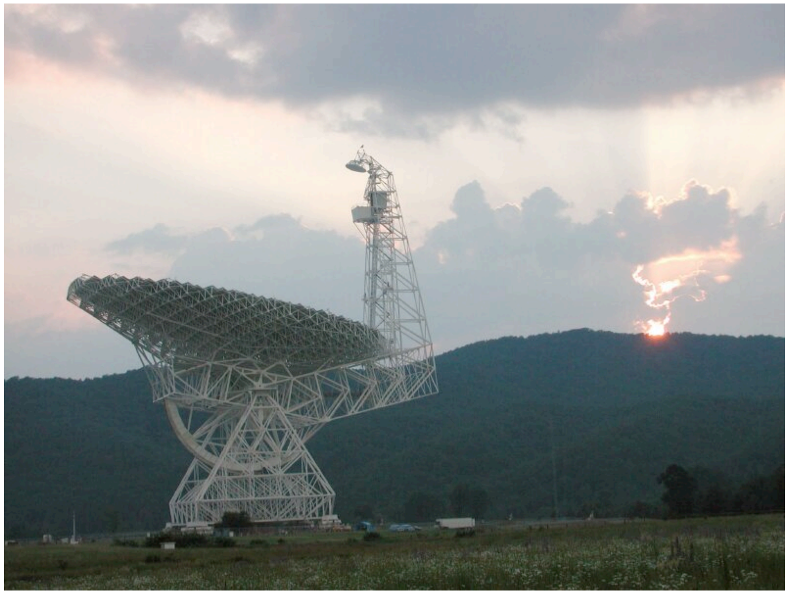
Background

Beta Test

Release

Vegetable ONLY
BIER GARTEN
BIER GARTEN
HOKKAIDO

Background: The GBT



- The Robert C. Byrd Green Bank Telescope is the world's largest fully steerable radio telescope.
- The Green Bank Telescope (GBT) is located at the National Radio Astronomy Observatory's site in Green Bank, WV.
- 100 m diameter
- Can observe over a larger range of frequencies than other comparable centimeter/millimeter single-dish telescopes.
- Is located in a continental, mid-latitude region where weather is dominated by water vapor and small scale effects.

Background – The Problem: Dynamic Scheduling

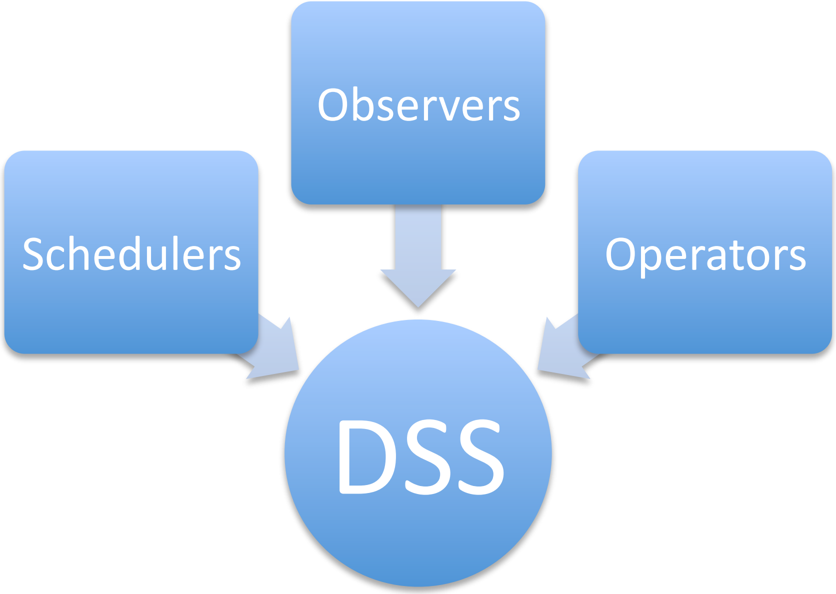
- The primary goal of the DSS is to improve the efficiency of GBT observations by matching the observing schedule to current weather conditions, while allowing each observer to retain interactive control of the telescope
- Example: use the GBT for high-frequency science for clear, calm weather, and schedule low-frequency science when the weather is otherwise.
- Exceptional Scheduling Paradigm: not scheduling scripts, but ***scheduling observers.***

Background – Problem: Dynamic Scheduling

- How it works:
  - Each day the DSS will examine the weather forecast, equipment availability, observer availability, and other factors, and set an observing schedule for the 24-hour period beginning the next day.
  - Observers will therefore get 24-48 hours notice before their project will observe.
  - Additionally, observers will have the opportunity to pause their observing program, set blackout dates indicating when they are unavailable for observing and back out of current observations if they find the observing conditions are not suitable to their science goals.
- ADASS talk on this last year, by site directory, Dr. Karen O'Neil; along with several posters.

Backround – The Problem: Dynamic Scheduling

- Since we're scheduling observers, the observers have to have a high level of interaction with the system.
- Hundreds of geographically dispersed users.



Background – The Solution

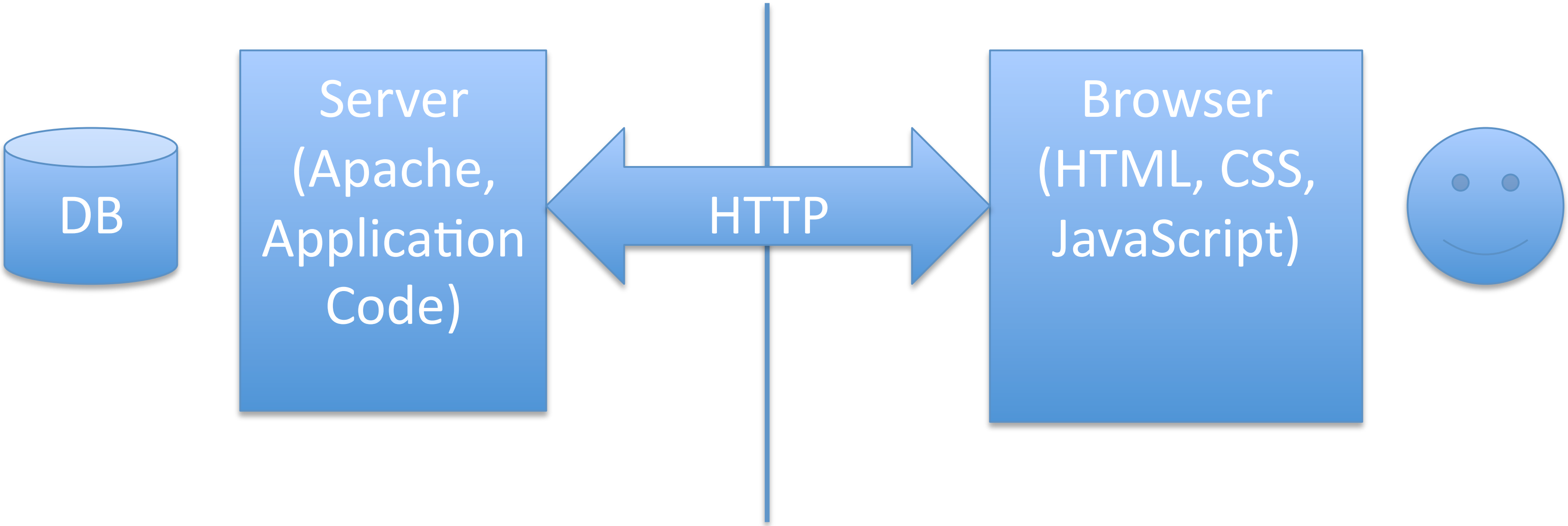
- What is required to dynamically schedule the GBT is not just complex scheduling algorithms (though we certainly need that)
- We also require complex interactions with all the users.
- ***Solution:*** A Web Application!
- And not just a traditional series of web pages, but a Web 2.0 Application.

Background – The Green Bank Software Development Division

- We have to implement this solution.
- Who's 'we'? ~2.5 FTEs over the project life-time.
- Until quite recently, we have focused on three main areas of activity, utilizing three languages:
  - Monitor & Control – C/C++
  - Observation Management – Python
  - Data Analysis – IDL
- What type of skill-set and knowledge-base are required to make a web application?

Background – Web Applications, Grossly Simplified

Concepts: *Database, Server, HTTP, Authentication, HTML, CSS, JavaScript, DOM, AJAX, XML, JSON*


DB
Server (Apache, Application Code)
HTTP
Browser (HTML, CSS, JavaScript)

Overview

Background

Beta Test

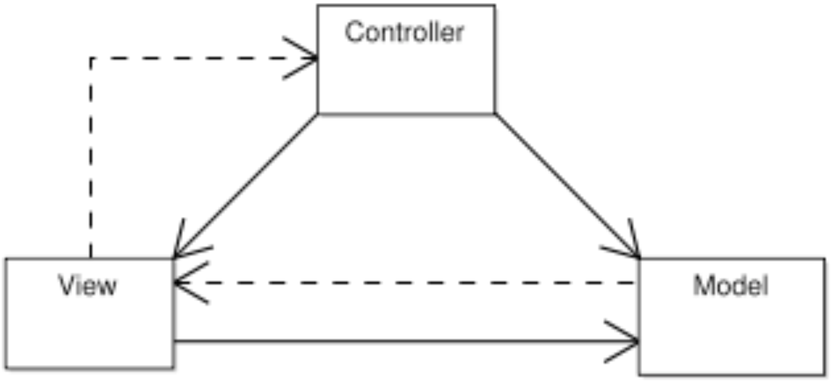
Release

Frameworks

- As discussed previously, many diverse components involved in a web application
- A framework pulls a lot of these components together for you with a number of benefits:
  - Manages many crucial activities by default
  - Enables a web application to get up off the ground quickly
  - Allows you to work in the language of your choice
- Many frameworks available, several in the language of our choice, Python
- For the Beta Test (2008), the choice came down to two: TurboGears & Django

TurboGears – a Python web framework

- All in python.
- TurboGears follows the MVC design pattern: isolates business logic from input and presentation.
  - Model: the domain-specific representation of the data on which the application operates.
  - View: Renders the model into a form suitable for interaction
  - Controller: Receives input and initiates a response by making calls on model objects.



Model – Object Relational Mapper (ORM)

- Wikipedia: "in computer software is a programming technique for converting data between incompatible type systems in relational databases and object-oriented programming languages"
- Can be a good compromise if data persists in a DB, but you'd like to work with objects.
- We used SqlAlchemy as our ORM
- Worked very well for our purposes:
  - DSS DB is fairly complex, but mapped to objects fine
  - Performance not an issue; DSS DB size is moderate

Controller - CherryPy

- Wraps the HTTP Protocol
- CherryPy maps URLs from the browser to python methods on the server
- Each method can then query the model, then render a template (a page refresh to the user), or redirect to another URL.
- Example: url = /addValues?a=1&b=2
- Calls method addValues, with keyword-value pairs a = 1, b = 2

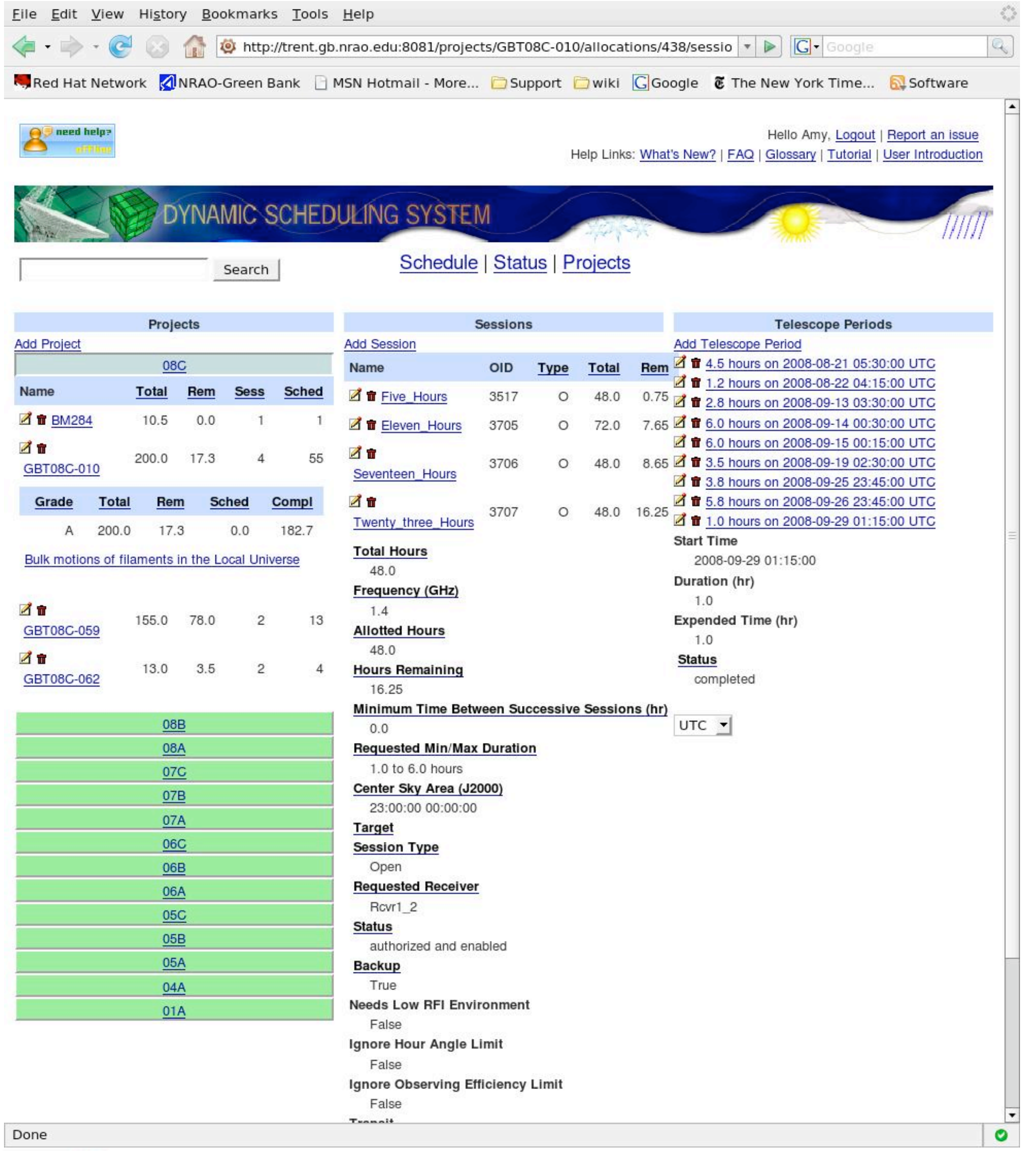
View – Genshi Templates

- A templating language that allows you to embed python in html.
- Excellent way of separating your display style from your content.
- Example: building a table dynamically.

TurboGears – other features

- With all the previously mentioned tools, you can get a functioning web application up and running quickly.
- More tools for making working with forms (validation, etc.) very easy.
- But apart from a few special widgets (e.g. date picker), web pages using just these components are mostly form based (very Web 1.0).
- TurboGears allows you to enhance your web pages with CSS and JavaScript, and supports AJAX.
- This is the basic approach we took to the Beta Test's Observers' Interface

DSS Beta Test Observers Interface


File Edit View History Bookmarks Tools Help
http://trent.gb.nrao.edu:8081/projects/GBT08C-010/allocations/438/sessio
Google
Red Hat Network
NRAO-Green Bank
MSN Hotmail - More...
Support
wiki
Google
The New York Time...
Software
need help? offline
DYNAMIC SCHEDULING SYSTEM
Search
Done

Hello Amy, Logout | Report an issue

Help Links: What's New? | FAQ | Glossary | Tutorial | User Introduction

Schedule | Status | Projects

Projects

Add Project

08C

Name	Total	Rem	Sess	Sched
BM284	10.5	0.0	1	1
GBT08C-010	200.0	17.3	4	55

Grade	Total	Rem	Sched	Compl
A	200.0	17.3	0.0	182.7

Bulk motions of filaments in the Local Universe

Name	Total	Rem	Sess	Sched
GBT08C-059	155.0	78.0	2	13
GBT08C-062	13.0	3.5	2	4

08B
08A
07C
07B
07A
06C
06B
06A
05C
05B
05A
04A
01A

Sessions

Add Session

Name	OID	Type	Total	Rem
Five_Hours	3517	O	48.0	0.75
Eleven_Hours	3705	O	72.0	7.65
Seventeen_Hours	3706	O	48.0	8.65
Twenty_three_Hours	3707	O	48.0	16.25

Total Hours
48.0

Frequency (GHz)
1.4

Allotted Hours
48.0

Hours Remaining
16.25

Minimum Time Between Successive Sessions (hr)
0.0

Requested Min/Max Duration
1.0 to 6.0 hours

Center Sky Area (J2000)
23:00:00 00:00:00

Target

Session Type
Open

Requested Receiver
Rcvr1_2

Status
authorized and enabled

Backup
True

Needs Low RFI Environment
False

Ignore Hour Angle Limit
False

Ignore Observing Efficiency Limit
False

Transit

Telescope Periods

Add Telescope Period

4.5 hours on 2008-08-21 05:30:00 UTC
1.2 hours on 2008-08-22 04:15:00 UTC
2.8 hours on 2008-09-13 03:30:00 UTC
6.0 hours on 2008-09-14 00:30:00 UTC
6.0 hours on 2008-09-15 00:15:00 UTC
3.5 hours on 2008-09-19 02:30:00 UTC
3.8 hours on 2008-09-25 23:45:00 UTC
5.8 hours on 2008-09-26 23:45:00 UTC
1.0 hours on 2008-09-29 01:15:00 UTC

Start Time
2008-09-29 01:15:00

Duration (hr)
1.0

Expended Time (hr)
1.0

Status
completed

UTC

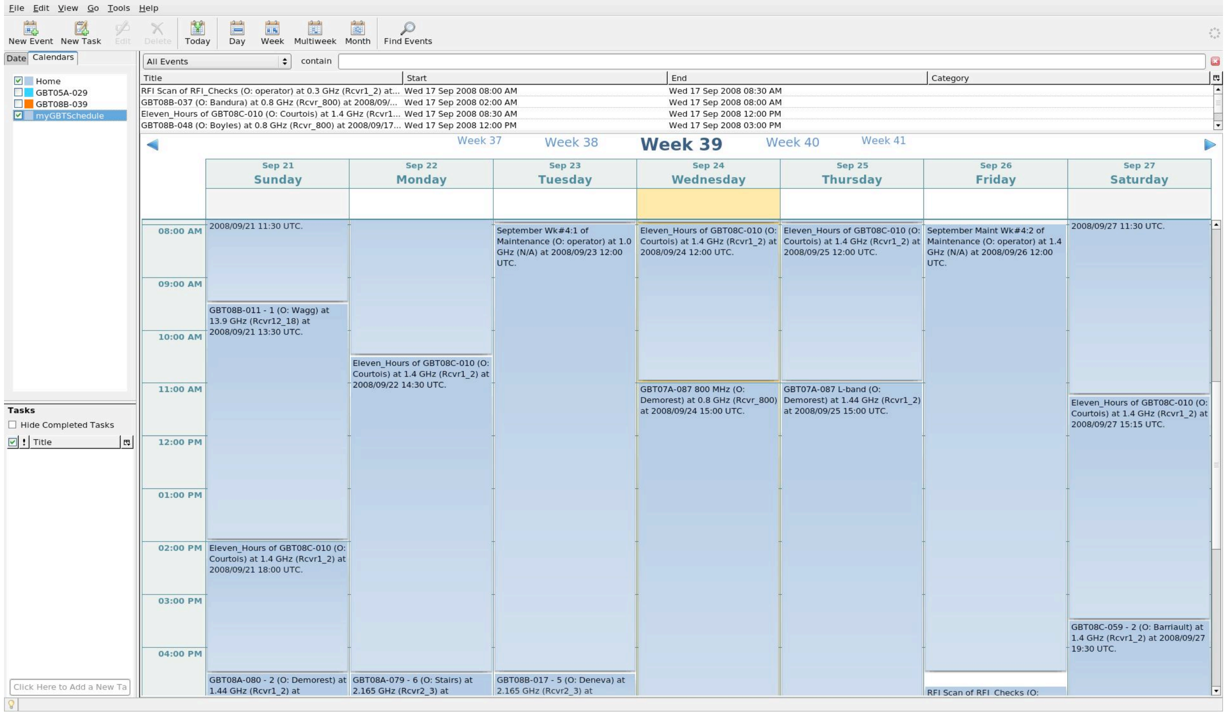
Created with:

- TurboGears
- CSS package: Blueprint
- a sprinkle of JavaScript.

DSS Beta Test Observers Interface

ICAL

- There some other technologies that bear mentioning in this interface:
- iCalendar format – wikipedia: "a computer file format which allows internet users to send meeting requests and tasks to other internet users, via email, or sharing files with an .ics extension"



☑ Home
☐ GBT05A-029
☐ GBT08B-039
☑ myGBTSchedule

☐ Hide Completed Tasks

Title	Start	End	Category
RFI Scan of RFI_Checks (O: operator) at 0.3 GHz (Rcvr1_2) at...	Wed 17 Sep 2008 08:00 AM	Wed 17 Sep 2008 08:30 AM	
GBT08B-037 (O: Bandura) at 0.8 GHz (Rcvr_800) at 2008/09/...	Wed 17 Sep 2008 02:00 AM	Wed 17 Sep 2008 08:00 AM	
Eleven_Hours of GBT08C-010 (O: Courtois) at 1.4 GHz (Rcvr1...	Wed 17 Sep 2008 08:30 AM	Wed 17 Sep 2008 12:00 PM	
GBT08B-048 (O: Boyles) at 0.8 GHz (Rcvr_800) at 2008/09/17...	Wed 17 Sep 2008 12:00 PM	Wed 17 Sep 2008 03:00 PM	

08:00 AM, 09:00 AM, 10:00 AM, 11:00 AM, 12:00 PM, 01:00 PM, 02:00 PM, 03:00 PM, 04:00 PM

Sep 21 Sunday
2008/09/21 11:30 UTC.
GBT08B-011 - 1 (O: Wagg) at 13.9 GHz (Rcvr12_18) at 2008/09/21 13:30 UTC.
Eleven_Hours of GBT08C-010 (O: Courtois) at 1.4 GHz (Rcvr1_2) at 2008/09/21 18:00 UTC.
GBT08A-080 - 2 (O: Demorest) at 1.44 GHz (Rcvr1_2) at

Sep 22 Monday
Eleven_Hours of GBT08C-010 (O: Courtois) at 1.4 GHz (Rcvr1_2) at 2008/09/22 14:30 UTC.
GBT08A-079 - 6 (O: Stairs) at 2.165 GHz (Rcvr2_3) at

Sep 23 Tuesday
September Wk#4:1 of Maintenance (O: operator) at 1.0 GHz (N/A) at 2008/09/23 12:00 UTC.
GBT08B-017 - 5 (O: Deneva) at 2.165 GHz (Rcvr2_3) at

Sep 24 Wednesday
Eleven_Hours of GBT08C-010 (O: Courtois) at 1.4 GHz (Rcvr1_2) at 2008/09/24 12:00 UTC.
GBT07A-087 800 MHz (O: Demorest) at 0.8 GHz (Rcvr_800) at 2008/09/24 15:00 UTC.

Sep 25 Thursday
Eleven_Hours of GBT08C-010 (O: Courtois) at 1.4 GHz (Rcvr1_2) at 2008/09/25 12:00 UTC.
GBT07A-087 L-band (O: Demorest) at 1.44 GHz (Rcvr1_2) at 2008/09/25 15:00 UTC.

Sep 26 Friday
September Maint Wk#4:2 of Maintenance (O: operator) at 1.4 GHz (N/A) at 2008/09/26 12:00 UTC.
RFI Scan of RFI_Checks (O:

Sep 27 Saturday
2008/09/27 11:30 UTC.
Eleven_Hours of GBT08C-010 (O: Courtois) at 1.4 GHz (Rcvr1_2) at 2008/09/27 15:15 UTC.
GBT08C-059 - 2 (O: Barriault) at 1.4 GHz (Rcvr1_2) at 2008/09/27 19:30 UTC.

DSS Beta Test Observers Interface

- Other tools used in the observers interface:
  - RSS – wikipedia : “Rich Site Summary” is a family of web feed formats used to publish frequently updated works—such as blog entries, news headlines, audio, and video—in a standardized format.”

    ask live! chat online
  - Chat software – we incorporated third party chat software so that observers can get live help from GBT staff.
- Although a more sophisticated interface for observers was desired, these simpler web pages worked well.
- However, for the scheduler’s interface, some of the requirements for the interface required us to enhance what the browser was doing.

DSS Beta Test Scheduler Interface

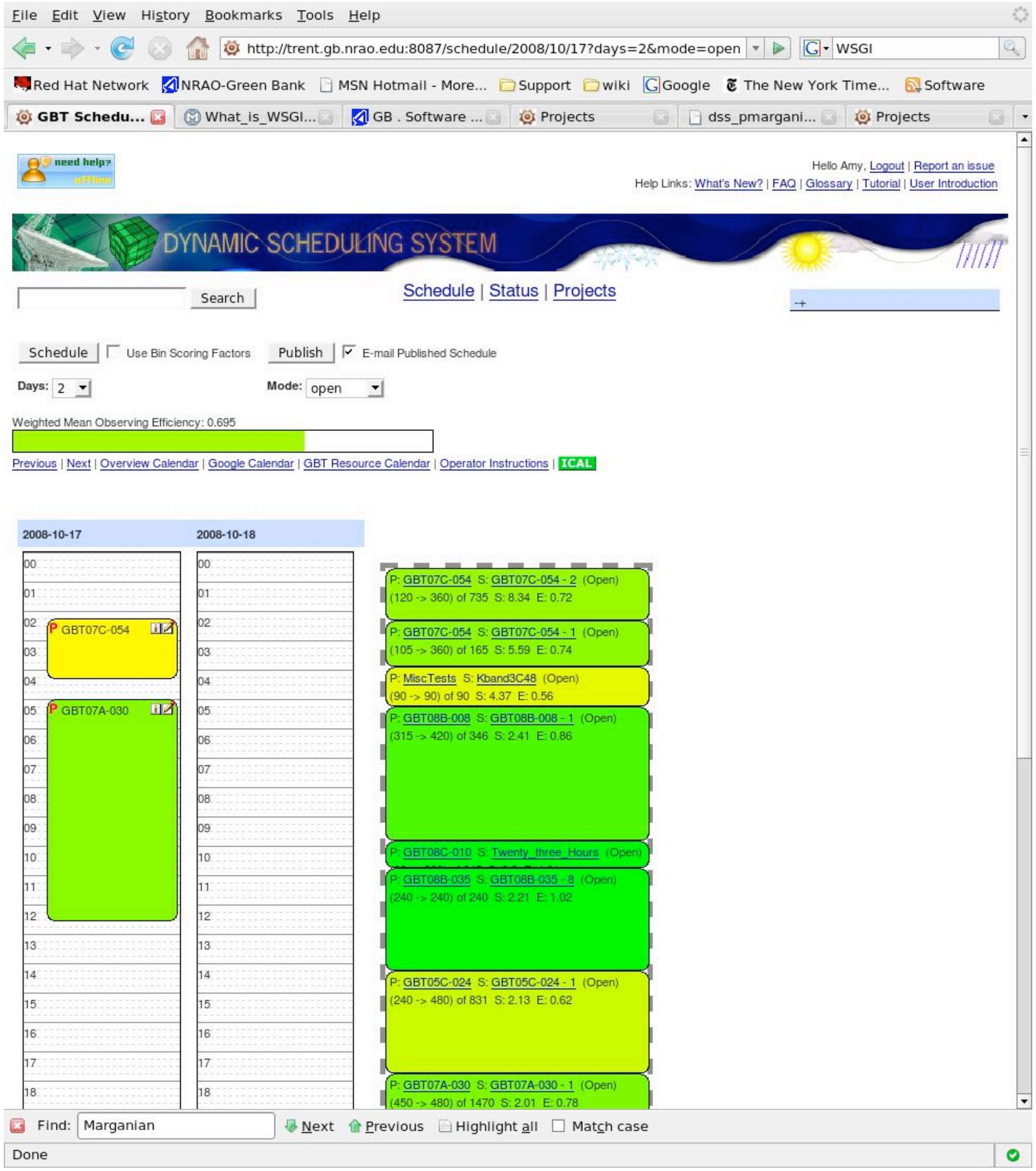
- Basically, we had to provide a graphical calendar that supported the graphical manipulation of observing periods on the telescope, with all the feedback necessary so that the scheduler can maximize the observing efficiency of the GBT.
- So, we had to move to JavaScript., and make use of AJAX. We took advantage of a JavaScript library, JQuery.
- JQuery provides:
  - DOM element selections
  - DOM traversal and modification
  - Events
  - CSS manipulation
  - Effects and animation

DSS Beta Test Schedulers Interface

Features:

- Drag and drop calendar
- Feedback
- Color coded efficiencies
- Easy access to project information

Only supported Firefox 2.0


File Edit View History Bookmarks Tools Help
http://trent.gb.nrao.edu:8087/schedule/2008/10/17?days=2&mode=open
WSGI
Red Hat Network
NRAO-Green Bank
MSN Hotmail - More...
Support
wiki
Google
The New York Time...
Software
GBT Schedu...
What_is_WSGI...
GB . Software ...
Projects
dss_pmargani...
Projects
need help? offline
DYNAMIC SCHEDULING SYSTEM
Search
Schedule | Status | Projects
Schedule
Publish
Previous | Next | Overview Calendar | Google Calendar | GBT Resource Calendar | Operator Instructions | ICAL
Find:
Next
Previous
Highlight all
Match case
Done

Hello Amy, Logout | Report an issue

Help Links: What's New? | FAQ | Glossary | Tutorial | User Introduction

□ Use Bin Scoring Factors ☑ E-mail Published Schedule

Days: 2 Mode: open

Weighted Mean Observing Efficiency: 0.695

2008-10-17 2008-10-18

00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18

P GBT07C-054

P GBT07A-030

P: GBT07C-054 S: GBT07C-054 - 2 (Open)
(120 -> 360) of 735 S: 8.34 E: 0.72

P: GBT07C-054 S: GBT07C-054 - 1 (Open)
(105 -> 360) of 165 S: 5.59 E: 0.74

P: MiscTests S: Kband3C48 (Open)
(90 -> 90) of 90 S: 4.37 E: 0.56

P: GBT08B-008 S: GBT08B-008 - 1 (Open)
(315 -> 420) of 346 S: 2.41 E: 0.86

P: GBT08C-010 S: Twenty_three_Hours (Open)

P: GBT08B-035 S: GBT08B-035 - 8 (Open)
(240 -> 240) of 240 S: 2.21 E: 1.02

P: GBT05C-024 S: GBT05C-024 - 1 (Open)
(240 -> 480) of 831 S: 2.13 E: 0.62

P: GBT07A-030 S: GBT07A-030 - 1 (Open)
(450 -> 480) of 1470 S: 2.01 E: 0.78

Find: Margarian

DSS Beta Science Algorithms

- Without these there'd be no need for the Web Application.
- Implemented in Python.
- Interfacing them with the web application was trivial: simply imported and called from the CherryPy controller methods.
- Example:
  - Scheduler hits 'Schedule' in the interace,
  - which gets mapped to a 'schedule' method on the server
  - Calls 'schedule' algorithm in the science python modules

Beta Test: Lessons Learned

- Feedback from users from helpdesk and survey:
  - Observers interface was actually as sophisticated enough for users.
  - Scheduler's interface was *not* sophisticated enough.
- Experience of developers:
  - TurboGears framework worked well for us.
  - Working with JavaScript was pain-full.
  - Science Algorithm code much to slow, and error prone.
- Used lessons learned to prepare for the production release (Oct. 1, 2009)

Overview

Background

Beta Test

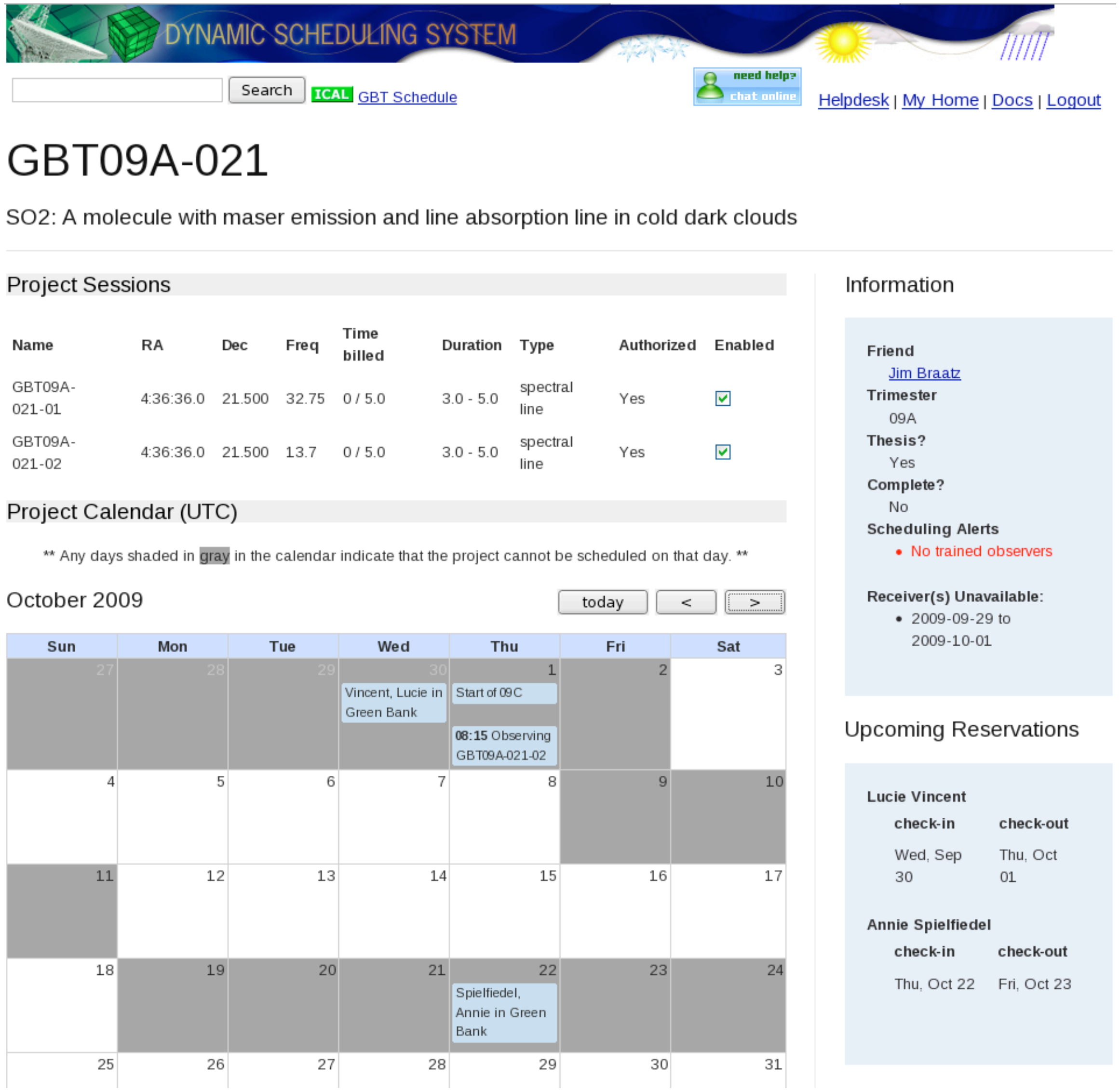
Release

Release: Django Framework

- TurboGears served us well in the Beta Test, but in the mean time, the Django framework really took off.
- Django framework is extremely similar to TurboGears. Switching over was very simple.
- Instead of brining together different 3rd party libraries like TurboGears, Django uses it's own ORM, templates, etc.
- On the whole offers more features then TurboGears.
- Larger user base meant more 'freebies'. Some of these turned out to save us a great deal of time.

Release: Observers Interface

- We were able to quickly develop an observer's interface using mostly features from Django.
- Project Calendar using the FullCalendar plugin for JQuery.



GBT09A-021

SO2: A molecule with maser emission and line absorption line in cold dark clouds

Project Sessions

Name	RA	Dec	Freq	Time billed	Duration	Type	Authorized	Enabled
GBT09A-021-01	4:36:36.0	21.500	32.75	0 / 5.0	3.0 - 5.0	spectral line	Yes	☑
GBT09A-021-02	4:36:36.0	21.500	13.7	0 / 5.0	3.0 - 5.0	spectral line	Yes	☑

Project Calendar (UTC)

** Any days shaded in gray in the calendar indicate that the project cannot be scheduled on that day. **

October 2009

Sun	Mon	Tue	Wed	Thu	Fri	Sat
27	28	29	30 Vincent, Lucie in Green Bank	1 Start of 09C **08:15** Observing GBT09A-021-02	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22 Spielfiedel, Annie in Green Bank	23	24
25	26	27	28	29	30	31

Information

Friend
Jim Braatz
Trimester
09A
Thesis?
Yes
Complete?
No
Scheduling Alerts
- No trained observers

Receiver(s) Unavailable:
- 2009-09-29 to 2009-10-01

Upcoming Reservations

Lucie Vincent

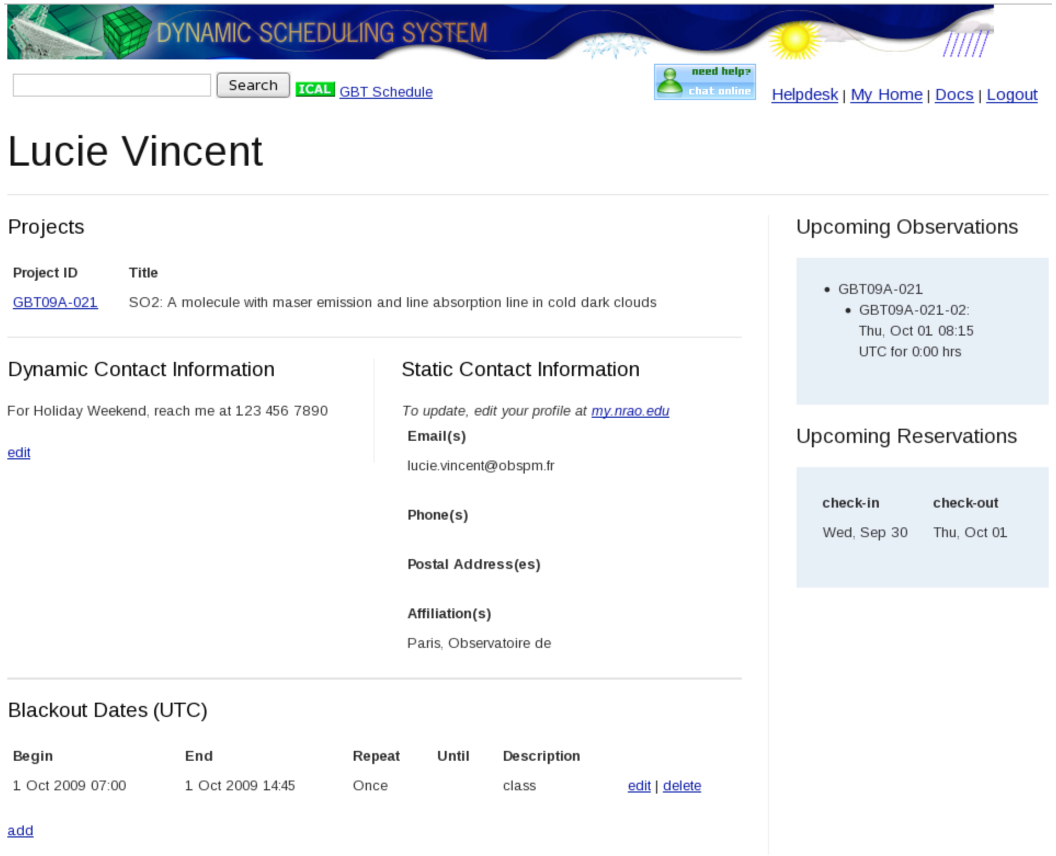
check-in	check-out
Wed, Sep 30	Thu, Oct 01

Annie Spielfiedel

check-in	check-out
Thu, Oct 22	Fri, Oct 23

Release: integrating with other NRAO systems

- share a single database of user contact info (avoiding the Beta Test nightmare)
- other NRAO systems provide URL query services that serve up XML encoded data.
- Ex: user contact information



Lucie Vincent

Projects

Project ID	Title
GBT09A-021	SO2: A molecule with maser emission and line absorption line in cold dark clouds

Dynamic Contact Information

For Holiday Weekend, reach me at 123 456 7890

edit

Static Contact Information

To update, edit your profile at my.nrao.edu

Email(s)

lucie.vincent@obspm.fr

Phone(s)

Postal Address(es)

Affiliation(s)

Paris, Observatoire de

Blackout Dates (UTC)

Begin	End	Repeat	Until	Description	
1 Oct 2009 07:00	1 Oct 2009 14:45	Once		class	edit | delete

add

Upcoming Observations

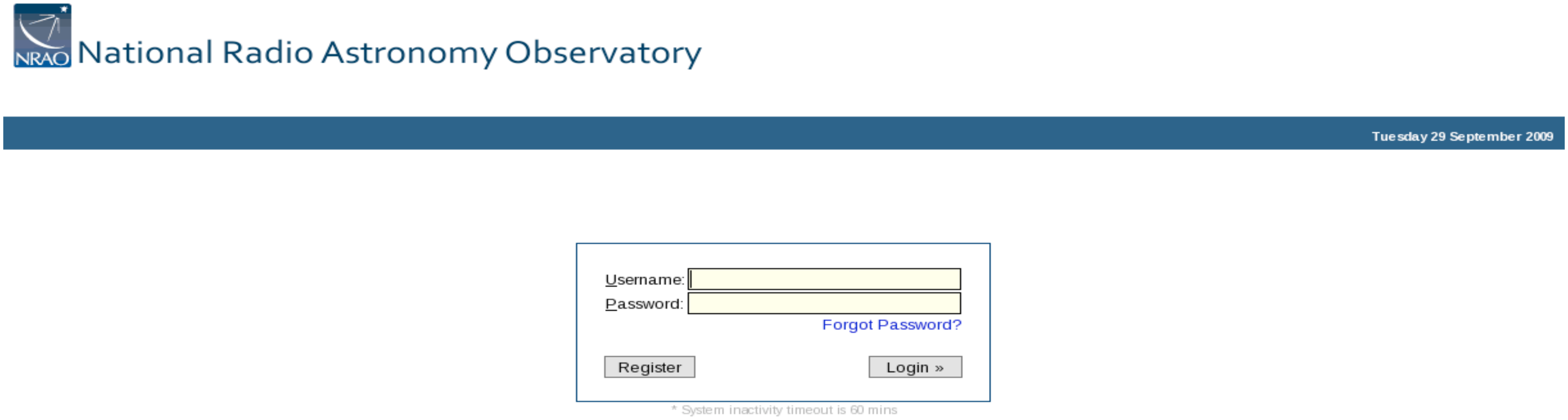
- GBT09A-021
  - GBT09A-021-02: Thu, Oct 01 08:15 UTC for 0:00 hrs

Upcoming Reservations

check-in	check-out
Wed, Sep 30	Thu, Oct 01

Release: Authentication with CAS

- NRAO is using an open-source single-sign-on service: Central Authentication Service (CAS)
- This way, users can log in to one NRAO service (ex: Proposal Submission Tool), and seamlessly move to another serve (ex: DSS) without logging in again.
- Django allows you to specify the authentication backend you want to use.



National Radio Astronomy Observatory

Tuesday 29 September 2009

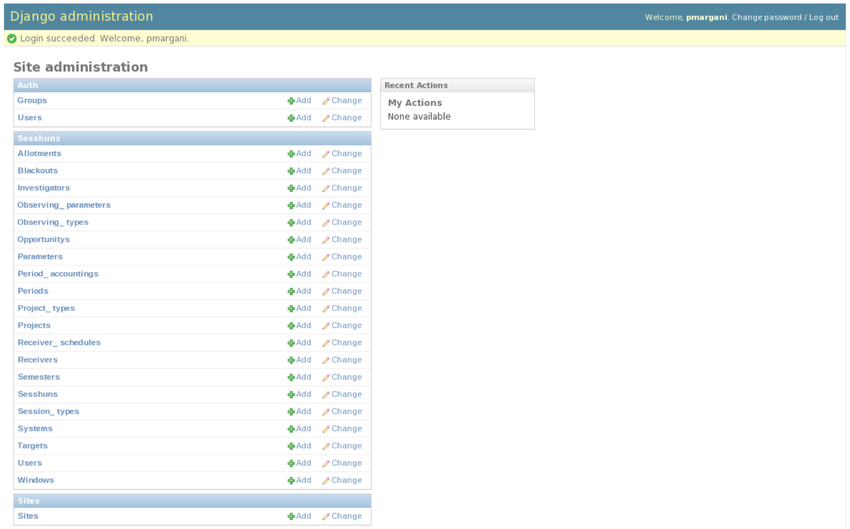
Username:

Password:

* System inactivity timeout is 60 mins

Release: Django Admin Tool

- Django comes with a built in Admin Tool, which is basically a sophisticated view into your database (like phpMyAdmin).
- Is extensible to handle custom actions.
- Result: functionality that may not be covered in your own interface is often covered by the Admin Tool.



Django administration

Welcome, **pmargani**. Change password / Log out

Login succeeded. Welcome, pmargani.

Site administration

Auth		
Groups	Add	Change
Users	Add	Change

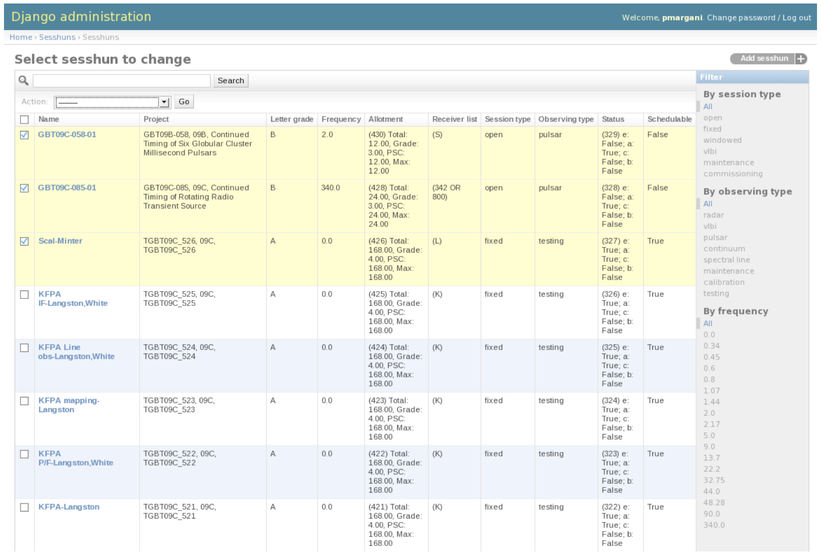
Sesshuns		
Allotments	Add	Change
Blackouts	Add	Change
Investigators	Add	Change
Observing_ parameters	Add	Change
Observing_ types	Add	Change
Opportunitys	Add	Change
Parameters	Add	Change
Period_ accountings	Add	Change
Periods	Add	Change
Project_ types	Add	Change
Projects	Add	Change
Receiver_ schedules	Add	Change
Receivers	Add	Change
Semesters	Add	Change
Sesshuns	Add	Change
Session_ types	Add	Change
Systems	Add	Change
Targets	Add	Change
Users	Add	Change
Windows	Add	Change

Sites		
Sites	Add	Change

Recent Actions

My Actions

None available



Django administration

Welcome, **pmargani**. Change password / Log out

Home › Sesshuns › Sesshuns

Select sesshun to change

□	Name	Project	Letter grade	Frequency	Allotment	Receiver list	Session type	Observing type	Status	Schedulable
☑	GBT09C-058-01	GBT09B-058, 09B, Continued Timing of Six Globular Cluster Millisecond Pulsars	B	2.0	(430) Total: 12.00, Grade: 3.00, PSC: 12.00, Max: 12.00	(S)	open	pulsar	(329) e: False; a: True; c: False; b: False	False
☑	GBT09C-085-01	GBT09C-085, 09C, Continued Timing of Rotating Radio Transient Source	B	340.0	(428) Total: 24.00, Grade: 3.00, PSC: 24.00, Max: 24.00	(342 OR 800)	open	pulsar	(328) e: False; a: True; c: False; b: False	False
☑	Scal-Minter	TGBT09C_526, 09C, TGBT09C_526	A	0.0	(426) Total: 168.00, Grade: 4.00, PSC: 168.00, Max: 168.00	(L)	fixed	testing	(327) e: True; a: True; c: False; b: False	True
□	KFPA IF-Langston,White	TGBT09C_525, 09C, TGBT09C_525	A	0.0	(425) Total: 168.00, Grade: 4.00, PSC: 168.00, Max: 168.00	(K)	fixed	testing	(326) e: True; a: True; c: False; b: False	True
□	KFPA Line obs-Langston,White	TGBT09C_524, 09C, TGBT09C_524	A	0.0	(424) Total: 168.00, Grade: 4.00, PSC: 168.00, Max: 168.00	(K)	fixed	testing	(325) e: True; a: True; c: False; b: False	True
□	KFPA mapping-Langston	TGBT09C_523, 09C, TGBT09C_523	A	0.0	(423) Total: 168.00, Grade: 4.00, PSC: 168.00, Max: 168.00	(K)	fixed	testing	(324) e: True; a: True; c: False; b: False	True
□	KFPA P/F-Langston,White	TGBT09C_522, 09C, TGBT09C_522	A	0.0	(422) Total: 168.00, Grade: 4.00, PSC: 168.00, Max: 168.00	(K)	fixed	testing	(323) e: True; a: True; c: False; b: False	True
□	KFPA-Langston	TGBT09C_521, 09C, TGBT09C_521	A	0.0	(421) Total: 168.00, Grade: 4.00, PSC: 168.00, Max: 168.00	(K)	fixed	testing	(322) e: True; a: True; c: False; b: False	True

Filter

By session type
All
open
fixed
windowed
vlbi
maintenance
commissioning

By observing type
All
radar
vlbi
pulsar
continuum
spectral line
maintenance
calibration
testing

By frequency
All
0.0
0.34
0.45
0.6
0.8
1.07
1.44
2.0
2.17
5.0
9.0
13.7
22.2
32.75
44.0
48.28
90.0
340.0

Release: Schedulers Interface

- Recall that the Beta Test's Schedulers Interface was not sophisticated enough.
- But creating a more sophisticate interface in the browser would mean more JavaScript, and we'd had enough of that in the Beta Test (even supporting just one browser!).
- Decided to be bold and try out a completely new technology: Google Web Toolkit

Google Web Toolkit

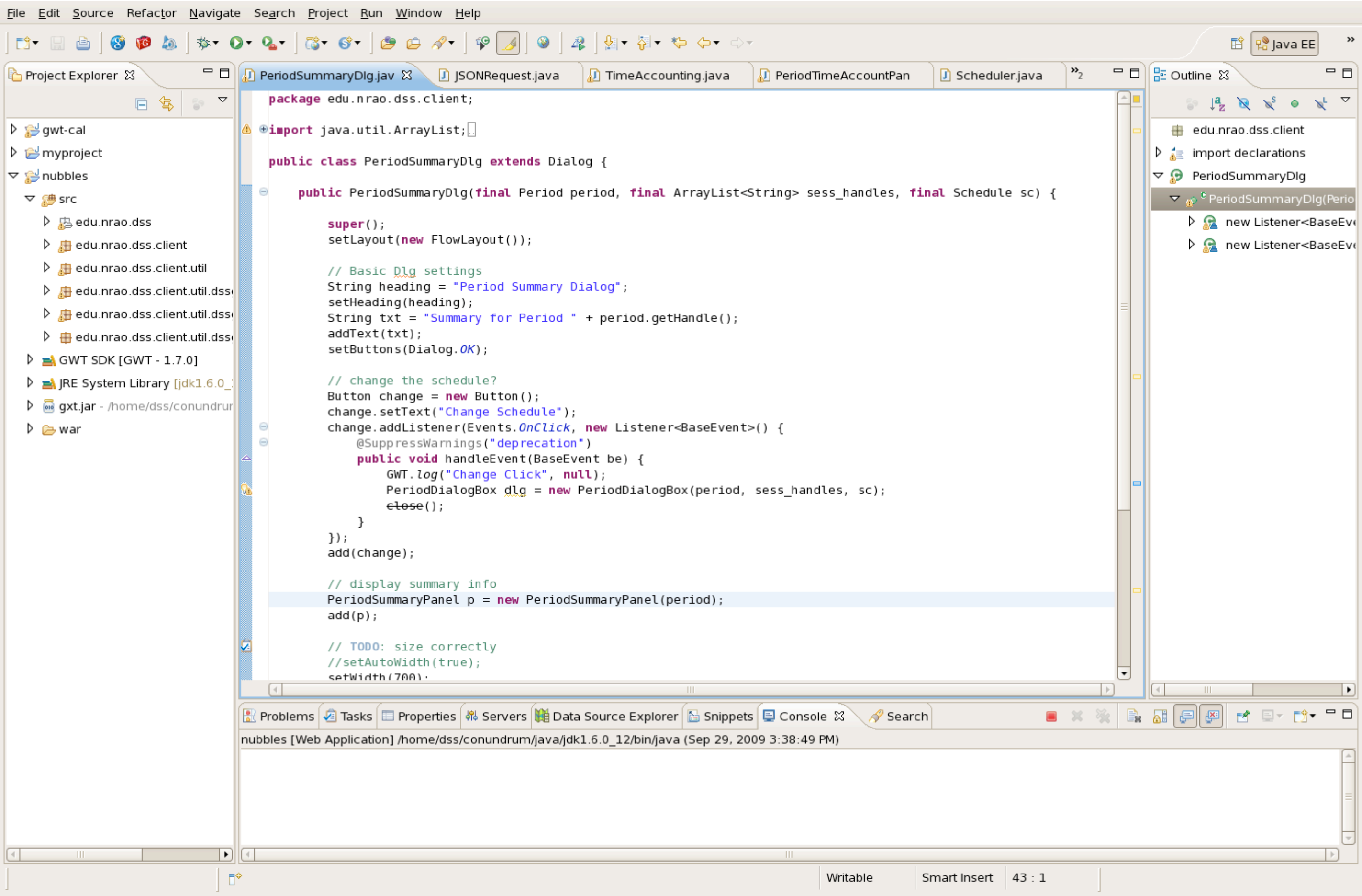
- **Google Web Toolkit** (GWT) is an open source set of tools that allows web developers to create and maintain complex JavaScript front-end applications in Java.
- Using GWT, developers can rapidly develop and debug AJAX applications in the Java language using the Java development tools of their choice.
- When the application is deployed, the GWT cross-compiler translates the Java application to standalone JavaScript files that are optionally obfuscated and deeply optimized.
- Creating an advanced UI for the browser feels like developing a typical Java Desktop interface.

Google Web Toolkit

- GWT Java-to-JavaScript Compiler – code in Java!
- GWT Hosted Web Browser – run app as Java in the JVM; great for debugging
- JRE emulation library – common features of java.lang and util for JavaScript
- GWT Web UI class library – a set of custom widgets
- A set of custom interfaces and classes for creating widgets.
- GWT handles all cross-browser issues for the developer.
- JUnit integration
- Lot's of 3rd party libraries; including one for even fancier widgets: Ext GWT

GWT and Eclipse

We found it easiest to develop GWT with the GWT plugin for Eclipse:

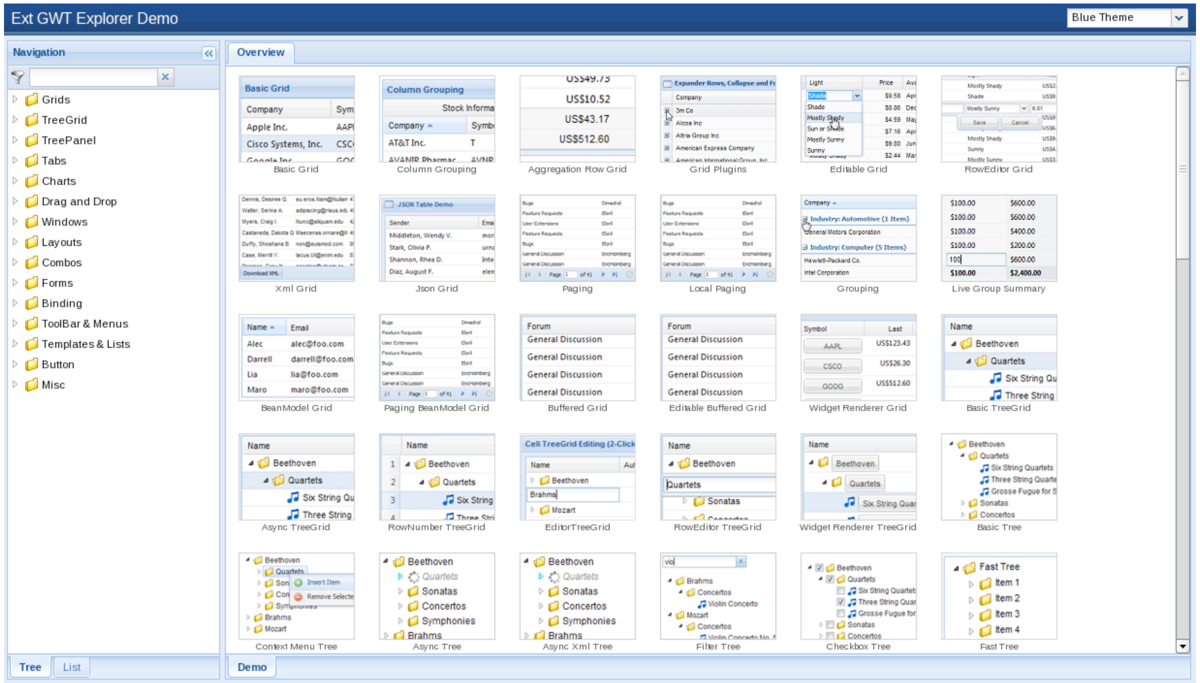


```
package edu.nrao.dss.client;

import java.util.ArrayList;

public class PeriodSummaryDlg extends Dialog {

    public PeriodSummaryDlg(final Period period, final ArrayList<String> sess_handles, final Schedule sc) {

        super();
        setLayout(new FlowLayout());

        // Basic Dlg settings
        String heading = "Period Summary Dialog";
        setHeading(heading);
        String txt = "Summary for Period " + period.getHandle();
        addText(txt);
        setButtons(Dialog.OK);

        // change the schedule?
        Button change = new Button();
        change.setText("Change Schedule");
        change.addListener(Events.OnClick, new Listener<BaseEvent>() {
            @SuppressWarnings("deprecation")
            public void handleEvent(BaseEvent be) {
                GWT.log("Change Click", null);
                PeriodDialogBox dlg = new PeriodDialogBox(period, sess_handles, sc);
                ~~close~~();
            }
        });
        add(change);

        // display summary info
        PeriodSummaryPanel p = new PeriodSummaryPanel(period);
        add(p);

        // TODO: size correctly
        //setAutoWidth(true);
        setWidth(700);
```

```
nubbles [Web Application] /home/dss/conundrum/java/jdk1.6.0_12/bin/java (Sep 29, 2009 3:38:49 PM)
```

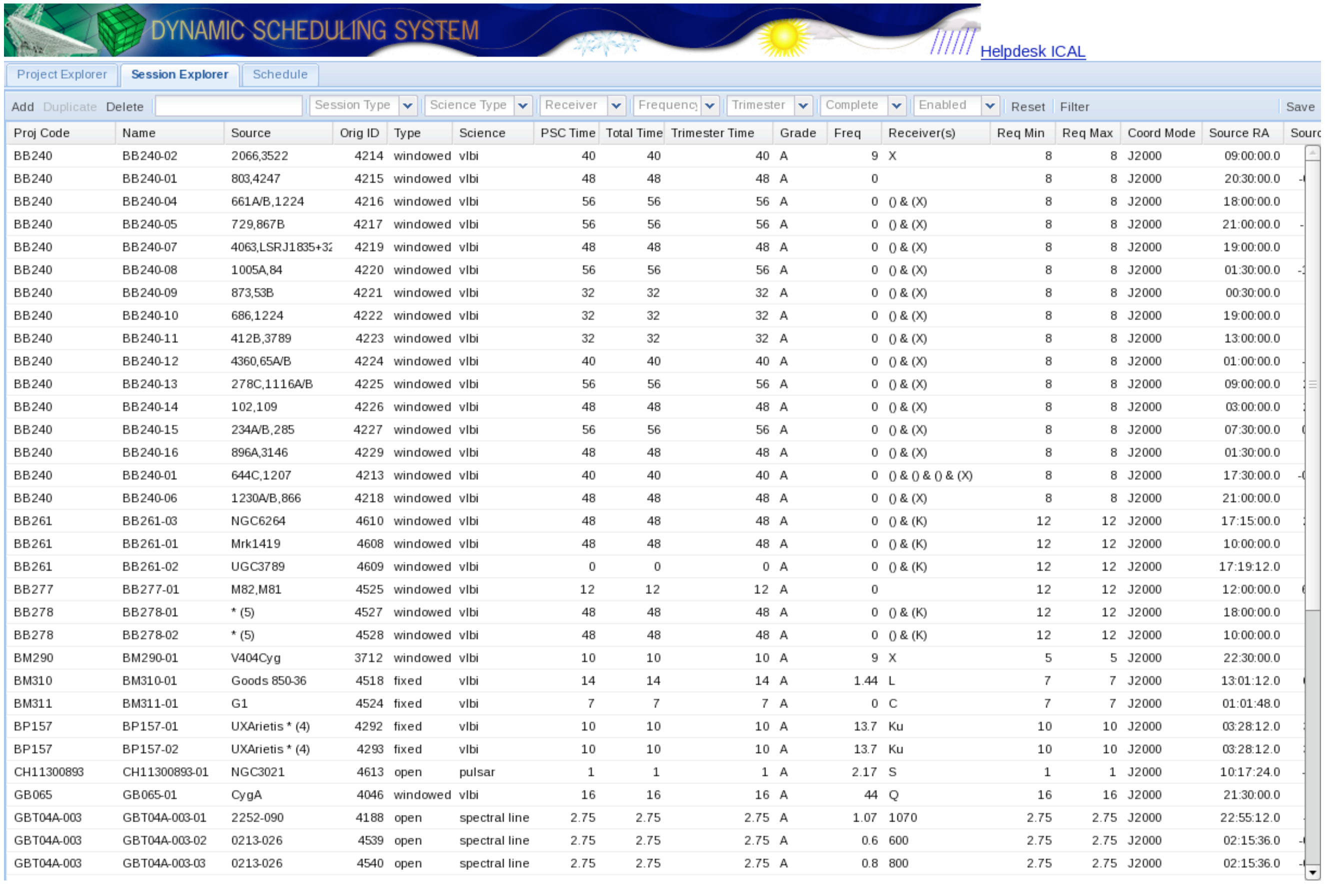
Ext GWT

- Provides a library of advanced widgets for GWT
- See the demo at: http://www.extjs.com/explorer
- We took advantage of the Grid libraries



Release: Scheduler’s Tools with GWT

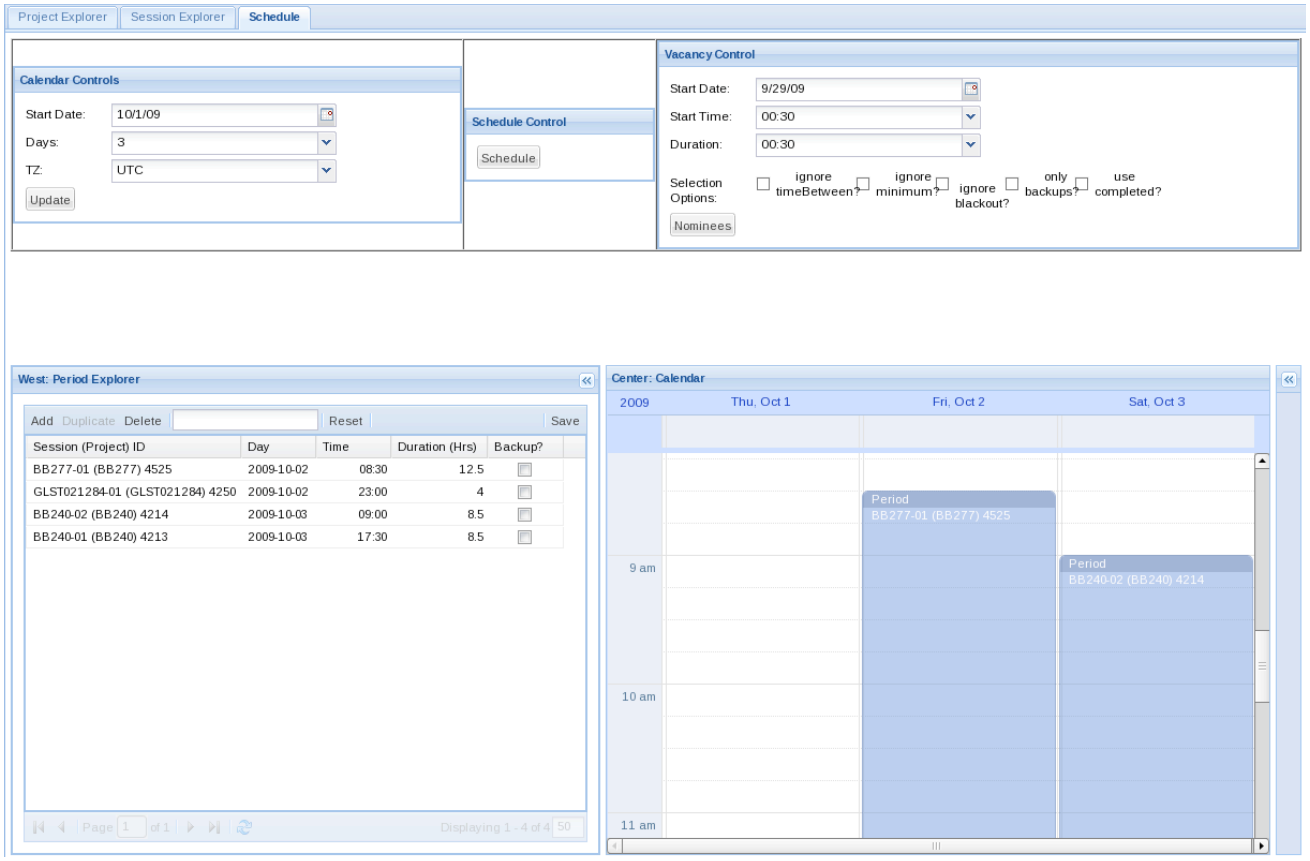
- Created using GWT and the Ext GWT Grid library.
- Emulates a spreadsheet for project data.



Proj Code	Name	Source	Orig ID	Type	Science	PSC Time	Total Time	Trimester Time	Grade	Freq	Receiver(s)	Req Min	Req Max	Coord Mode	Source RA	Sourc
BB240	BB240-02	2066,3522	4214	windowed	vlbi	40	40	40	A	9	X	8	8	J2000	09:00:00.0	
BB240	BB240-01	803,4247	4215	windowed	vlbi	48	48	48	A	0		8	8	J2000	20:30:00.0	
BB240	BB240-04	661A/B,1224	4216	windowed	vlbi	56	56	56	A	0	() & (X)	8	8	J2000	18:00:00.0	
BB240	BB240-05	729,867B	4217	windowed	vlbi	56	56	56	A	0	() & (X)	8	8	J2000	21:00:00.0	
BB240	BB240-07	4063,LSRJ1835+32	4219	windowed	vlbi	48	48	48	A	0	() & (X)	8	8	J2000	19:00:00.0	
BB240	BB240-08	1005A,84	4220	windowed	vlbi	56	56	56	A	0	() & (X)	8	8	J2000	01:30:00.0	
BB240	BB240-09	873,53B	4221	windowed	vlbi	32	32	32	A	0	() & (X)	8	8	J2000	00:30:00.0	
BB240	BB240-10	686,1224	4222	windowed	vlbi	32	32	32	A	0	() & (X)	8	8	J2000	19:00:00.0	
BB240	BB240-11	412B,3789	4223	windowed	vlbi	32	32	32	A	0	() & (X)	8	8	J2000	13:00:00.0	
BB240	BB240-12	4360,65A/B	4224	windowed	vlbi	40	40	40	A	0	() & (X)	8	8	J2000	01:00:00.0	
BB240	BB240-13	278C,1116A/B	4225	windowed	vlbi	56	56	56	A	0	() & (X)	8	8	J2000	09:00:00.0	
BB240	BB240-14	102,109	4226	windowed	vlbi	48	48	48	A	0	() & (X)	8	8	J2000	03:00:00.0	
BB240	BB240-15	234A/B,285	4227	windowed	vlbi	56	56	56	A	0	() & (X)	8	8	J2000	07:30:00.0	
BB240	BB240-16	896A,3146	4229	windowed	vlbi	48	48	48	A	0	() & (X)	8	8	J2000	01:30:00.0	
BB240	BB240-01	644C,1207	4213	windowed	vlbi	40	40	40	A	0	() & () & () & (X)	8	8	J2000	17:30:00.0	
BB240	BB240-06	1230A/B,866	4218	windowed	vlbi	48	48	48	A	0	() & (X)	8	8	J2000	21:00:00.0	
BB261	BB261-03	NGC6264	4610	windowed	vlbi	48	48	48	A	0	() & (K)	12	12	J2000	17:15:00.0	
BB261	BB261-01	Mrk1419	4608	windowed	vlbi	48	48	48	A	0	() & (K)	12	12	J2000	10:00:00.0	
BB261	BB261-02	UGC3789	4609	windowed	vlbi	0	0	0	A	0	() & (K)	12	12	J2000	17:19:12.0	
BB277	BB277-01	M82,M81	4525	windowed	vlbi	12	12	12	A	0		12	12	J2000	12:00:00.0	
BB278	BB278-01	* (5)	4527	windowed	vlbi	48	48	48	A	0	() & (K)	12	12	J2000	18:00:00.0	
BB278	BB278-02	* (5)	4528	windowed	vlbi	48	48	48	A	0	() & (K)	12	12	J2000	10:00:00.0	
BM290	BM290-01	V404Cyg	3712	windowed	vlbi	10	10	10	A	9	X	5	5	J2000	22:30:00.0	
BM310	BM310-01	Goods 850-36	4518	fixed	vlbi	14	14	14	A	1.44	L	7	7	J2000	13:01:12.0	
BM311	BM311-01	G1	4524	fixed	vlbi	7	7	7	A	0	C	7	7	J2000	01:01:48.0	
BP157	BP157-01	UXArietis * (4)	4292	fixed	vlbi	10	10	10	A	13.7	Ku	10	10	J2000	03:28:12.0	
BP157	BP157-02	UXArietis * (4)	4293	fixed	vlbi	10	10	10	A	13.7	Ku	10	10	J2000	03:28:12.0	
CH11300893	CH11300893-01	NGC3021	4613	open	pulsar	1	1	1	A	2.17	S	1	1	J2000	10:17:24.0	
GB065	GB065-01	CygA	4046	windowed	vlbi	16	16	16	A	44	Q	16	16	J2000	21:30:00.0	
GBT04A-003	GBT04A-003-01	2252-090	4188	open	spectral line	2.75	2.75	2.75	A	1.07	1070	2.75	2.75	J2000	22:55:12.0	
GBT04A-003	GBT04A-003-02	0213-026	4539	open	spectral line	2.75	2.75	2.75	A	0.6	600	2.75	2.75	J2000	02:15:36.0	
GBT04A-003	GBT04A-003-03	0213-026	4540	open	spectral line	2.75	2.75	2.75	A	0.8	800	2.75	2.75	J2000	02:15:36.0	

Release: Scheduler's Tools with GWT

- Created using GWT and the gwt-cal package
- Emulates a google-style calendar



Calendar Controls

Start Date: 10/1/09
Days: 3
TZ: UTC

Vacancy Control

Start Date: 9/29/09
Start Time: 00:30
Duration: 00:30
Selection Options: □ ignore timeBetween? □ ignore minimum? □ ignore blackout? □ only backups? □ use completed?

Session (Project) ID	Day	Time	Duration (Hrs)	Backup?
BB277-01 (BB277) 4525	2009-10-02	08:30	12.5	□
GLST021284-01 (GLST021284) 4250	2009-10-02	23:00	4	□
BB240-02 (BB240) 4214	2009-10-03	09:00	8.5	□
BB240-01 (BB240) 4213	2009-10-03	17:30	8.5	□

NRAO

Scheduler's Tools with GWT: conclusions

- Team was inexperienced using Java & Eclipse. A learning curve when we really didn't have time for it.
- Using GWT to emulate some of the more basic Desktop widgets (panels, combo-boxes, etc.) was relatively straight-forward.
- Implementing the more advanced widgets, such as the grids (Project Explorer) with Ext GWT were more difficult.
- So, is the problem Ext GWT, GWT, or us?

DSS Release: Science Algorithms

- Recall that in the Beta Test, our Science Algorithms were coded in Python.
- This made it trivial to integrate into the application, but was slow and error prone.
- For the release, we again decided to make the bold move of rewriting the algorithms in a completely different language: Haskell

Science Algorithms with Haskell

- **Haskell** is a standardized, general-purpose purely functional programming language, with non-strict semantics and strong static typing.
- **Purely functional** is a term in computing used to describe algorithms, data structures or programming languages that exclude destructive modifications (updates). According to this restriction, variables are used in a mathematical sense, with identifiers referring to immutable, persistent values.
- A different programming paradigm then Python, C/C++.
- Referential Transparency: put the same thing in, always get the same thing out. It's State-less.
- No 'for' loops: everything is recursion and pattern matching

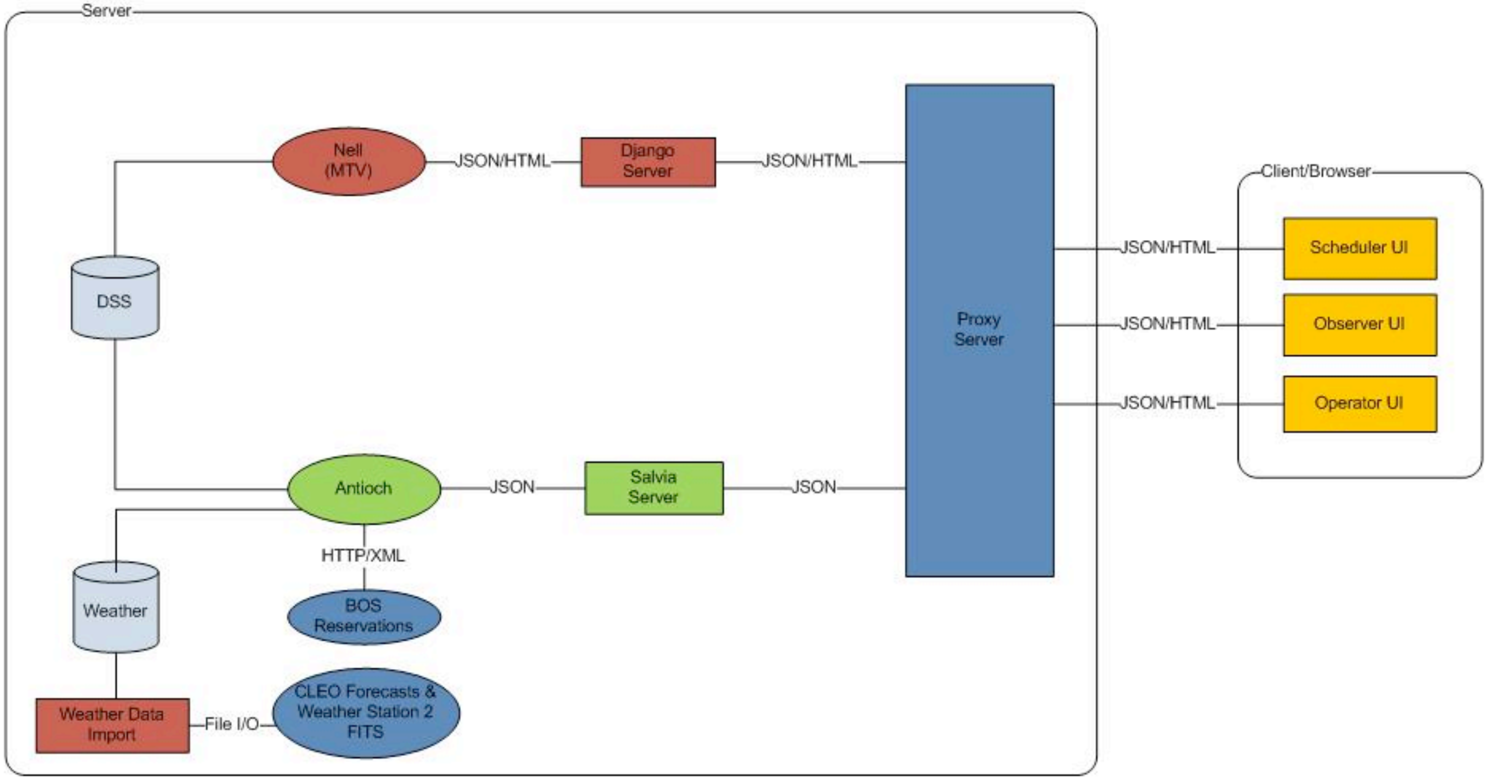
NRAO

Science Algorithms in Haskell

- Pros:
  - Reduced code size by a factor of 2
  - Increased performance by a factor of 10 !!!
  - Mathematical nature of Haskell more closely resembles science.
  - Code easier to understand by team scientists
- Cons:
  - Steep learning curve: a completely different programming paradigm for us
  - Not as many practical third party libraries as a language like Python (though enough)
  - Will someone be around to support this in 10 years?
  - Doesn't interface easily with other languages (just C right now)

Putting it all together:

- When you use what you think is the tool for each part of your problem, how do you get them to all work together?
- Proxy server – let the browser think there's just one server



Conclusions

- Inexperienced team delivered two web applications (Beta, Release) on time and on budget, using new tools and environments.
- We would strongly recommend using some kind of framework.
- There's lots of tools and resources to help build a better user experience (iCal, RSS, etc.).
- GWT has a lot of promise. Would strongly recommend it if you are already familiar with Java.
- Haskell is a great programming language, but is it worth the risk?