Table of Contents

Draft Operational Roll Out Plan for Stellar 2001 v.2

Mike Barker

This plan contains a number of sections, dealing with various aspects of transitioning Stellar from development to operational use.

Stellar Scheduling Items

- Training
- HW setup (UPS, RAID, etc.)
- Documentation
- Support (training, familiarization)
- Release (when do we add new functions? Incremental or big-bang?)
- Cutover

Key approach comments:

- Integration and test done on a separate system; users on dedicated "operations" systems (this month will be an Ultra 5; July 13 the E250 will be ready for end user use)
- "rings" of users: FLs and Key Adopters first, then spreading to Faculty, TAs and students
- Training occurs through (1) embedded short-courses (i.e. using Stellar to teach Stellar) (2) one-on-one via FLs and Key Adopters (3) short intensives if needed
- Documentation: key documentation is embedded help. An additional quick-reference can be developed if needed
- Support: use existing support. E.g., SMA will continue to use faculty liaisons. Other users will use the IS help desk, IS consultants, departmental experts, etc. Key is to provide those people with early familiarization and continuing backup support (i.e. they answer questions wherever they know the answers, and have a quick response route to pose questions they don't know how to answer).
- Service: use existing service organizations. E.g. IS Athena Operations and similar IS teams provide 24x7, 365 day service. We should build on those services. The EMCC has already started conversations around this.
- Release: for integration and testing purposes, it is useful to have small incremental releases. However, development this summer will result in a number of additions being available during August and September. I propose that we "clump" these additions into release 1.1, release 1.2, and later. We may conduct integration and test at a finer level (e.g. whenever a specific tool is ready), but for planning purposes we should focus on release 1.1 in early August and release 1.2 in late August/early September.
- Cutover: the system will be ready for use in July. Additional functions and additional hardware can be added after that. From the point of view of support and service, keeping the students, faculty, TAs on a single system is highly preferable. Further, while we can transfer data from an older system such as Hi-Command, we do not want to do this on a regular basis (the manual fixup required is not desirable long-term). We should not plan on trying to run both systems in parallel for an extended period of time.

Stellar Summer Additions

- Discussion Board (Jive, Panfora)
- Websis, MIT libraries, MIT Event calendar integration
- MyStellar "miniportal" – student, faculty
- Reuse and shared content tools
- Additional styles (look-and-feel)
- Course setup tools
- User management tools
- Assessment tools (quiz/test, homework handling, survey)
- FAQ management
- Chat/zephyr/IM
- uportal integration

Release 1.1: Jive, Websis, MIT libraries, MIT event calendar, MyStellar "miniportal" student and faculty, additional styles, course setup, user management
Release 1.2: Assessment tools
Later Releases: reuse and shared content tools, FAQ management, Chat/zephyr/IM, uportal integration

Hardware Support

- We can easily bring up additional "alpha" systems using standard Athena workstations. These will be particularly useful in familiarizing users with the system.
- We have an Enterprise 250 available for an early operational site. We should add memory (up to 2 Gbyte possible) and at least one additional disk (for RAID/Mirroring purposes). The system should also be put on UPS or moved to W91 (I prefer W91).
- Disk Storage: the SMA courses are estimated at 50-70 Mbytes each. Assuming 100Mbytes each, there have been about 45 courses so far, requiring 4.5 Gbytes. At 15 courses per semester, this will grow about 1.5 Gbytes each semester.
- The E250 comes with 18Gbyte standard, and can have up to 6 18G or 36G disks internally. When we need further storage, external storage can easily go up to about 900 Gbytes. I propose for the near-term that we buy 2 18G disks and run them in a mirrored fashion (to ensure data backup on the fly).
- Multiple servers vs. single server: while it may seem desirable for accounting and other purposes to put each project or program on an individual server, this reduces one of the key benefits of Stellar: reuse and sharing of content. This benefit is particularly useful for faculty. Further, it is quite feasible to obtain economies of scale through having multiple programmes and projects on a single server.
- Fallback/Disaster Recovery: see risk analysis.

Testing Approach

- Testing consists of four major phases: usability, functional, error, and load.
- Usability testing has been performed repeatedly during the design and implementation work. It will continue during familiarization to help identify any remaining areas for improvement. This has also included review and testing for accessibility.
- Functional testing is performed by the programmer during unit testing. It is repeated during integration and test, when all of the major functions of the system are exercised. See the Draft Testing Protocol attached. (Regression testing also is performed whenever changes are made. This consists of repeating all the functional tests to verify that only the expected changes have occurred.)
- Error testing is similar to functional testing, however various error conditions are deliberately invoked. The Draft Testing Protocol indicates some of the error conditions which will be tested.
- Load testing consists of performing large numbers of simultaneous operations. We will test both performing a large number of the same operations, and performing a large number of a variety of operations. This can be partially automated, however, we expect to perform some testing by having a number of people attempt operations at the same time.
- Disaster recovery testing is basically a combination of error and functional testing. For example, we will load a variety of information on the system and deliberately induce various serious system errors. Then we will perform the designated error recovery (reload the database, recover from backup, etc.) and test to be sure that the system is fully functional.
- We should consider the use of a "database snapshot" to allow testing against a known normal state.

Contingency/Risk Planning

There are several layers of possible risk to consider.

1. Hardware problems:
    a. CPU: any comparable Sun system will work. For optimal response, we may want to purchase a standby system. This could be kept online, ready for use, with Oracle data mirroring.
    b. Disk: as mentioned above, I propose that we consider simple mirroring as our first protection against these failures. We will also put the system on regular backups.
    c. Network: while this has low likelihood, we can consider a mirrored system (in Singapore for SMA, other sites for other programs). The simplest approach would be to enforce that the MIT Stellar site is the "master" site where all updates occur, while the other sites mirror the data and can provide access services.
    d. Power, etc.: the system should be put in a server environment. W91 provides this kind of service, and we have an existing system there. I propose that we host the new Stellar servers there.
2. Software:
    a. Solaris: well-known and robust operating system. We will need to ensure that appropriate updates/patches are in place. The easiest way to do this is to use the Athena services. If there are problems, the operators should be able to restart the system easily.
    b. Oracle: well-known and robust. Has rollback and other mechanisms which we could use if needed.
    c. Apache webserver: well-known and in wide use at MIT. Can restart easily if needed (and should be automatically restarted on a reboot of the system).
    d. Other packages: Xalan, etc. Mostly well-known systems with alternatives available if needed (some commercial).
    e. Stellar software: recently developed and not as heavily tested through use. Mostly, we will need to keep an eye on this and ensure that the Stellar developers and maintainers are ready to deal with problems when they are found.
3. Data
    a. Disaster recovery: disk mirroring, backups, replicated database
    b. Data integrity: versioning and database rollback allow some recovery from data problems. Editing and correction of entries also provides recovery. Detection of data problems can be done through integrity checking programs (not currently planned for implementation at this point).

Communications and Publicity

1. At key points in the rollout, we should provide written summaries of the work done. E.g., when integration and testing occur, we should summarize the results. These can be posted to the rollout website (see below) and email notification sent to the team and stellar-announce (see below).
2. We should provide MIT publications with articles about Stellar. For example, we will have an article in the IS Insider this fall. We can also place articles in the Faculty Newsletter and the various campus publications. We will continue to put articles into the larger press, such as Syllabus, ACM eLearn, and others.
3. For information about the status of the rollout, we should setup a mailing list (stellar-announce). This should be a public list, with faculty liaisons, early adopters, and anyone else interested in Stellar.
4. For suggestions, bug reports, and similar incoming information, we should setup a mailing list (stellar-suggest). We should publicize the address of this list, but the membership should be restricted to those on the team who are answering.
5. For support of the faculty liaisons and early adopters, we should setup a mailing list (stellar-support).
6. We should designate a "monitor" to collect information from the mailing lists and develop FAQ entries.
7. We should set up a "web site" for the rollout. This site will provide basic links to the various servers, cgiemail forms for comments, suggestions, bug reports, etc., papers or links to various resources. This can use the current stellar.mit.edu space, but should be expanded to act as a working entry to the rollout.

Stellar Overview Schedule

Dates	Development	Integration/Test	Users
June	Core Done*		Familiarization (early adopters)
July		Operations Server (E250) Ready for Use	Training And Familiarization (faculty, TAs, students)
August	Release 1.1	Release 1.1	Release 1.1
September	Release 1.2	Release 1.2	Release 1.2
October			
November			
December			
January	OKI Release 1.0		

* there are continued integration fixes at this point, however the main code base is in place and operational.

Stellar 3 Month Schedule

Dates	Development	Operations	Use
June 22	Core Done	Integration/Test	Familiarization using UI
June 29		Report on test results	Familiarization using "alpha" site
July 6			
July 13		Operations server (E250) Ready for Use	
July 20			FLs and key adopters use E250
July 27			
August 3	Release 1.1	Integration/test	
August 10		Report on test results	
August 17		Release 1.1	
August 24			
August 31	Release 1.2	Integration/test	

Draft Testing Protocol

Stellar-June	**Functional Test**	**Error Tests**
Student View		
SMA logo	Links to SMA site on every page	
Course Title	Appears	Null Title; Very Long Title
Topics	Can View Topics Page	View Empty Topics Page; not logged in; not authorized for class
People		
Staff List	Can View Staff List (and has correct members)	Empty Staff List; Very long staff list; not logged in; not authorized for viewing
Class List	View class list	Empty class list; very long class list; not logged in; not authorized
Announcements	View announcements	Empty announcements; not logged in; not authorized
Materials		
Readings	View Readings; download readings files; go to readings URLs; view online entry text	No readings; long set of readings; URL does not exist; online text contains HTML; not logged in; not authorized
Lecture Notes	View lecture Notes; download lecture files; go to lecture URLs; view online entry text	No lecture Notes; long set of lectures; URL does not exist; online text contains HTML; not logged in; not authorized
Videos	View Videos	No Videos; video server down; Video links misdirected; not logged in; not authorized
Other	View other materials; download other materials; go to other URLs; view online	No other; long set of other materials; URL does not exist; online text contains HTML;

	entry text	not logged in; not authorized
Homework		
Assignments	View assignments; download assignments; go to assignment URLs; view online entry text	No assignments; long set of assignment materials; URL does not exist; online text contains HTML; not logged in; not authorized
Solutions	View solutions; download solutions; go to solutions URLs; view online entry text	No solutions; long set of solutions; URL does not exist; online text contains HTML; not logged in; not authorized
Home	Links to course homepage on all pages	Not logged in; not authorized
Calendar		
All Items	View items and dates are correct; can download; link to URLs; online text	No items; long set of same date items; long set of items overall; URL does not exist; not logged in; not authorized
Discussion (webboard)	Links to course Web board	Web board server down; not logged in; not authorized
Help		
Class Help	Displays proper help for context; download help files	Not logged in; not authorized
Stellar Help	Displays proper help for context	Not logged in; not authorized
Faculty View		
Edit Class Description (type entry)	Able to edit class description	Empty description; not logged in; not authorized
Topics Page (new topic, change topic order, edit topic, delete topic, add document, change document order, edit document, delete document)	Add new topic change topic order edit topic delete topic add document change document order edit document delete document	Empty fields; long fields; return inside fields; point to non-existent URL; not logged in; not authorized

Readings Page (edit and delete)	Edit readings (change type, etc.) Delete readings	Not logged in; not authorized
Lecture Notes (edit and delete)	Edit lecture Notes (change type, etc.) Delete lectures	Not logged in; not authorized
Assignments (edit and delete)	Edit assignments (change type, etc.) Delete assignments	Not logged in; not authorized
Solutions (edit and delete)	Edit solutions (change type, etc.) Delete solutions	Not logged in; not authorized
Class Help page (Edit Help, add help document, edit help document, delete help document)	Edit Help Add help document Edit help document Delete help document	Not logged in; not authorized
Add announcements, edit announcements, change order (online entry)	Add announcements Edit announcements Change order	Not logged in; not authorized

Disaster Recovery Tests (examples):

1. Database failure. Shut down system and remove database. Restore from backup and test system.
2. Disk failure. Shut down system and remove one disk. Restart using other disk and verify functional.
3. System failure. Start with stock system (e.g. Ultra) and install system, then restore data from backup. Restart and test system. (Note: some loss of data would be expected in this case)
4. (After mirror system setup) Network problems. Shut down system. Verify that other system provides useful set of classes and operates independently.