SAM Grid Team Work Plan for the July 2002 Milestone

The plan is mostly *feature-oriented*. Coordination work will continue in the background. Three groups have been identified.

FNAL team

It has a minimal, most pessimistic plan, as well as highly desired enhancements which will be contingent on the team manpower as well as deliverables from collaborators. Depending on how much work the minimal workload really turns out, the bulk of the work may in fact go into the enhancements, such as 2.6.

1. The Minimal (pessimistic) workload. Deliverable feature: fully distributed basic monitoring of the system and of unstructured SAM and MC jobs, current state only.

TASK	Target Date	Comments
1.1. Understand the monitoring use cases and therefore define the problem(s) to solve.	6/1/02	Done?
1.2. Continue deployment of the prototype Grid.		
1.2.1. At a minimum, maintain 2 Linux and 1 IRIX “sites” at FNAL, 1 at UK-IC and 2 at UTA. Ensure full connectivity on the graph, i.e. enable job submission (from anyone) from any vertex to any vertex.	6/15/02	Done 6/18/02, 3Linux+1Irix at FNAL.
1.2.2. Deploy MDS-2 at all the sites: install, configure, test with dummy schemas.	6/21/2002	
1.2.2.1. Ensure that MDS instances share information as needed for site autonomy and global information collection.	6/21/2002	Done 6/15/2002
1.2.3. Deploy SAM stations and/or MCFarm at each vertex. Run a simple (or toy) SAM application and/or MC application in a semi-automated fashion.	07/01/2002	Need 1 CDF station At FNAL
1.3. Design the views of the system that will be enabled.	06/21/2002	Andrew – preliminary design. Revisited later once experience gained
1.3.1. Basic view of the system		
1.3.2. SAM analysis jobs and their detail		
1.3.3. MC jobs – interact with UTA on understanding the desired level of detail to display.		
1.3.4. Aggregations and cross-references of the views, “drill-downs”.		
1.4. Establish an MDS schema for the prototype.		
1.4.1. Understand the policies involved in global ID allocation	06/15/2002	Done
1.4.2. Obtain ID space, if necessary, for the FNAL/D0/SAM domain – interact with Globus as necessary	06/15/2002	Applied with IANA 06/14/2002, c/o Ruth
1.4.3. Reuse as much as possible from other schemas, including any Globus ideas,	07/01/2002	May not have unique answer

EDG ideas or inter-grid coordination ideas. Decide how to maintain reasonable independence for the duration of the prototype deployment and experimentation, while benefiting from other efforts.		
1.5. Research for available LDAP/MDS browsers. Install, configure, test anything that is relevant. Evaluate for immediate applicability.	06/21/2002	
1.6. If necessary, enhance an existing LDAP browser so as to display useful information, or develop a new one using JSP – the “front end”.	07/21/2002	
1.7. Develop information providers in Python and Shell scripts – the “back end”. Integrate the specialized providers from UTA for the MC jobs.	07/15/2002	
1.8. Integrate the front and back ends, deploy Web servers, operate the prototype system, as necessary to display the principal capabilities for the user community as well as the Grid community.	07/31/3001	Marks end of this part of the project

2. Optional Enhancement: Mining of historical data for monitoring of past state such as history of recent jobs. **Assigned to John Weigand with Igor.**

2.1. Understand the architecture (not necessarily the code) of the SC2001 demo.	06/01/2002	Done
2.2. Design the logging message format, based on studies of XML and the ideas from EDG.	06/15/2002	Done, Attr=Value …
2.3. Clone the SAM logging service while using the new message format as necessary. Use the Python API (no C/C++ clients expected).	07/01/2002	
2.4. Choose the database (mysql or oracle).		Done, Oracle
2.5. Save selected information from the log file in the database.	Not until job submission tools tested	
2.6. Amend the clients that mined SC2001 database to mine the new database for the jobs and other new entities to be displayed.	07/21/2002	

3. Optional enhancement. Deliverable feature: basic brokering of SAM and MC jobs, with some display of the affecting factors. **Contingent on Condor deliverables and Gabriele available > 20%.**

3.1. Elaborate on the architecture, make initial decisions, define package structure	7/17/2002	

3.2. Choose the primary mechanism to advertise resources, between Condor ClassAds and MDS information providers.	7/1/02	DONE. Advertise Condor ClassAds
3.3. Develop information converter between ClassAds and MDS in the direction decided by the above choice.		These should exist at some level in either direction. Ruth will look next week.
3.3.1. Learning how to query the collector remotely, understand the classAd output		
3.3.2. Convert the above classAd output into LDAP, plug into MDS.	Assumes MDS schema includes it	
3.3.3. Tailor the appropriate package, see 3.1, for this		
3.4. Augment the basic system view, and the schema respectively, with the characteristics important for the brokering	Initial- 8/09/2002	
3.4.1. Include characteristics of a station – SAM software availability, station name, cache size, project activity, etc.		
3.4.2. Include characteristics of the farm MCFarm software availability, number of processors, etc – interact with UTA as needed.		
3.4.3. If necessary, augment the client for the system view as to include the above characteristics (not necessary of a standard, universal LDAP browser is adopted).	Depends on 1.5	
3.4.4. Develop the Condor ranking function based on the first, most intuitive considerations. Specialize the ranking function for the two orthogonal cases:		
3.4.4.1. SAM analysis jobs – require SAM station and prefer cached files.		
3.4.4.2. MC jobs – require MCFarm software and prefer a lot of CPU.		
3.4.5. Develop the package that	Contingent on Condor	

configures the Condor MMS	fixes for parameter passing	
3.5. Implement a job submission client that accepts a job description in a loosely defined language (bare-bones Condor ClassAds is perfect) and formats a Condor JDL, incorporating the above ranking function, such that the job is suitable for submission to the Request Broker from CondorG.	08/29/2002	

UTA Group

1. Deploy Condor-G and take the testbed to the next level – the prototype Grid		
1.1. Two UTA farms		Done?
1.2. One dedicated machine for SAM station and analysis jobs		
2. Run MCFarm in the new farms as they become part of the prototype Grid, submitting the jobs via CondorG.		
2.1. When request broker is available from the FNAL-Condor collaboration, learn to submit an MC job w/o specifying the target farm, such that the broker chooses “the best” place to run		
3. Export the MC Job status from the MCFarm software to MDS such that the general D0 Grid tools may display (nearly) the same information as the Wlodek bookkeeper.		
4. Continue work on the D0 Job Definition, discuss within D0.		
5. Evaluate HawkEye and Dagman from the perspective of usability to the D0 Grid		In progress

UK-IC Group

1. Finish making Condor one of SAM’s supported batch systems, using the recently introduced interface		
1.1. Support BS-specific JDF’s		
1.2. be able to start multiple jobs		

2. Introduce GSI into SAM for the purposes of data transfers		
2.1. Understand if and how CAS may be used. Evaluate the need to write own AAA service		CAS is in alpha. It is being rewritten in response to site security team inputs
2.2. Change the SAM server code so as to propagate user credentials from the request creation command to the file transfer command		
2.2.1. Include AAA client in station master and FSS		
2.2.2. Send a token to stagers		
2.2.3. Stagers will pass tokens through sam_gridftp		
3. Update the technology review document, rid of blank chapters one way or another		
4. Take the UK part of the CondorG testbed to the next level, namely a prototype d0 grid. Associate CondorG sites with SAM stations and run analysis jobs. Demonstrate the power of the SAM to potential collaborators. Promote and coordinate d0grid testbed expansion in EUROPE.		