Evan says, and we even had some evidence, that Renderman doesn't work reliably if your files use mapped drives. (It doesn't make any sense to me.) Before users can submit a Maya scene to the farm, users run their scene file through a Python script called FixOscar. It writes a copy of your scene file, and any files it references, such that the new files all have UNC path names. You can then submit this scene to the render farm. The script takes 15 minutes to run.

Work item #1: Research whether Renderman renders that use mapped drives really don't work on the farm. If mapped drives work or can be made to work we can skip the onerous conversion step. To test: instead of running render.exe on the render machines, run a wrapper script that can map the W: drive before rendering. Do "net use" before and after the render to verify that the drive got mapped and stayed mapped.

Once jobs are on the farm, there is no way to remove them, prioritize them, kill them, or preempt them.

Work item #2: Add priority fields to jobs and minimum-priority fields to render nodes such that a) render jobs are selected in priority order and b) a machine won't run a job whose priority is lower than the machines threshold.

Work item #3: Make a gui that can remove/kill/prioritize/preempt a job or task.

There are little scripts that register or unregister a machine as a render node. Nobody uses them and you can't easily tell what your machine is doing.

Work item #4: Make a little gui that says whether your machine is online/offline/working as far as the render farm knows. Have buttons for Online/Offline/Get off!. That last one kills any running render job on your machine.

Currently the FarmView GUI program generates reports about render node status and job status based solely on the Hydra database. The data isn't tied to what's actually happening in the real world.

Work item #5: Enhance the FarmView program to get real-time data from the render machines.

So, what to do next?

They're not using the farm, supposedly because they need instant turnaround for doing tests. We need to make it so that a sufficiently motivated artist can make his job run right away. Currently they log into a farm or user machine and do a batch render. That means that they can get their file to render correctly on that other machine, so if it uses mapped drives, they must certainly be mapped on that other machine. So the onerous conversion process can't really be necessary.

The next issue is that there might not be any farm machines available. If we had priorities they could make their job high priority, and if we had pre-empting, we could pre-empt some lower-priority jobs to free up processors. That seems sufficient?