

Prodsys2 Meeting

# Core components of the DEFT Web UI and the immediate work plan

Maxim Potekhin

BNL

Aug 29, 2013

# Documentation and revision control

- ◆ As a reminder, the root for all of Prodsys2 TWiki documentation is this: <https://twiki.cern.ch/twiki/bin/viewauth/Atlas/ProdSys2>
- ◆ The DEFT Web UI info is here:  
<https://twiki.cern.ch/twiki/bin/viewauth/Atlas/DeftGui>
- ◆ All documentation is updated reasonably often, and contributing team members are certainly encouraged to add or modify the content as required by the project
- ◆ All code is actively maintained in SVN. The layout of the latter was modified earlier this month based on suggestions of team members, for some reasons recent code changes were lost in transition (mishap), almost fixed now
- ◆ <https://svnweb.cern.ch/trac/panda/browser/panda-deft> is the root for both deft-core and deft-ui

# General state of DEFT

- ◆ Database schemas are being brought in line with what's currently the API of JEDI- $\alpha$  and ATKR adapter. These schemas are always kept up-to-date in the SVN (\*.sql).
- ◆ As described in the S&C week in June'13, DEFT consists of two distinct components: "core" and "ui". Development of the two has only one common dependency, which is the Oracle schemas, and can proceed in parallel
- ◆ Testing of JEDI has shifted to the ADCR database, which has positives and negatives: it makes integration work more difficult, but allows for more freedom for the UI development, which can proceed fast w/o fears of affecting production

# DEFT UI: the platform

- ◆ We chose Django for multiple reasons which we won't repeat here
- ◆ A development/alpha testing machine was quickly setup at CERN after the June S&C meeting (thanks to Mr.Baranov) and the support has been quite good
- ◆ Django allows for very quick creation of template-based HTML content delivery, and is also easily instrumented with JSON serialization functionality. The assumption is that for a while these two methods will co-exist in development.
- ◆ A simple Django application has been in place for a while, serving both JSON and HTML .
- ◆ HTML: right now, no effort is put into making it pretty as it is primarily a development tool allowing for quick and dirty data visualization.

# DEFT UI: base functionality

- ◆ Record of our design documents  
[https://twiki.cern.ch/twiki/bin/viewauth/Atlas/DeftGui#Views\\_Screens\\_Functions\\_and\\_gene](https://twiki.cern.ch/twiki/bin/viewauth/Atlas/DeftGui#Views_Screens_Functions_and_gene)
- ◆ In a nutshell, the very basic functionality is covered by:
  - Two monitoring pages, for tasks and Meta-Tasks, with obvious cross-links between the two as needed
  - Template library: derivation of Meta-Tasks from prefab examples and editing/adjusting their parameters as needed (w/o change of the Meta-Task topology). Closely related to this is Meta-Task cloning functionality, which can be co-located on the same page
  - Approval and Control: administrative page reserved for managers
- ◆ The above list covers the minimal but complete functionality of the DEFT end-user and manager interface
- ◆ The good news is that it's only 4 screens total, which is not a large number. Works can be split nicely using Django modular organization. We will need a reasonable navigation bar etc.
- Administration of user access is included in Django, can be left for later will we finish alpha-testing.

# DEFT UI: timeline

- ◆ Prodsys2 must be in production in the first quarter of 2014
- ◆ In pre-production – late 2013
- ◆ DEFT UI must follow a similar trajectory – simple but workable, with base functionality in late 2013
- ◆ More polished and “final” UI in Jan-Feb 2014

# DEFT UI: project participants

- ◆ M.Potekhin – deft-core and deft-ui, total of 0.5FTE for both: 0.25
- ◆ L.Sargsyan – an unspecified fraction of FTE, at the level of 0.1-0.2. Laura brings expertise in building table base visualization and navigation elements. Django learning curve.
- ◆ A.Stradling – 0.2FTE, started work in July, on the Django learning curve.
- ◆ D.Golubkov – speaks Django, 2 month FTE TBD (?)
- ◆ S.Baranov – technical support/facilities, I believe we are covered here
- ◆ A.Petrosyan – potentially the developer of the JEDI monitoring app, which we agreed in principle could be hosted by the same service as DEFT UI. Ideally, would share visual elements to provide consistent look and feel to the users. Artem's availability and schedule for this work are undefined right now.
- ◆ A.Vaniachine – tester, maybe at the level of 0.1 FTE.

# DEFT UI: work breakdown (proposal)

- ◆ M.Potekhin – overall coordination, navigation bar, schema design and core application coding, 0.25FTE
- ◆ L.Sargsyan – integration and re-use of existing Javascript /AJAX code for visualization needs of the task and Meta-Task monitoring pages, 0.15FTE. Table Master.
- ◆ D.Golubkov – **principal** template library page and task editor developer, 2 month 1.0FTE (based on extensive expertise in validation and parsing of various task parameters)
- ◆ A.Stradling – user administration, approval and control page 0.2FTE (some work already under way)
- ◆ S.Baranov – technical support/facilities
- ◆ A.Petrosyan – code review and advice, minor coding 0.1FTE
- ◆ A.Vaniachine – tester, 0.1 FTE.

# DEFT UI: hurdles

- ◆ M.Potekhin – oversubscribed
- ◆ L.Sargsyan – learning Django, fractional FTE. Integration of the existing hbrowse code may not be trivial, since working applications present a Web framework all by themselves, so we need to avoid duplication. Factoring out AJAX/JS needs to happen reasonably soon.
- ◆ D.Golubkov – work must be structured in such a way that reasonable deliverables can occur in 2 months. Real danger of being stuck with JEDI- $\alpha$  work, this needs to be capped somehow.
- ◆ A.Stradling – learning Django, fractional FTE

# DEFT UI: proposed strategy

- ◆ Keep it simple at every level, from basic code organization to the Django module layout.
- ◆ Don't worry *too much* about visual appearance at this stage, prioritize functionality
- ◆ I propose a staged approach to the UI development
  - According to the timeline presented above, DEFT UI capability needs to be in sync with integration work
  - No matter how pretty or ugly are the Web pages, functionality must be there in late 2013 – **stage one**.
  - This needs to be the focus (even if no AJAX support exists in the first stages)
  - This buys extra time to AJAXify/prettify/jQuerify the application in Nov-Feb 2014: **stage two**. Note that some work will be done in parallel with stage one.

# DEFT UI: conclusions

- ◆ Even with diminished resources, the project can be done on schedule with proper planning and minimalistic approach to the UI
- ◆ Communications are of essence, and so far have been satisfactory, with shared and extensive documentation as well
- ◆ It may make sense for M.Potekhin to spend a few days at CERN working with Laura and Dmitry this Fall to speed up progress.