STRATEGIES FOR MIGRATING TO A NEW EXPERIMENT SETUP TOOL AT THE NATIONAL IGNITION FACILITY

Allan Casey, Richard Beeler, Chantal Fry, Jessica Mauvais, Eric Pernice, Misha Shor,
Joseph Spears, Douglas Speck, Susan West
Lawrence Livermore National Laboratory (LLNL), USA

Abstract

For the last 10 years, the National Ignition Facility (NIF) has provided scientists with an application, the Campaign Management Tool (CMT), to define the parameters needed to achieve their experimental goals. Conceived to support the commissioning of the NIF, CMT allows users to define over 18,000 settings. As NIF has transitioned to an operational facility, the low-level focus of CMT is no longer required by most users and makes setting up experiments unnecessarily complicated. At the same time, requirements have evolved as operations has identified new functionality required to achieve higher shot execution rates. Technology has also changed since CMT was developed, with the availability of the internet and web-based tools being two of the biggest changes. To address these requirements while adding new laser and diagnostic capabilities, NIF has begun to replace CMT with the Shot Setup Tool (SST). This poses challenges in terms of software development and deployment as the introduction of the new tool must be done with minimal interruption to ongoing operations

INTRODUCTION

CMT has been under steady development for almost 15 years. Initially created to commission the NIF laser, it was designed to put many low-level experiment configuration details into the hands of expert users and architected primarily around the repeating patterns of the NIF laser hierarchy: 192 beams, 48 quads, 24 bundles, four clusters, two laser bays, and one NIF. However, since the start of NIF science campaigns in 2009, the overwhelming source of feature development pressure has come not from evolving the NIF laser but from ongoing development and evolution of the target chamber diagnostic systems employed to capture x-ray, neutron, and optical radiation generated during NIF experiments. These "target diagnostic" systems carry, individually, a tiny fraction of the complexity of the NIF laser, but they evolve at a much faster pace as physicists invent novel approaches for extracting evermore useful data from NIF experiments.

During the DOE-mandated 120 Day Study, completed in 2014 and conducted to identify changes in NIF operations necessary to significantly increase the shot rate, one of the findings was the need to make experiment configuration faster and simpler for experimentalists, i.e., simplify CMT. Even prior to that study, an increased rate of target diagnostic development coupled with staff changes in the Shot Configuration project that maintains CMT had brought into sharp focus the challenges in continuing with the existing architecture and development lifecycle.

DECIDING ON THE PATH FORWARD

In theory, the overall requirement of simplifying CMT was straightforward; just make the tool easier to use by the user community and make it quicker. But before the team could begin to make such a transition, they had to assess what other high-level requirements could be addressed as part of this change.

Interviewing the key stakeholders of the User Office, NIF operations, experimentalists and Control Systems software developers and by talking to the SST developers themselves added eight more high level requirements that the team was to develop to (see table 1).

Table 1: Shot Setup Tool High Level Requirements

Requirements
Ensure data is consistent with other User Tools.
Use rule sets to set up an experiment.
Integrate with the facility configuration management system.
Provide integrated access controls.
Employ a data group-centric setup.
Support non-contiguous experiment setup.
Maintain interfaces to external systems.
Be easier to maintain and evolve.
Do not interrupt current NIF operations

With the 120 Day Study providing a programmatic mandate, the door was open to update or replace CMT, and after an extensive requirements-gathering phase and internal architecture review, the Shot Configuration team determined that modifying CMT to meet current programmatic needs was not a viable solution.

At this point, the obvious issue confronting the team was how to go about re-implementing the one million lines of code currently used by CMT in an entirely new tool and adhere to the ninth high level requirement of not disrupting NIF operations. For many reasons, a single, big bang deployment of a replacement application would be extremely risky. It would be difficult to create an accurate plan that would estimate when the tool would be complete, user needs would likely change over the duration of the development and in reality, there would be a lot of new bugs that would need to be addressed as users found them.

To be successful, the development effort would need to be broken into smaller more manageable pieces, leading the team to decide to base the new tool around the Data Group centric requirement.

DATA GROUP CENTRIC DESIGN

When a CMT experiment is exported for use by the control system, it is formatted as a single XML document in a hierarchical tree structure and with related pieces of data co-located in the relevant nodes. For example, Target Diagnostics (TDs) are defined in a single branch and each individual TD as sub branch within it as seen in Figure 1.


```
▼<Target_Diagnostics>
  ▶<SXI location="TC161-326">...</SXI>
  ▶<SXI location="TC018-123">...</SXI>
  ▶<DIM location="TC090-078">...</DIM>
  ▶<DIM location="TC090-315">...</DIM>
  ▶<Dante location="TC143-274">...</Dante>
  ▶<Dante location="TC064-350">...</Dante>
  ▶<EMP location="TC102-084">...</EMP>
  ▶<FABS location="Q31B">...</FABS>
  ▶<FABS location="Q36B">...</FABS>
  ▶<GRH-6M location="TC064-020">...</GRH-6M>
  ▶<NAD-20M location="TC116-316">...</NAD-20M>
  ▶<NAD-Flange location="TC007-000">...</NAD-Flange>
  ▶<NAD-Well location="TC064-241">...</NAD-Well>
  ▶<NBI location="Q31B">...</NBI>
  ▶<NBI location="Q33B">...</NBI>
```

Figure 1: Target Diagnostic hierarchical structure.

To achieve the goal of creating many smaller development efforts, the designers simply took the current export and broke it up into many distinct groups of related data. These Data Groups were then prioritized by assessing the state of the current support for the data group, the complexity of the setup for the users to create that data group, the time since that major update to the data group and finally the complexity of the requirements of data group itself. Table 2 was the result of this analysis.

Table 2: Data Group Development Priority

Data Group	Priority
Target Diagnostics without setup support	Highest
Target Diagnostics with setup support	
Target Specification	
Main Laser Specification	Lowest

The key project design choice taken at this time was the decision to migrate data groups incrementally from CMT to SST while both tools were still in operational use. There are many benefits from taking this approach; namely:

- It gives the users the opportunity to provide feedback much earlier and more regularly in the overall development lifecycle rather than at the end of the overall development effort, and it facilitates introduction of new requirements in rolling releases.
- Significantly reduces the training burden that comes with the development of a new tool as the amount of change to the user interface is limited by each release.
- Incremental delivery also helps bound performance problems as they arise and answers the question "what changed in the release?".
- Data groups can be developed in parallel more easily and aids developer cross training.
- Follows AGILE development principles of delivering small releases more frequently.
- The data groups also make it easier to define a simple ruleset based setup aka express setup as the scope is limited. This limits the numbers of stakeholders that need to be consulted to mine requirements.

The most obvious drawback is that the users must use two tools in parallel to create their experiments. While this is inconvenient for the users, the team thought that the benefits of the approach outweighed the drawbacks, and with the support of the User Office, the team took time to explain the approach to the user community through working group meetings and the NIF User Fora.

Once approval for the development of the new tool was granted, the first release was planned with the goal of implementing two target diagnostics; one that was very simple and one for which an "express setup" concept had already been defined. The express setup is a new functional element planned for SST data groups that is intended to enable non-subject matter experts to configure complex diagnostic devices requiring only their knowledge of the overall physics goals of a particular experiment.

DATA SERVER UPDATE

The CMT data server is responsible for generating the experiment XML export to the control system and to the other tools in the campaign management suite. It also performs data validation to ensure that the experimental setup can be executed within the safe operating parameters of the laser.

The data group centric approach allows the team to reuse most of this code as part of the new implementation. However, one significant update was required, the creation of a merge service.

To migrate a data group from CMT to SST, the user interface for that data group is removed from the CMT and reimplemented in SST. Once this is complete, the XML for that data group is no longer generated by CMT but will be created by SST. When an experiment export is requested by a user, both tools create XML files; CMT creates an
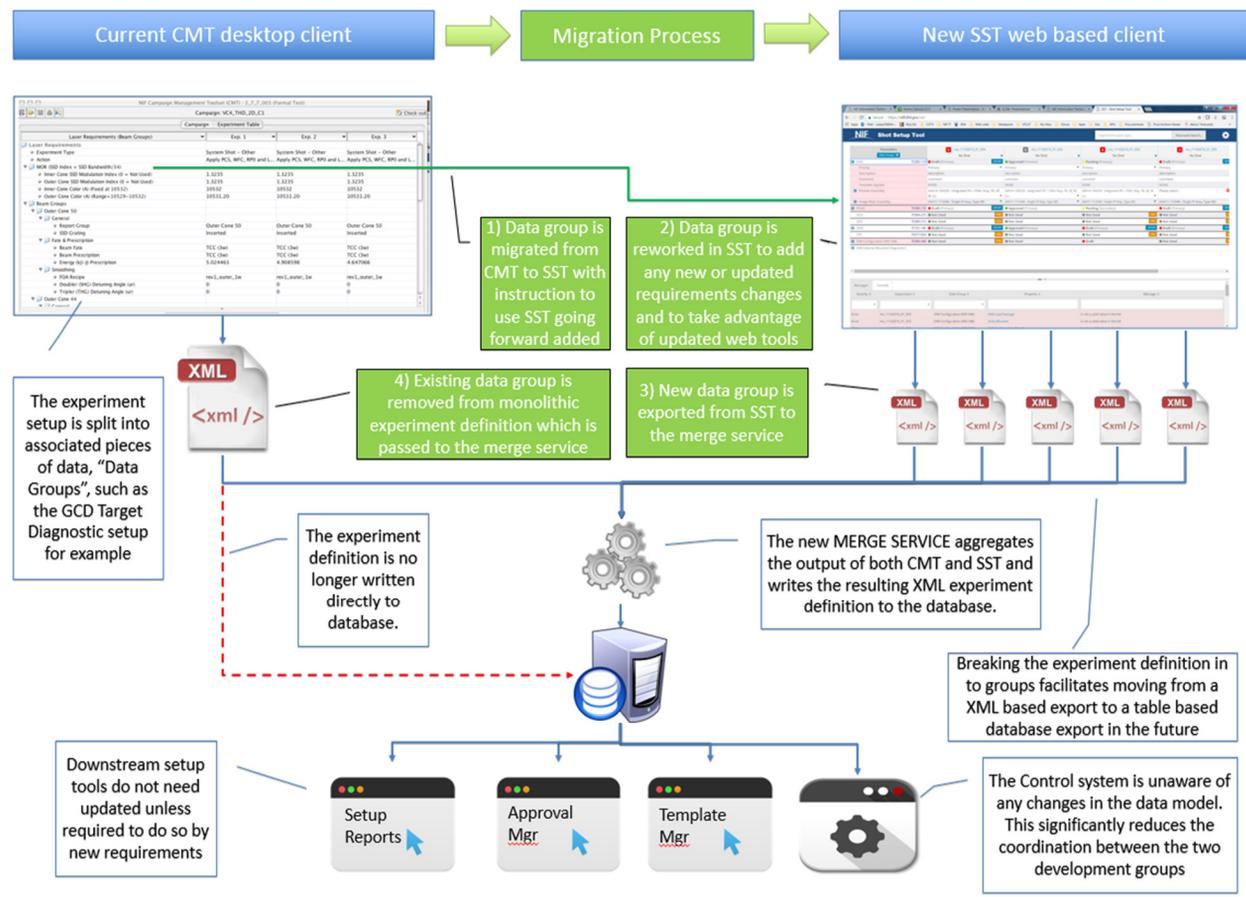
Figure 2: The Data Group migration process from the CMT tool to the SST tool

XML document the data groups it manages and SST generates XML documents for each of the data groups defined in it. A merge service takes these various XML documents
and makes a single XML document that is identical to the original created by CMT. This allows outside applications with dependencies on the monolithic experiment XML document to continue to function with few or no changes.

Testing and verification of the output from the merge service is straightforward as it is nothing more than a comparison of the experiment export before the migration to the experiment export after the migration. If no new export data was requested during the migration of the data group, the two files should be the same.

TECHNOLOGY UPDATES

The current tool was architected utilizing a thick client based on the no longer supported Oracle JNLP technology, and the UI was built on a custom framework. Moving off these technologies was a key driver in the project and so the opportunity was taken to refresh the user interface technology stack using web based technologies at the same time.

Before development began, research was carried out to assess technology trends and developer skill sets to make sure that any chosen technologies would not be obsolete as soon as we used them. For example, discussions were held with LLNL's other software development groups to check the direction that they were taking in terms of web based technologies. Eventually, the team decided to use TypeScript[1], jQuery[2], and Angular [3] for the user interface implementation.

From a project standpoint moving to these web based technologies realized many benefits. The most important benefit for the longevity of the tool, was for the developers. Using Angular and jQuery components provided an opportunity for developers to refresh their skill sets which in turn aids retention and motivation. If it becomes necessary to hire new developers, use of these technologies will make it easier to find and attract personnel rather than trying to find developers to work on a bespoke, out of date technology.

An example of the problem faced can be seen with a simple web search. No results are returned for the CMT property model framework and there are limited results and examples for JNLP. However, there are many sites that can be used for coding assistance when searching for help with the modern technology stack. This simple change can really boost developer productivity as they can focus on the application rather than trying to answer the question "How do I…?".
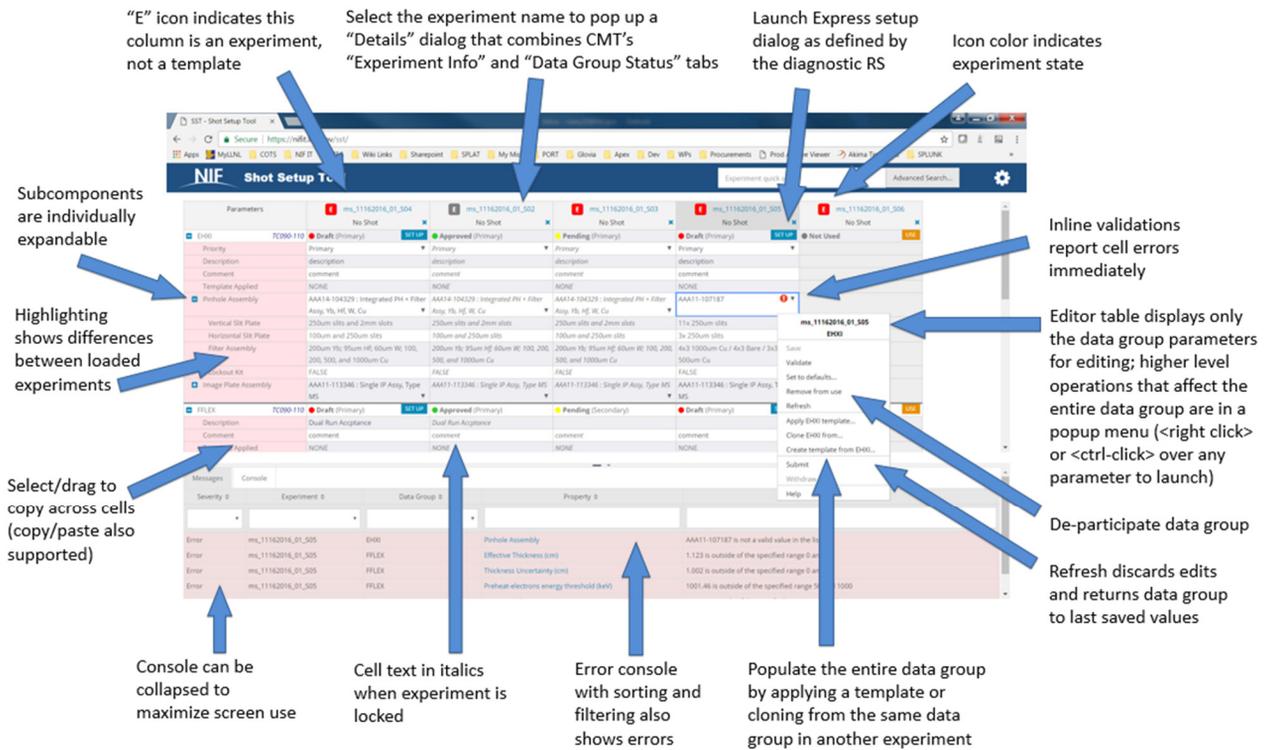
Figure 3: Some of the user interface features of the SST tool

Finally, by creating a clean interface to the new components using the Model-View-Controller pattern, updates can be added without significant changes to the underlying data model. This will facilitate the on-going maintenance of the tool over the coming years.

In terms of the user interface itself, the update changed the user experience from something that looked like a simple tree structure editor to a tool that resembles many of the web sites used today. Figure 3 shows some of the features of the new interface

The interface has been designed around context based menus associated with the data groups and what the user can to do with that data group. Once the user is familiar with the concept, learning how to use the tool is greatly simplified; there is no need to remember where menu options are and context menus help guide the user through the setup process.

SUCCESSFUL DELIVERY

The first release deployed to production in January 2017 supporting the FFLEX and EXHI diagnostics. Since then, four other major releases have been delivered providing support for over twenty diagnostics.

User feedback is that the tool is easier to use through the context menus and express setup and that the new tool is significantly faster than the original setup tool. The users can manage the two-tool model but understandably, want the migration to a single tool to complete as soon as possible.

CONCLUSION

Referring back to table 1 and the high-level requirements set for the team, the Data Group centric design of the new tool helps to address several of the other requirements at the same time. Namely:

- Use rule sets to set up an experiment.
- Maintain interfaces to external systems.
- Be easier to maintain and evolve.
- Do not interrupt current NIF operations

In doing so and by using the data model as a significant element in the design process, a potentially long and risky migration was avoided whilst allowing the team to deliver a functional product for both users and the sponsor as quickly as possible

ACKNOWLEDGEMENT

REFERENCES

[1] TypeScript website, *https://www.typescriptlang.org/*
[2] jQuery website, *https://jquery.com/*
[3] Angular website, *https://angularjs.org/*
[4] React website, *https://facebook.github.io/react/*