ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-MOPHA015

REVERSE ENGINEERING THE AMPLIFIER SLAB TOOL AT THE NATIONAL IGNITION FACILITY

A. Bhasker, R. D. Clark, J. E. Dorham
Lawrence Livermore National Laboratory, Livermore, USA

Abstract

This paper discusses the challenges and steps required to convert a stand-alone legacy Microsoft Access-based application, in the absence of original requirements, to a web-based application with an Oracle back-end and Oracle Application Express/JavaScript/JQuery front-end. The Amplifier Slab Selection Tool (ASL) provides a means to manage and track the amplifier slabs on National Ignition Facility (NIF) beamlines. ASL generates simulations and parameter visualization charts of seated Amplifier Slabs as well as available replacement candidates to help optics designers make beamline configuration decisions. The migration process, undertaken by the NIF Shot Data Systems (SDS) team at Lawrence Livermore National Laboratory (LLNL), included reverse-engineering requirements from an end-user perspective, identifying obsolete requirements, and identifying new requirements due to evolving processes and changing NIF usage patterns.

INTRODUCTION

The National Ignition Facility (NIF) at Lawrence Livermore National Laboratory (LLNL) is composed of 192 laser beamlines, that converge on a hydrogen-filled target the size of a peppercorn to cause nuclear fusion. Each of these 192 beamlines contains two large amplifier sections, the main amplifier and the power amplifier. These amplifiers consist of precisely positioned neodymium-doped phosphate glass slabs and provide 99.99% of NIF's power and energy [1]. There are 3072 slabs in use across all 192 beamlines. These slabs are expected to last through years of daily use but do suffer damage over time. After a decade of operations, and over 2700 shots fired, amplifier slabs are being replaced due to wear and tear at the greatest rate since NIF experimental operations began.

Replacing an amplifier slab on a NIF beamline requires a number of criteria to be satisfied, which calls for interplay between optics designers and component engineers. The Amplifier Slab Selection Tool serves as a single place for all of this – ranging from identifying available candidates to verifying the candidacy of an available amplifier slab for a given position by performing complex optical calculations for a given beam path.

THE LEGACY APPLICATION

The original ASL was developed when NIF was under construction, over a decade ago. The original tool was built using MS Access. Its front-end was comprised of forms and reports for component engineers and optics designers-serving as an optics tracking tool for the former, and a simulation environment for amplifier slab configurations on NIF beamlines for the latter. It saved the beamline configuration of amplifier slabs in an MS Access Database backend. This standalone MS Access tool was important for getting NIF built. However, it did not evolve with NIF. This was largely because amplifier slabs did not demand significant maintenance for many years, so the standalone tool did not require any enhancements during that time.

THE TARGET APPLICATION

One of the major applications used by NIF Engineers and Scientists is a tool suite called Production Optics Reporting and Tracking (PORT), which has been developed using Oracle Application Express (APEX). Oracle APEX is a web-application development tool for the Oracle databases [2], that supports utilization of JavaScript and JQuery components as well.

The PORT tool covers a wide domain of capabilities – ranging from reporting and analysis to task-scheduling and optics installation, as well as optics and targets tracking. For the project described in this paper, the legacy MS Access Amplifier Slab tool was reverse engineered, revamped and given a new home as a component of PORT.

PORT was implemented in APEX version 5.0 at the time of this undertaking.

METHODOLOGY

The basic methodology used to convert this legacy application was to reverse-engineer, re-design and re-implement the application. The aim was to have a new application that was functionally equivalent to the legacy application but redesigned in a way to allow the team to fully leverage the power and features of Oracle APEX.

Since the low-level requirements needed to be extracted, all the exploratory reverse engineering work was performed in a development environment that was a replica of the production environment. The process began with extraction of the low-level design details from the source code, and then extracting the high-level design information from that. This information collected drove the design and implementation of the new system using the target technology. The conversion of this tool from a legacy MS Access application to an Oracle APEX application involved the following steps (Fig. 1):

Identifying All the Legacy Functionality

The initial goal was to gain a broad understanding of what features this tool provided to users. In the absence of the original requirements, key application users were asked for their input to understand usage patterns for the legacy tool. Steps were also taken to gain a preliminary understanding of how data from other NIF systems, pertaining to

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-MOPHA015

available replacement amplifier slab candidates was made available to the standalone MS Access tool.

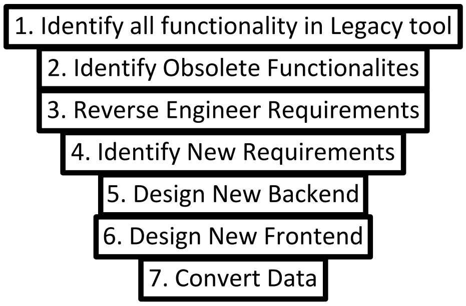


Figure 1: Methodology followed.

Identifying Obsolete Functionalities

As an extension of identifying all the features offered by the legacy tool, the team further identified features that were no longer needed – a likely occurrence in a research environment, where project needs and scopes are constantly evolving. The Amplifier Slab tool both tracks available amplifier slabs and helps optics designers simulate amplifier slab configurations. The team was able to identify some information reported by the legacy tool that was no longer required. For example, the optics inventory storage standards evolved over time. The location of available Amplifier slabs used to be identified by a bin number, but that is no longer needed by the users. Similarly, the legacy tool offered a complex Auto-Selection functionality, that had been of great value when NIF was being designed. It would build a configuration of 16 amplifier slabs for a beamline, based on an intricate set of rules. Today, however, no more than one amplifier slab needs to be replaced on a beamline at a time. Thus, the legacy requirement to build completely new configurations on a beamline could be deprecated.

Reverse Engineering Requirements

The goal at this step was to perform a deep-dive analysis to uncover the design requirements of the legacy tool. MS Access Applications allow for creation of Tables, Queries, Forms, Reports and Macros, from which the team needed to reverse engineer the original requirements by delving deep into the source code. Based on the identified functionalities, a two-pass plan was implemented to (1) identify the backend database components corresponding to identified application features and reverse-engineer the low-level requirements for them, and (2) to identify the low-level requirements for the application features themselves. The identified backend database requirements were then used to reconstruct high-level requirements, which were then used to design the new backend database. The low-level frontend requirements were similarly used to construct high-level frontend requirements. The frontend requirements that were identified can be broadly classified in following two categories (Fig. 2):

- **Functional Requirements:** These were the requirements that were reverse engineered by meticulously going through source code. This involved deep-diving into the MS Access source code and building a list of use-cases. The functional requirements gathered during this process included *technical definitions* of content displayed, *navigational structure* and *access control* constraints.
- **User Interface (UI) Requirements:** These requirements had more to do with reverse engineering the UI design decisions implemented in the legacy tool that needed to be carried forward into the new tool. This essentially involved identifying the application entities and actions associated with them, which arguably made identifying this kind of requirement more challenging, in the absence of the original requirements. These UI requirements could be either *dynamic* or *static* application implementations. An example of a static UI requirement is color-coding the background color of the amplifier slab serial number by slab vendor, allowing the user a quick visual of vendor distribution across a beamline. An example of a dynamic UI requirement could be any scenario that involves a user action – e.g. a mouse-click on region that opens a pop-up window.

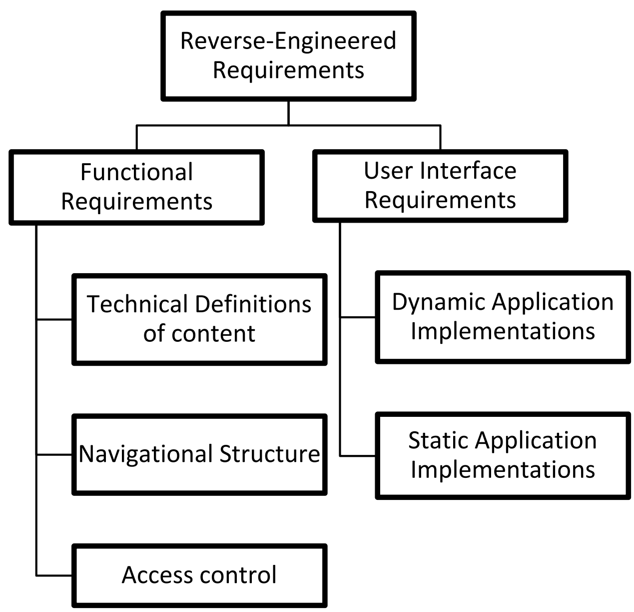


Figure 2: Types of reverse-engineered requirements in the project.

Identifying New Requirements

Once the requirements had been reverse-engineered from the legacy application, and data flow recorded, some new requirements were added into the system. These new requirements included valuable new functionality to the application that the legacy system did not have as well as requirements pertaining to the new implementation environment. The new requirements that were identified can be classified as follows (Fig. 3):

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-MOPHA015

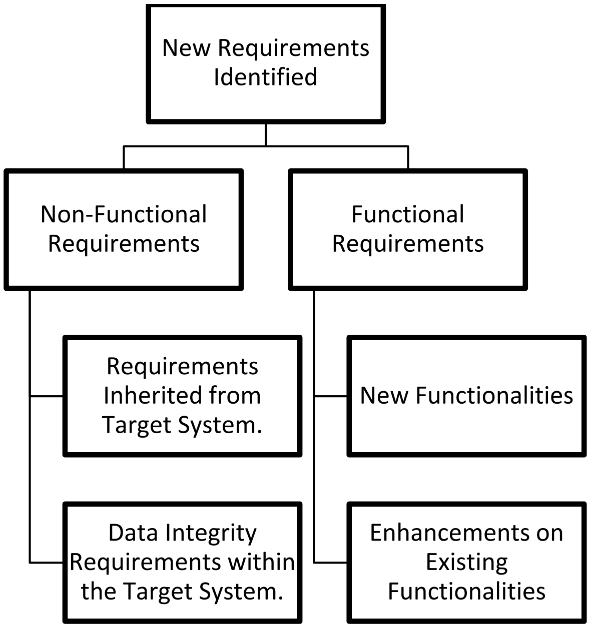


Figure 3: Types of new requirements identified in the project.

- **Non-Functional Requirements:** These included requirements pertaining to platform compatibility (Oracle APEX, web browsers etc), user authorization within the new target environment, and maintenance of data integrity by use of referential constraints. Since this project involved reimplementing a legacy application within an existing system, several non-functional requirements, such as development platform and security were inherited from the target system. Referential constraints were also identified and set up to ensure a user could only add an amplifier slab that existed in the overarching data system.
- **Functional Requirements:** These include requirements that represent new features or functionalities. These ranged from turning some report columns into hyperlinks directing users to more detailed information, to newly designed webpages giving users visibility into the difference in their simulation and the actual NIF configuration. Other new functional requirements included an added ability to refresh parameter data for the amplifier slabs in real-time, permitting more precise calculations when selecting a replacement amplifier slab and setting constraints to ensure that no more than 16 amplifier slabs are placed on a NIF beamline.

These new requirements, along with the reverse-engineered requirements helped piece together an abstract design for the back-end and front-end of the reimplemented application.

Designing the New Back-end

The new application's back-end was redesigned to be implemented in Oracle 12c, under the PORT schema. The back-end requirements gathered resulted in a more compact database redesign, reducing the number of tables from 100 to 8. The major contributing factors to this drastic reduction in the number of tables were:

- The obsoleted requirements meant that several tables could be deprecated.
- The technical capability offered by Oracle APEX to use APEX_COLLECTION, which is discussed in the next subsection.
- Import tables were no longer required to hold source data, since this application now resided with the source data schema and could be directly queried.

Design decisions were also made to have data formats match the target system (PORT), which also houses the main data sources for the application, rather than matching the legacy tool's data formats, in an effort to make a seamless transition. Audit tables were new requirements introduced into the system, that were incorporated to keep track of changes made in the simulation environment.

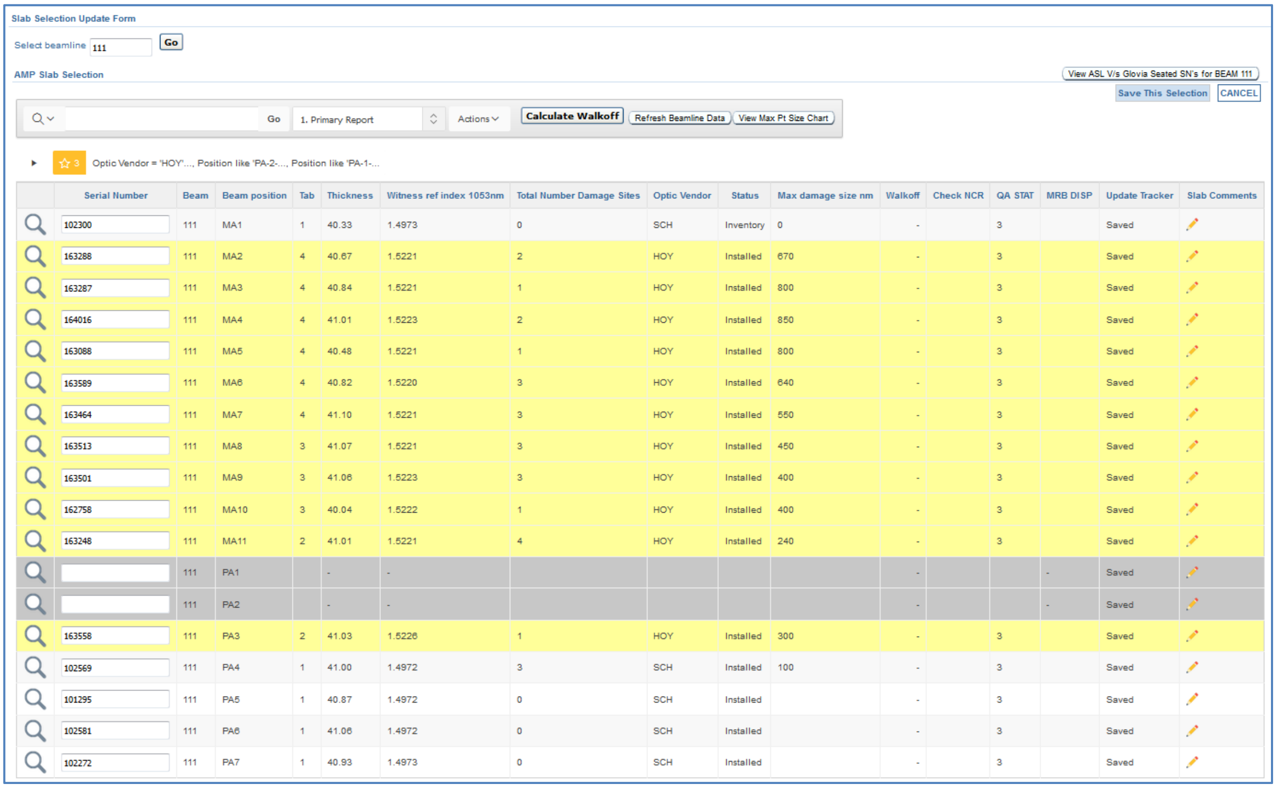
Designing the New Front-end

The new application's front-end was redesigned as a part of the PORT tool suite (Fig. 4), and thus able to leverage Oracle APEX tools. In particular, the team was able to leverage APEX_COLLECTION to implement the bulk-load of the functionality offered by the legacy tool. APEX_COLLECTION are Oracle APEX constructs that enable the temporary capture of one or more non-scalar values [3]. It allows storage of rows and columns currently in the session state, so that they can be accessed, manipulated, or processed during a user's specific session [3]. Given the nature of the application as a simulation environment for NIF Engineers and Scientists to experiment with different configurations of amplifier slabs on NIF beamlines, this feature was heavily leveraged – allowing users to try several amplifier configurations before deciding to save one to the Oracle database tables. This contrasted with approximately 20 temporary tables used by the MS Access application to achieve the same effect. The new application also leveraged Oracle APEX's modal dialog pages to enhance the user experience when looking for available replacement amplifier slab candidates. A modal dialog is an overlay window positioned within the same browser window, that remains active and focused until the user closes it [4]. In contrast with the legacy application's functionality offering a small list of replacement options to the user, allowing them to copy the selection to the clipboard, and then pasting it in the desired location on the simulation page, the new tool offers a list of options of candidate amplifier slabs that, when selected by the user, would autopopulate in the simulation tool.

Data Conversion and Population

The database conversion tool used to convert database constructs in the MS Access database to Oracle 12c was FullConvert v7 [5]. This allowed for automatic, direct conversion with minimal user inputs. This converted data was then used as seed data for the newly implemented tables, which could be further refreshed with the latest parameter data.

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-MOPHA015



Serial Number	Beam	Beam position	Tab	Thickness	Witness ref index 1053nm	Total Number Damage Sites	Optic Vendor	Status	Max damage size nm	Walkoff	Check NCR	QA STAT	MRB DISP	Update Tracker	Slab Comments
102300	111	MA1	1	40.33	1.4973	0	SCH	Inventory	0	-		3		Saved	
163288	111	MA2	4	40.67	1.5221	2	HOY	Installed	670	-		3		Saved	
163287	111	MA3	4	40.84	1.5221	1	HOY	Installed	800	-		3		Saved	
164016	111	MA4	4	41.01	1.5223	2	HOY	Installed	850	-		3		Saved	
163088	111	MA5	4	40.48	1.5221	1	HOY	Installed	800	-		3		Saved	
163589	111	MA6	4	40.82	1.5220	3	HOY	Installed	640	-		3		Saved	
163464	111	MA7	4	41.10	1.5221	3	HOY	Installed	550	-		3		Saved	
163513	111	MA8	3	41.07	1.5221	3	HOY	Installed	450	-		3		Saved	
163501	111	MA9	3	41.06	1.5223	3	HOY	Installed	400	-		3		Saved	
162758	111	MA10	3	40.04	1.5222	1	HOY	Installed	400	-		3		Saved	
163248	111	MA11	2	41.01	1.5221	4	HOY	Installed	240	-		3		Saved	
	111	PA1		-	-					-			-	Saved	
	111	PA2		-	-					-			-	Saved	
163558	111	PA3	2	41.03	1.5226	1	HOY	Installed	300	-		3		Saved	
102569	111	PA4	1	41.00	1.4972	3	SCH	Installed	100	-		3		Saved	
101295	111	PA5	1	40.87	1.4972	0	SCH	Installed		-		3		Saved	
102581	111	PA6	1	41.06	1.4972	0	SCH	Installed		-		3		Saved	
102272	111	PA7	1	40.93	1.4973	0	SCH	Installed		-		3		Saved	

Figure 4: The Amplifier Slab selection tool on Oracle APEX. For a user-selected NIF Beamline, it shows the Serial Numbers of the Amplifier Slabs installed, along with relevant parameter

CURRENT STATUS AND FUTURE WORK

The conversion project of the Amplifier Slab tool from MS Access to Oracle APEX was one of a series of such conversions that have all been successfully completed. The converted application is now actively maintained by the PORT team and is being prepared to undergo an upgrade from Oracle APEX 5.0 to Oracle APEX 19 at the time of this writing.

ACKNOWLEDGEMENT

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344, release number LLNL-CONF-790279.

REFERENCES

[1] National Ignition Facility & Photon Science, `https://lasers.llnl.gov`

[2] What Is Application Express?, `www.oracle.com/database/technologies/appdev/apex-what-is.html`

[3] Application Express API Reference, `https://docs.oracle.com/database/121/AEAPI/apex_collection.htm#AEAPI531`

[4] Application Express App Builder User's Guide, `https://docs.oracle.com/database/apex-18.1/HTMDB/creating-dialog-pages.htm#HTMDB30095`

[5] Full Convert Overview, `https://www.spectralcore.com/fullconvert`