Flexible Data Driven Experimental Data Analysis at the National Ignition Facility

A. Casey, R. Bettenhausen, E. Bond, R. Fallejo, M. Hutton, J. Liebman, A. Marsh, T. Pannell, S. Reisdorf, A. Warrick
Lawrence Livermore National Laboratory, 7000 East Avenue, Livermore, CA 94550, USA

Abstract

After each target shot at the National Ignition Facility (NIF), scientists require data analysis within 30 minutes from ~50 diagnostic instrument systems. To meet this goal, NIF engineers created the Shot Data Analysis (SDA) Engine that uses the Oracle Business Process Execution Language (BPEL) platform to configure analyses and archive results. While this provided for a very powerful and flexible analysis product, it still required software developers to create each unique analysis configuration executed by the SDA engine. As more and more diagnostics were developed and the demand for analysis increased, the development team was not able to keep pace with the rate of change. To solve this problem, the Data Systems team took the approach of creating a data-driven framework that allows users to specify the analysis configuration (analysis routine, inputs and outputs), input data sources, and results archive destinations as data that is stored in the database. The creation of this Data Driven Engine (DDE) has decreased the manpower required to integrate new analysis and has simplified maintenance of existing configurations. The architecture and functionality of the Data Driven Engine will be presented along with examples.

Data Mapper Technologies

BPEL
- BPEL COTS product
- Web Services
- XML

The original implementation used a COTS workflow engine that executes XML-based logic.

Data Driven Engine
- Java/JDBC
- Web Services
- XML
- Excel Spreadsheets

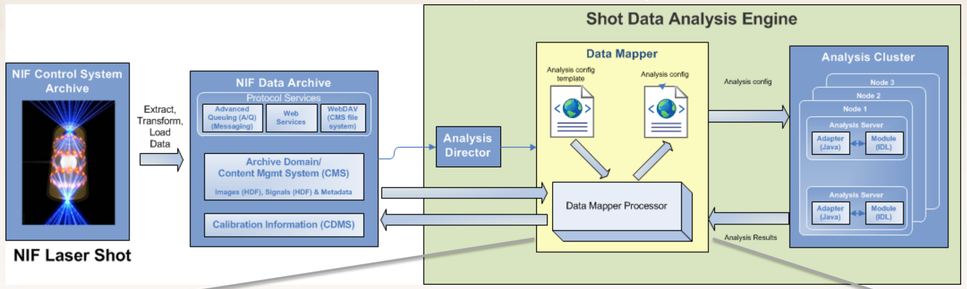
Next, we built a Java framework that included reusable data mapping functions and error handling. Analysts specify all data mapping in Excel.

New Data Driven Engine
- Java
- Web Services
- XML
- WebDAV
- Groovy/Velocity Macros
- Struts/JQuery/HTML

Last, we added user-defined Macros for complex data mapping logic, and provided a web interface to the Analysis Interface Specification from which we extract data map requirements.

The Shot Data Analysis Engine consists of an ***Analysis Director*** that sequences the analysis for each diagnostic; a ***Data Mapper*** that maps data from data sources (Archive, Calibration, NIF Configuration) to analysis, and maps results to the archive; and a Cluster of ***Analysis Servers*** that execute the analysis routines. We migrated the Data Mapper from a COTS Workflow product to an in-house developed data-driven application that significantly reduces the time to integrate new analyses.

Shot Data Analysis Processing



Data Mapper Evolution

BPEL
Business Process Execution Language

Pros:
- COTS Workflow Tool
- Immediate development

Cons:
- Not suited for volume spikes
- Requires special developer training
- Requires special system administration
- Inefficient re-use model:
  - based on re-usable processes
  - Introduces communication overhead that can cause timeout failures
- Lack of data mapping transparency

DDE
Data Driven Engine

Pros:
- Data Driven
- Impervious to volume spikes
- Framework provides re-use and data mapping functions
- No special system administration
- No special language
  - analysis developers specify the data mapping in Excel spreadsheets
- Transparent data mapping

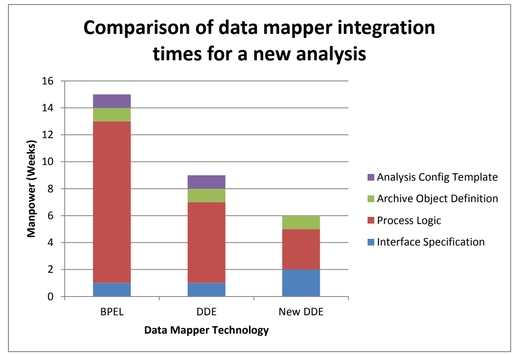
Cons:
- Lack of programming constructs can result in complex data specification
- Required time to develop the tool

New DDE
Data Driven Engine

Pros:
- Includes Macro ability for complex logic
- Provides a web interface to specify data mapping
- Interface mimics the Interface Document currently used by analysis developers

Cons:
- Required time to develop the tool

Analysis Integration Time



Summary

In the field of SW development, the general strategy is to use COTS products wherever possible in order to minimize local development effort and to maximize the capabilities and experience of another development team.

However, there are times when the replacement of COTS products with custom software yields significant benefits in terms of tailored functionality that fully meets the user needs and makes better use of development dollars.

In migrating from BPEL to the DDE, the Analysis team at NIF achieved:
1. ***more efficient re-use*** of existing capabilities and functions;
2. ***simpler, user-specified*** data mapper configurations;
3. ***increased transparency and maintainability*** of data mapper configurations;
4. load balancing that ***handles peak loads predictably and reliably***;
5. ***less manpower*** to add a new analysis to the Shot Data Analysis Engine;
6. ***fewer dedicated, specialized software developers***.

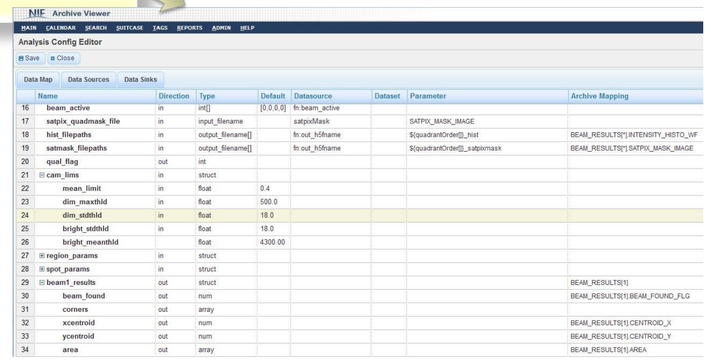
With the new DDE, the team is expecting to be able to achieve:
1. ***greater simplification and maintainability*** of data mapper configurations;
2. ***additional decrease in manpower*** needed to integrate a new analysis.

Sample BPEL XML Specification:

```
<switch id="BpSwt1" name="Switch_Diag">
-<case condition="starts-with(bpws:getVariableData('origDataTaxon'), 'CONFIG')">
-<bpelx:annotation>
<bpelx:pattern patternName="case">
Got Config Taxon</bpelx:pattern>
</bpelx:annotation>
-<sequence id="BpSeq3" name="Sequence_InputIsConfigData">
-<assign id="BpAss9" name="ParseAgain">
-<copy>
<from expression="substring-after(bpws:getVariableData('parsedTaxon'), '|')"/>
<to variable="parsedTaxon"/>
</copy>
</assign>
```

Sample DDE Specification:

FROM_DATASOURCE	FROM_PARAM	TO_PATH
setup_parameters	TARGET_VIEW_ANGLE	dante_setup.target_viewangle
setup_parameters	TARGET_LEH_DIAMETER	dante_setup.LEH_diameter
setup_parameters	CHAN_COMP_APPLY_ALIGN	chan_comp.apply_align
fixed_params	TCC_DISTANCE	dante_setup.TCC_distance
fixed_params	CHAN_COMP_FALLPEAK_RATIO	chan_comp.fall_peak_ratio
fixed_params	CHAN_COMP_DWNSAMP_TSTEP	chan_comp.downsamp_time_step
fixed_params	NUM_ITER	blackbody.num_iter
fixed_params	BLACK_BODY_START_TEMP	blackbody.start_temp
fixed_params	VOLT_RATIO	blackbody.voltratio_value
fixed_params	VOLT_RATIO_APPLY_CORR	blackbody.voltratio_applycorr
fixed_params	LOW_ENERGY_END_RANGE	energy_range.lowenergy_end_range
fixed_params	LBAND_START_EV	energy_range.lband_start_ev
fixed_params	LBAND_END_EV	energy_range.lband_end_ev
fixed_params	LBAND_START_RANGE	energy_range.lband_start_range



LLNL-POST-644238

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.