



Sandia
National
Laboratories

Exceptional service in the national interest

Safeguards Model Improvement and Archiving

Benjamin Cipiti¹, Philip Honnold¹, Pat Moosir¹,
Oliver Paleen¹, Nathan Shoman¹

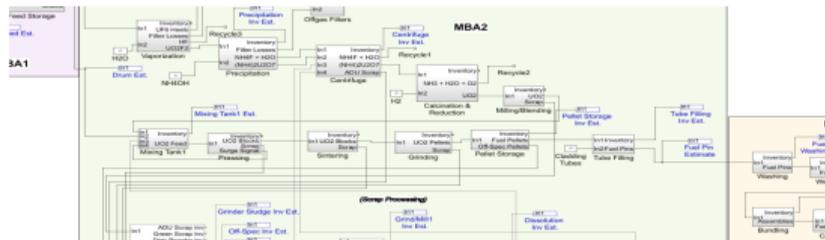
¹Sandia National Laboratories

June 26, 2023

SNL FY23 work package overview: Reprocessing Modeling Support



- Work package consists of several key thrusts
 - SSPM model and flowsheet archiving
 - SSPM dev toolkit
 - MAPIT development
- Overarching theme is to **improve access to mod/sim tools to support safeguards**
- Focus on open source availability and improving documentation



MAPIT



**SSPM model and
flowsheet archiving**

**Team: Pat, Philip,
Oliver**

SSPM developed and used for over 15 years

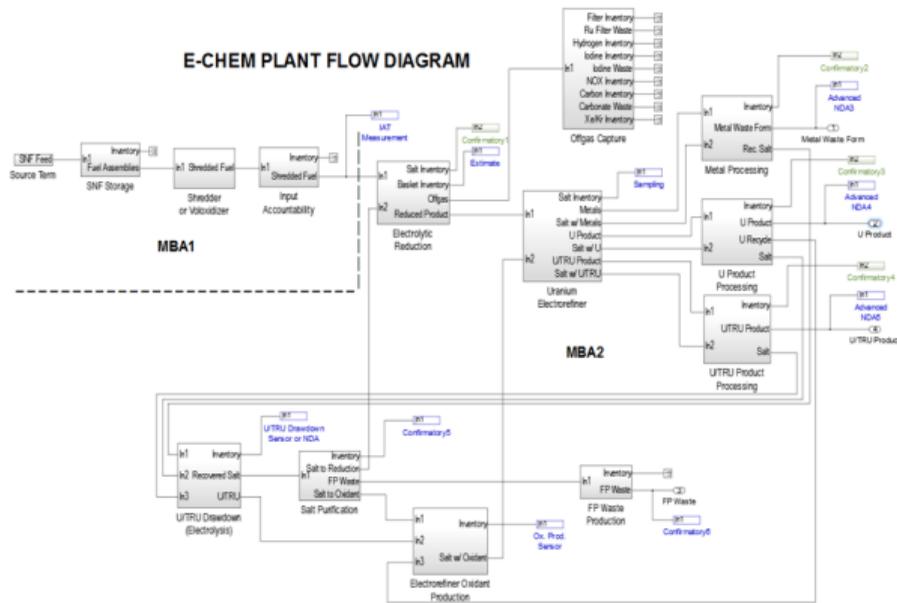


- Nuclear fuel cycle facilities face significant economic challenges in meeting safeguards, security, and environmental regulations
- Future facilities need affordable and accurate material accountancy and plant monitoring systems for both domestic and international safeguards
- SSPM was created to examine the impact of different measurement technologies on traditional safeguards metrics
 - Originally started as internal Sandia LDRD circa 2008 (*Advancing the State of the Art in Materials Accountancy through Safeguards Performance Modeling, SAND2008-5100*)
 - Application has expanded to a variety of safeguards tasks
- A variety of sponsors, including DOE-NE and NNSA, have funded the development of several SSPM models
 - PUREX, UREX+, Enrichment, Fuel Fabrication, and Pyroprocessing

SSPM is a powerful tool designed to explore challenging safeguards objectives



- The SSPM is a transient mass tracking and safeguards model based in Matlab Simulink.
 - Spent fuel source term library – integration with SCALE
 - Full elemental and isotopic tracking (1675 isotopes)
 - Customizable measurement points
 - Simulation of gamma spectra with GADRAS
 - Integration of process monitoring data



Challenge: Knowledge retention across projects and staff changes



- SSPM results finalized in reports, but specific project aspects are sometimes lost
- New staff often lack the insight of original authors when model was created
- Time lost in some cases when projects are restarted and/or follow-on work is conducted
- Changes to overall model philosophy may not be propagated to all models
 - More on this in the dev toolkit section
- Over reliance on institutional knowledge and deliverables has led to poorer documentation



FY23 progress and outlook

- Started archiving efforts by starting per flowsheet Git repositories
- Interviewing relevant staff with questions regarding flow sheets
- Started collecting relevant technical documents

Project goals

- Sharable and open documentation on flow sheet specifications
- Integration with dev-toolkit (how-to, tutorials, etc)
- Ties to historical SAND reports
- Utilization of Simulink-specific version control tools

Anticipated end of FY status

- Git repositories for all flowsheets
- Finish document collection
- No anticipated issues
- Estimated carryover: 10K

The slide features a central dark blue diamond shape with a white border. Two diagonal lines, one from the top-left to the bottom-right and one from the top-right to the bottom-left, cross at the center. These lines are composed of several colored segments: cyan, purple, orange, green, and red. The text is centered within the diamond.

SSPM dev toolkit

**Team: Pat, Oliver,
Philip, Nathan**

Challenge: SSPM requires government use due to potential sensitivities



- Paperwork to share SSPM even internally to U.S. Citizens at labs can take weeks
- User experience can vary based on flowsheet
- Results outside original model intent not well validated in some cases
 - Largely due to large of out-of-scope benchmarking
- Improvements in newer models might not be propagated to older models with different flowsheets
- Models often difficult to use due to lack of documentation for outside users and focus on SNL-based experts



Developing foundational tools for next-generation safeguards research and development

Goal: Open-source components used to build SSPM-based models

- Individual components could be combined to model flowsheets
- Components could be validated individually and could be monitored for accuracy
- Improved maintainability and accessibility to others in the community
- Currently in initial development stage

Anticipated end of FY status

- Develop libraries for 1-2 components for a fuel fab facility
 - Work required to “template” the process of developing components
- Develop testing suite for components
- Estimated carryover: N/A
 - Note: Project is exploratory and is being proposed as a work package for FY24

The logo for MAPIT features the word "MAPIT" in a white, sans-serif font centered within a dark purple circle. Above and below the text are two horizontal rows of colored dots in shades of teal, orange, and light blue. The logo is set against a dark blue diamond background.

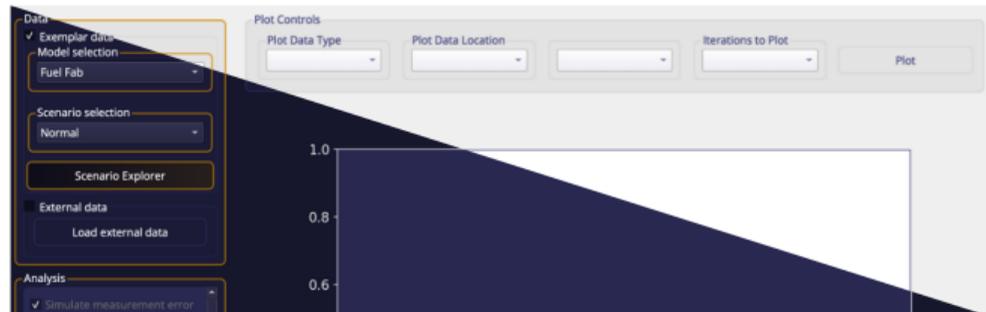
MAPIT

Team: Nathan, Pat



Material Accountancy Performance Indicator Toolkit

- **Goal:** Improve accessibility and transparency for material accountancy
- Written entirely in Python
- Automated error propagation
- Implementation of common statistical tests
 - MUF, σ_{MUF} , SITMUF, Page's trend test
- Robust I/O support
 - *csv*, *mat*, *npz* files
- Intuitive GUI
- Rich API



```
MUF = StatsTests.MUF(inputAppliedError = inputAppliedError,  
                    inventoryAppliedError= inventoryAppliedError,  
                    outputAppliedError = outputAppliedError,  
                    processedInputTimes = processedInputTimes,  
                    processedInventoryTimes = processedInventoryTimes,  
                    processedOutputTimes = processedOutputTimes,  
                    MBP = mbaTime)
```

100.00% | [Progress Bar] | [Elapsed: 00:02 | Remaining: 00:00]

MAPIT: Wider use within community



- Used for in-person bilateral engagement
- Featured in domestic NMCA training to provide visual and interactive exercises
- Showcased at INMM in Vienna
- Input from international users incorporated in latest release

The screenshot shows the MAPIT documentation website. At the top, there is a blue header with the MAPIT logo and a search bar labeled "Search docs". Below the header is a dark grey sidebar with a "CONTENTS:" section. The sidebar lists the following items: "MAPIT documentation home", "Downloading & Installing MAPIT", "Tutorials", "MAPIT Introduction" (which is expanded to show "Fuel fabrication overview", "Walkthrough", "Getting started", "Using your own data (.csv)", and "Using your own data (.mat)"), "Guided exercises", "Theory", and "API home".

MAPIT Introduction

Fuel fabrication overview

The purpose of this tutorial is to introduce you to the included example dataset. This dataset was based on a fuel fabrication facility described by [IAEA-STR150](#). There are some unique features of MAPIT that are only present when using the included datasets.

Some information about the fuel fabrication facility is noted below:

- 300 MT UO_2
- 3.0% weight percent ^{235}U
- Final products are LWR fuel assemblies
- Feed material:
 - Low enriched UF_6
 - Uranyl nitrate
 - UO_2 powder
 - Material from facility scrap recovery

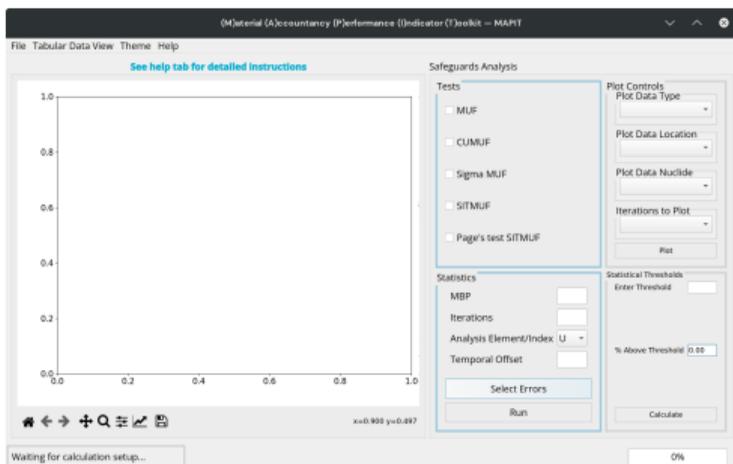
Walkthrough

This tutorial describes basic functions of MAPIT and how to get started using the sample dataset that has been included.

Note

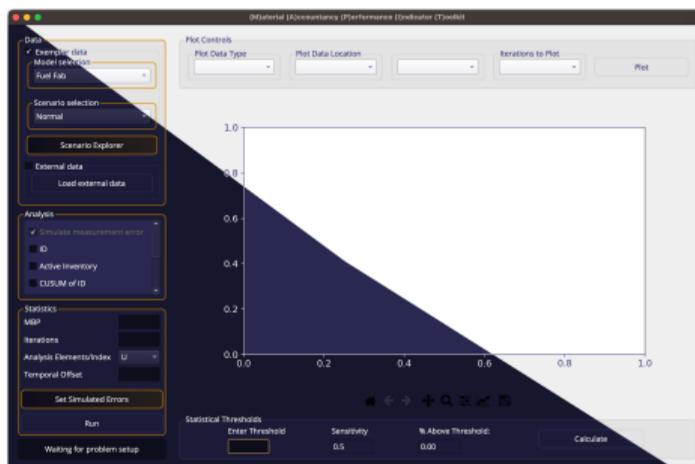
This tutorial contains notes that are not necessary to understand the tool's basic functionality, but may be useful to users looking for more advanced functionality.

FY23 Accomplishment: GUI overhaul



Old design

- Feedback used to overhaul user interface while retaining functionality
 - Consultation with UI expert
 - Non-expert user feedback
 - Input from graphic designers



New design

- More compact layout
- Support for HiDPI screens
- New visualizations
 - SEID contributions
- Ability to adjust font sizes

FY23 Accomplishment: API development



- Developed application program interface (API)
 - Currently based in Python
- Targeted towards advanced workflows and secondary analysis
- All capabilities in GUI available
- End-to-end pipeline shown in example notebook
- Documentation for all key functions available

```
MUF = StatsTests.MUF(inputAppliedError = inputAppliedError,
                    inventoryAppliedError= inventoryAppliedError,
                    outputAppliedError = outputAppliedError,
                    processedInputTimes = processedInputTimes,
                    processedInventoryTimes = processedInventoryTimes,
                    processedOutputTimes = processedOutputTimes,
                    MBP = mbaTime)
```

100.00% [Progress Bar] [Elapsed: 00:02 | Remaining: 00:00]

MAPIT_internal.core.StatsTests.MUF(inputAppliedError, processedInputTimes, inventoryAppliedError, processedInventoryTimes, outputAppliedError, processedOutputTimes, MBP, GUIObject=None, GUIparams=None)

Function to calculate Material Unaccounted For (MUF), which is sometimes also called ID (inventory difference). Specifically calculates the material balance sequence given some input time series.

$$MUF_t = I_t - O_t - (C_t - C_{t-1})$$

I_t is input at time t

O_t is output at t

C_t is inventory at time t (note C is used to denote container to have clearer notation rather than using I with subscripts for both inventory and input)

Important

The lengths and shapes of appliedErrors and processedTimes should be the same. For example:

FY23 Accomplishment: Unit test implementation



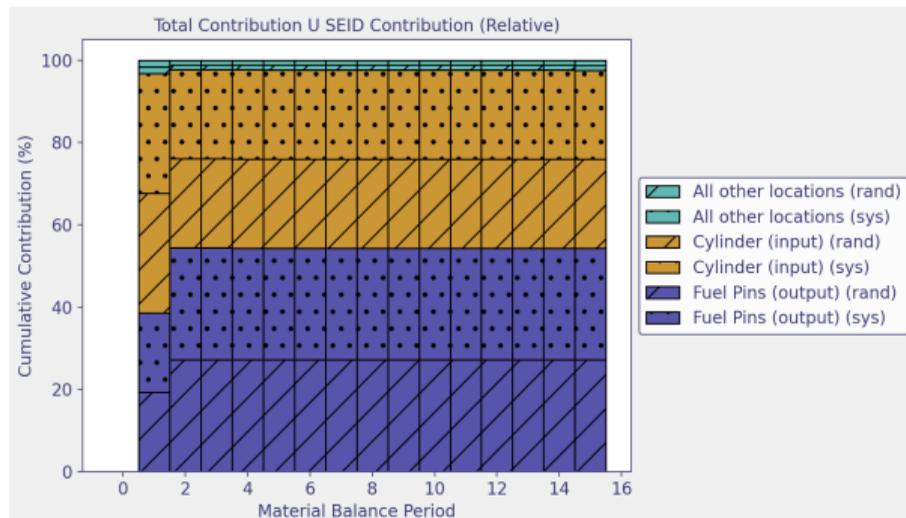
- MAPIT designed in Python, which can run on multiple platforms
 - Challenge: Not all dependency packages available on all platforms
 - Continuous build health tests report weekly status (Windows, Mac, Linux)
- Ensure integrity of mathematical operations
- Tests functionality to ensure code base changes to not impact performance or accuracy

Status	Job	Stage	Name
passed	#15115305 📄 master → d3c73c99 docker centos	test	core.AuxFunctions_unix
passed	#15115304 📄 master → d3c73c99 docker centos	test	core.Preprocessing_unix
passed	#15115303 📄 master → d3c73c99 docker centos	verify_contain...	check_imports_unix
passed	#15115302 📄 master → d3c73c99 docker centos	build_contain...	build_and_push_unixCo...
passed	#15115301 📄 master → d3c73c99 windows shell	win_build	MAPIT_windows_build_...

FY23 Accomplishment: Statistical tests and other improvements



- Added CUMUF tests
- Added capability for domestic or international use (terminology)
- Added visuals for SEID contribution
- Improved input/output capabilities
- Offloaded computation to separate threads to improve GUI responsiveness
- Improved install scripts for users new to Python
- General bug fixes and performance improvements





Additional planned tasks before end of FY

- Progress on high-performance capabilities (multi-threading)
- Full inclusion into NMAC training course
- Initial progress on theory documentation
- Work on improving user guide
- No issues anticipated
- Est carryover: 10K
- Est total carryover (archiving task, dev toolkit task, MAPIT task): 20k

Acknowledgements



This work was funded through the U.S. Department of Energy's Office of Nuclear Energy (DOE-NE).

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.