LA-UR-24-24919

Approved for public release; distribution is unlimited.

Title: Advanced Algorithms for Scrutiny of Mandatory State Reports Declarations to the IAEA: Final Project Report

Author(s): Mummah, Kathryn Ann
Jackson, Daniel Erik
Oakberg, John
Apt, Kenneth Ellis
Henzl, Vlad

Intended for: Report

Issued: 2024-05-16

Advanced Algorithms for Scrutiny of Mandatory State Reports Declarations to the IAEA

Final Project Report

Kathryn A. Mummah, Daniel E. Jackson,
John Oakberg, Kenneth Apt,
PI: Vlad Henzl

April 2024

Los Alamos
NATIONAL LABORATORY

Contents

Figures

Contents

Tables

Glossary

ABM agent-based modeling

API application programming interface

CSA comprehensive safeguards agreement

FA Facility Attachment

IAEA International Atomic Energy Agency

ICR inventory change report

INMM Institute for Nuclear Materials Management

KMP key measurement point

LOF locations outside facilities

MBA material balance area

MBR material balance report

MDC material description code

PRNG pseudo-random number generator

QCVS Quality Control Verification Software

R&D Research and Development

SME subject matter expert

Abstract

In compliance with their Comprehensive Safeguards Agreements, based on INFCIRC/153 (corrected) (International Atomic Energy Agency, 1972), States Party to the Treaty on the Non-Proliferation of Nuclear Weapons (NPT) are obligated to declare to the International Atomic Energy Agency (IAEA) all changes in their nuclear material inventory as well as movement of the material across boundaries of IAEA recognized material balance areas (MBA), inventories and nuclear material balances. This project addresses capabilities to detect irregularities in State reports, thus ensuring their accuracy and completeness, and in the broader context, States' compliance with safeguards obligations of the NPT. A recent study (Henzl et al., 2022) (lead by this project's PI) demonstrated how analysis of dynamic correlations in nuclear material movement within the entire fuel cycle of a State (viewed as a single system) can reveal variances consistent with and indicative of "irregular" activities. Expanding on this concept, novel ways to analyze State declarations themselves—again, for the State as a whole entity—will help the IAEA draw accurate safeguards conclusions and trust the validity and authenticity of State-generated declaration reports. Introducing new declaration analyses capabilities explored in this project will help to provide credible assurance of both the non-diversion of nuclear material from declared activities and of the absence of undeclared nuclear material and activities in the State in general.

1 Introduction

The International Atomic Energy Agency (IAEA, or Agency) has an interest in applying machine learning to safeguards problems [1]. However, the lack of machine training data for outside researchers is a significant impediment for progress in including advanced algorithms in safeguards practices. Information about a State's inventory and movements of nuclear material is highly sensitive, both from a commercial perspective and from a security perspective. States (and commercial entities) do not freely divulge this information, and the IAEA–which receives detailed information from States as accounting reports–considers all information concerning nuclear materials location and composition to be Safeguards Confidential. Even within the Agency, information marked as safeguards confidential can be accessed only on a direct need-to-know basis by those who analyze and summarize the information for that particular State.

For this reason, there have been few attempts to capture and study comprehensively nuclear materials on the State level. Previous research efforts using process or system-based nuclear fuel cycle simulation tools have focused on improving fidelity at the level of an individual facility, or even a single process. Because no actual International Atomic Energy Agency (IAEA) data are available to researchers, fuel cycle simulations have either focused on simulating with very high temporal and spatial fidelity for one facility, or very low fidelity (one month or longer time steps) for long-term nuclear energy planning. The results of this project provide a new area of safeguards analysis by developing tools to generate synthetic data on nuclear material inventory, composition, and movements with safeguards-relevant spatial and temporal fidelity across an entire State's nuclear fuel cycle.

2 Nuclear material accounting reports

States with a comprehensive safeguards agreement (CSA) are obligated to develop a State system of accounting for and control of nuclear material by which nuclear material accountancy is conducted and accounting information reported to the IAEA. The exact details of safeguards implementation are negotiated between the State and the IAEA through Subsidiary Arrangements. These arrangements have a general part that applies to the entire State and Facility Attachment (FA) that specifies the exact details of safeguards implementation for facilities and locations outside facilitiess (LOFs) in the State. Unlike CSAs, none of the information agreed upon in Subsidiary Arrangements becomes available to the public, unless a State chooses to divulge it.

While there is not a "one size fits all" process that succinctly describes all possible accounting structures, this work follows the nuclear material accounting concepts and reporting structure recommended in the IAEAs Safeguards Implementation Practices Guide on Provision of Information to the IAEA (IAEA Services Series 33) [2] and in Model Subsidiary Arrangements. Most notably, the 10th part, "Code 10 Contents, Format and Structure of Reports to the Agency", referred to as Code 10, details the reporting structure developed by the Agency [3].

There are three versions of Code 10 that States may choose when deciding how to submit their regular accounting reports to the IAEA: labeled format, fixed format, and more recently, XML format, all contain the same information, each presented in a slightly different way. This work focuses on replicating one of the three required State accounting report types, the inventory change report (ICR), in labeled format. The ICR is a record of all transactions in all of a State's facilities under safeguards and represents the bulk of the State's reporting data.

An ICR reports six primary processes:

1. Material enters material balance area (MBA),
2. Material exits MBA,
3. Nuclear process changes to material within a MBA,
4. Re-batching changes the quantity of material tracked as a single unit,
5. Start of safeguards, and
6. Termination of safeguards.

Inventory changes must be reported within 30 days of the calendar month in which the change occurs, which effectively limits ICRs to contain one month's worth of information or less. ICR reports generated in this work are always one calendar month, varying from 28 to 31 days in length depending on the month.

3 Fuel cycle simulation

Nuclear fuel cycle simulators are computational tools designed to track the presence, quantity, and characteristics of nuclear materials as they move between or within nuclear facilities. They range in capability and open access depending on their developer and intended purpose.

CYCLUS, an agent-based nuclear fuel cycle simulator [4] developed at the University of Wisconsin–Madison, was selected for this project. CYCLUS was designed to represent agents as individual processes or facilities in the nuclear fuel cycle. Agents interact by exchanging commodities, most commonly used to represent nuclear materials. The length of time steps is constant and set by the user. For this work, time steps are typically designated as one day in length, so nuclear materials are simulated to be processed and moved through the fuel cycle by the end of each day.

The primary function of CYCLUS is to track the movements of commodities (nuclear material) as they are exchanged, or transacted, between agents within the simulation. Each distinct commodity can have an associated quantity (mass) as well as nuclide composition, and when using the CYCAMORE facility models developed by the CYCLUS team, retains a record of its history when it is split or combined with a different discrete unit of material. This tracking of materials between facilities lends itself well to simulating inventories and inventory changes into and out of a facility or MBA.

Using an agent-based model (ABM) provides a number of benefits to the simulation of the nuclear fuel cycle. CYCLUS itself has no hard-coded notions of chemical or physical processes, instead relying on "plug-and-play" facility models that are responsible for calculating their own physics[5]. This has the effect of allowing any possible kind of nuclear facility or fuel cycle to be modeled, as long as the user has access to or writes their own facility or process models. This work demonstrates the applicability of CYCLUS using only the standard set of simplified facilities, called CYCAMORE, but future work could bring in higher-fidelity models such as a reactor that couples to ORIGEN to model depletion[6], or a pyroprocessing-specific reprocessing facility[7].

Every agent in the simulation acts independently, rather than as a fleet that turn on and off at the same time, and all nuclear materials are tracked individually rather than in aggregate.

Material entering or exiting an MBA, whether by import from another State, movement between facilities within a State, or transfer between MBAs within a facility, are tracked using the transactions data recorded by the nuclear fuel cycle simulator CYCLUS.

One challenge that has arisen with the use of an agent-based fuel cycle simulator for generating synthetic accountancy is the tracking of process changes to materials within an MBA. Agents using the CYCLUS application programming interface (API), which allows third-party nuclear facility models to plug into a simulation, are not required to report their process for making nuclear changes to the material it owns. The standard set of facility models, called CYCAMORE, do record this information, but third-party models that do not participate in the inventory reporting process may not be able to have detailed ICR tracking. To develop a tool that is most able to take advantage of the systems already in place, inventory changes that occur from nuclear changes are identified using the transactions that occur between agents in a facility. This is implemented by checking whether a resource has undergone a compositional change since it entered the facility from which it is recorded as leaving.

The objective of this project is to demonstrate a new capability in generating and evaluating synthetic State nuclear material accounting reports, not to model real countries or regions.

FAs are agreements concluded by a State and the IAEA and contain detailed information about a facility's design, safeguards measures, and accountancy. They include the precise breakdown of

MBAs and key measurement points (KMPs) for the facility, as well as the expected types of inventory changes to occur in the facility, and therefore factor heavily into how materials and their movements are recorded in accounting. Model FAs are often the basis or starting-point for an agreement and propose standard MBAs and KMPs. However, these documents are not widely available to the general Research and Development (R&D) community. Where available, model FAs were used to determine the most realistic set of MBAs to represent a given type of nuclear facility, otherwise a notional three-MBA model is used.

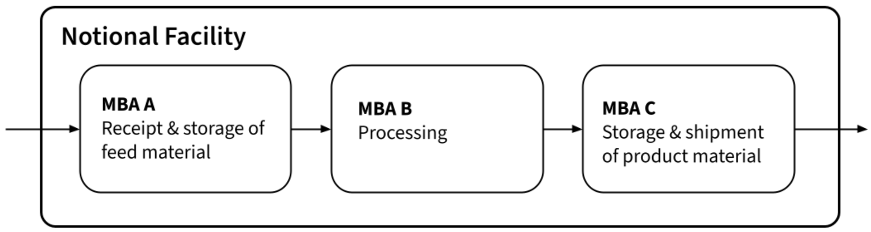


Figure 3-1: Notional three-MBA model for nuclear facilities

4 Increasing fuel cycle simulation fidelity

Nuclear fuel cycle simulators, although created to model flows of nuclear materials throughout the fuel cycle, were not designed with safeguards applications in mind. In our effort to re-purpose Cyclus for creating Code-10 compliant synthetic ICRs, we identified some capability gaps in the existing framework and facility models. The basic facility models developed for Cyclus typically attempt to fill their inventories at each time step. For simulations using one-month time steps this aggregated, bulk movement behavior is adequate, but cannot directly scale down to the one-day time steps that are needed for nuclear material accounting. We have improved Cyclus modeling capabilities to specifically address these shortcomings.

The most commonly used agents developed for Cyclus, the Cycamore facilities, were designed to seek to keep their inventories as full as possible. Whenever space is available, such as when their product is traded to another agent, they immediately attempt to buy more feed, in order to produce more product(s). This assumption is not adequate for simulations with time steps that are one day long, rather than one month, which is common for fuel cycle transition modeling.

Additional control over an agent's demand for feed material becomes available through the introduction of binary behavior states with several strategies to cycle between them. First, the active state, preserves existing buying behavior. The second and new state is dormancy. During a dormant state, agents will place no demands for their feed material, or incoming commodity, regardless of the space available. The default implementation of active and dormant cycling is a permanent active state. This is backwards-compatible with previous versions of the software. The first type of strategies for cycling between active and dormant states are based on time alone. In inventory management, this type of policy is known as a periodic review.

The most straightforward application of active and dormant cycles of behavior is characterized by two fixed values. Given a user-defined active length of time steps and dormant length of time steps, the agent regularly cycles between them. The active state allows regular behavior to proceed; if an agent does not have space in its inventory for new material, it will not place a request even during the active cycle. During the dormant phase, the agent may still process material already in its inventory, but will place no new requests for feed material regardless of space in inventory. There are several ways this behavior can be used to model realistic patterns of material movement. One example is the work week. A facility may be on for five days and then off for two. Given a fixed demand for material per time step, a facility with fixed cycles processes less material than an always-on facility.

Stochasticity can replicate behavior that is periodic and (pseudo-)random. Users can generate random numbers through the Cyclus kernel-managed pseudo-random number generator (PRNG) paired with a distribution of interest. Uniform and normal distributions are implemented in the kernel, but other distributions may be paired with the generator. Adding random distributions to periodic review behavior allows more realistic buying patterns to be modeled. For example, a reactor facility would like its batch of fresh fuel to arrive a certain number of weeks before the outage so the assemblies can be unpacked, moved to the fresh fuel vault, and inspected. However, the dormant time between fresh fuel shipments cannot be well-represented by a fixed number of days; there is likely a shipping window or tolerance. A random buying request may be used in a simulation replicate possible diversion actions that are less regular and more challenging to detect.

The other new capability to control the flow of nuclear material movements into a agent is through continuous review inventory policies. Inventory is reviewed at each time step to determine the remaining inventory space. Two commonly-used inventory management strategies, (s, S) or

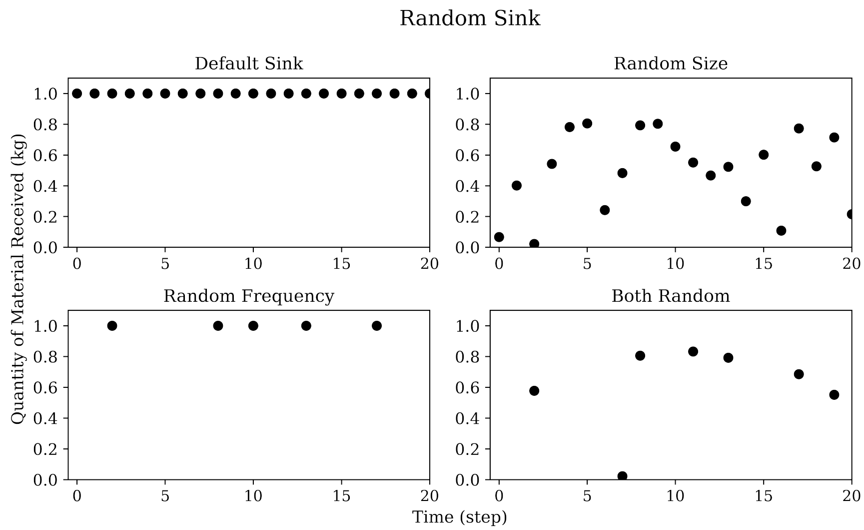


Figure 4-1: New capabilities allow CYCLUS facilities to vary their size and/or frequency of requests

minimum/maximum, and (R/Q) or reorder point/reorder quantity strategies were implemented.

In a minimum/maximum inventory policy, a facility will actively demand more material if it has an inventory below the minimum value, s, including when starting from zero. Once the maximum value, S, is reached the facility enters a dormant phase until the feed material has been processed and sold so that the facility inventory drops below s again. The requests for feed material are scaled at each time step to fill the inventory back up to the maximum. In a (R,Q) inventory policy, the minimum value R functions similarly to the minimum in (s,S). However, the amount to reorder is a fixed quantity, Q.

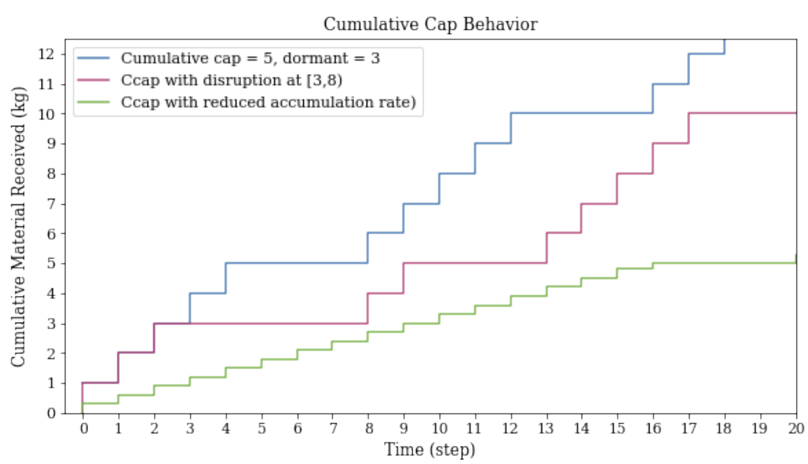


Figure 4-2: Cumulative capacity inventory management stays active until reaching a cumulative amount of material per cycle, in this case 5 kg.

Finally, the cumulative capacity inventory management strategy combines the concept of dormant period with an active cycle based on a total mass of material to be received. Regardless of whether the mass is received in one time step or one thousand, the agent will continue to request material until it has met its cap. Then, the agent will enter a dormant period based on time. This technique allows the simulation to be more flexible in the face of disruptions, either as slow-downs in planned shipments or restrictions based on material availability. This technique is particularly useful for agents that represent processes like fresh fuel buying, which requires a fixed mass every cycle, representing a single batch.

5 A tool to convert nuclear fuel cycle simulations to accounting reports

The data produced directly by nuclear fuel cycle simulators need additional processing to be reflected as synthetic nuclear material accounting data. In particular, there are areas where nuclear fuel cycle simulators produce too many data points, such as detailed accounting of mining and milling activities that are not covered under CSAs. The structures used to represent nuclear facilities for nuclear energy planning do not always correspond directly with core ideas such as MBAs, which are the fundamental spatial unit of IAEA nuclear material accounting.

This project addressed the issue of relevance in simulated data: both by ensuring that extra data are removed and by integrating FA information, such as MBA and KMP codes to correctly partition simulation data.

This situation where two agents (discrete actors) belong to the same MBA requires that agents in a nuclear fuel cycle simulation never contain more than one MBA. For example, Figure 5-1 shows two agents in the same MBA designation. In this case, any movements of nuclear material between the two agents will not be designated as an inventory change, and thus will remain invisible to the ICR. On the other hand, the opposite is not possible. Data for one agent cannot be disaggregated to produce information for two MBAs.

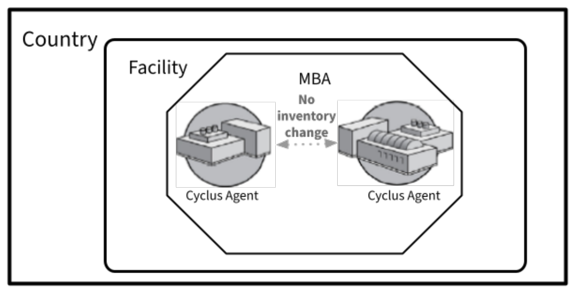


Figure 5-1: Two agents can belong to the same MBA

This project created a new computational tool, CNTAUR, to convert CYCLUS simulations of any fuel cycle to Code 10-type nuclear material accounting reports. We prioritized inventory change reports, which require the generation of nearly all of the data elements specified in the Model Subsidiary Arrangements, General Part, Code 10 Contents, Format, and Structure of Reports to the Agency[3].

Figure 5-2 shows the process of generating reports in Code 10 format for any nuclear fuel cycle that can be simulated using CYCLUS. First, information is gathered about a State's nuclear fuel cycle and other State-specific information. Only synthetic model fuel cycles were used as demonstrations rather than any claim to model a real State. However, the gathering of State-specific information steps mimics realistic gathering of information, but in this case, all the information is hypothetical. This information is combined into a CYCLUS input file, and a simulation is run for the duration of interest. Because ICRs must include the exact date of an inventory change (data label 412), simulations are typically run with a one-day or 86,400-second time step. If the fuel cycle being modeled has evolving facilities, the simulation can deploy and decommission new facilities at a specified time step, or in response to an increased demand for more production capacity by the State.

The CYCLUS simulation output contains detailed information about the location, composition, and movement of nuclear materials throughout every agent in the simulation. The current version of CNTAUR requires two additional user-input files that mimic information present in a FA, in addition to a CYCLUS simulation output in order to run. The first file, referred to as the MBA file, links the agents or individual actors in the CYCLUS world to the three relevant levels of information in an ICR: the country code, facility code, and MBA code. A Boolean parameter notes whether the

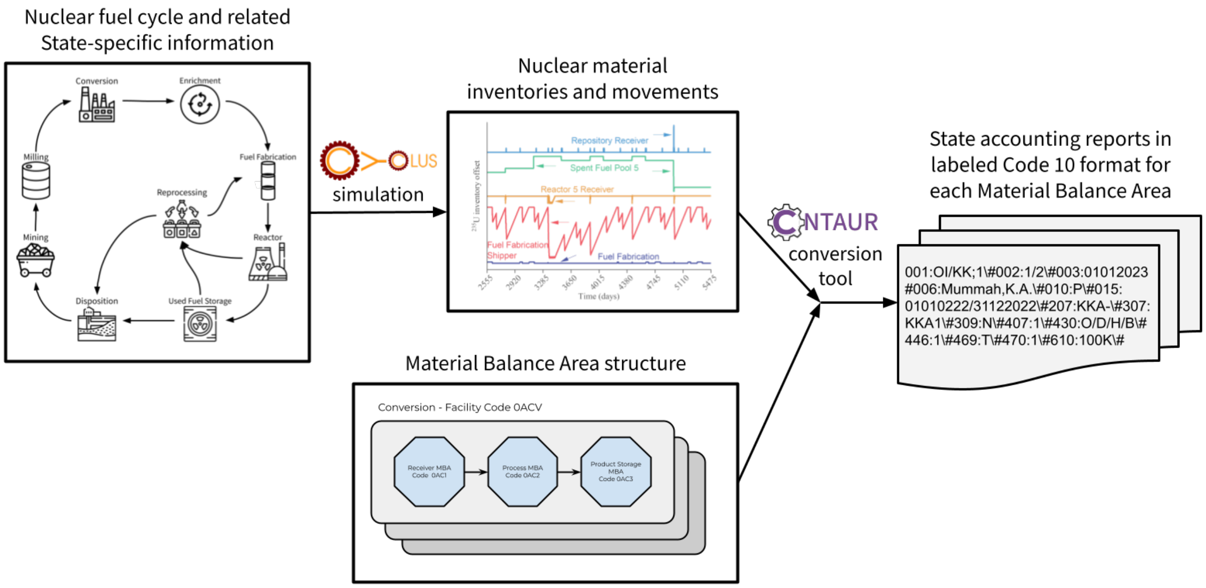


Figure 5-2: Process of generating simulated State accounting data

agent should be used to generate accounting reports. This should be used to exclude facilities that are not required to submit detailed accounting reports under a CSA, such as mining and milling facilities, and any other agents that are ancillary and not representative of real and physical facilities. The second file, called the material description code (MDC) file, helps to link nuclear materials in Cyclus to a four-character code that contains information about the physical and chemical form.

Cntaur takes a Cyclus simulation output file, an MBA file, and an MDC file and generates ICRs for each MBA of interest. Accounting reports can be generated in labeled format, one of the three standard formats that the IAEA will accept. Additionally, the same data can be saved in a convenient CSV format where column headers are the labels and rows are entries in a report.

Figure 5-4 lists accounting report label numbers, always three digits, as well as an example of the entries required for an ICR. While most of the report labels can be generated directly from a Cyclus output file, several pieces of input information are also be generated by users to prepare reports. Non-critical cases, such as correcting a previously-submitted report, or reports for explanation and clarification, Concise Note entries and Textual Reports, are not currently implemented. All labels not available in the current version of Cntaur are listed in Appendix B. Labels (data fields) not in the current version of Cntaur are not relevant to the objectives of this project.

Tables 5-1 through 5-3 describe the main entries of an ICR, with a brief description of how they will be calculated or generated synthetically. Detailed information on each element is available from the IAEA in the Model Subsidiary Arrangement Code 10 Contents, Format and Structure of Reports to the Agency.

Weight data are another key element of accounting reports; we have implemented the most commonly used labels (data fields). Most countries do not report weight data in terms of unified uranium, so Cntaur includes enrichment categories, uranium-235 for enriched uranium, and other reportable nuclear materials as elements. Table 5-4 shows the individual weight data labels currently implemented.

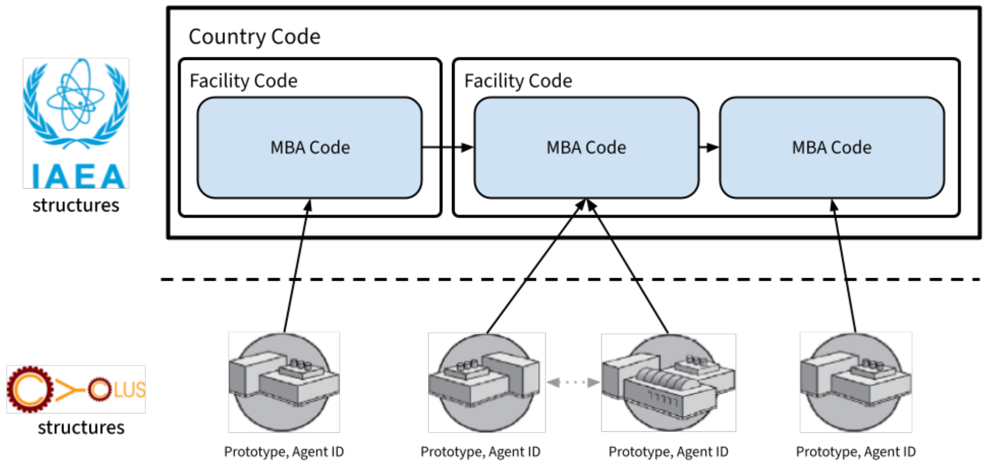


Figure 5-3: The MBA file links agents in a CYCLUS simulation to its country, facility, and MBA code for accounting reports

LABEL	REPORT TYPE	DATA ELEMENT
001: OI / NN ; 000015 #	ICR, MBR, PIL	REFERENCE NUMBER
002: 1 / 1 #	ICR, MBR, PIL	ENTRY NUMBER / TOTAL NUMBER OF ENTRIES
003: 20150124 #	ICR, MBR, PIL	REPORT DATE
006: NAME, I #	ICR, MBR, PIL	ENCODER'S NAME
010: I #	ICR, MBR, PIL	REPORT TYPE
015: 20141201 / 20141231 #	ICR, MBR, PIL	REPORTING PERIOD
099: / #	ICR, MBR, PIL	CONCISE NOTE REFERENCE
207: NND- #	ICR, MBR, PIL	FACILITY CODE
307: NND1 #	ICR, MBR, PIL	MBA CODE
309: N / ; / #	ICR, MBR, PIL	ENTRY STATUS AND CROSS REFERENCE CODE
310: #	ICR	STATE ACCOUNTABILITY SYSTEM RECORD IDENTIFICATION
370: NN / NN-B #	ICR	SHIPPER OF NUCLEAR MATERIAL
372: NN / NND1 #	ICR	RECEIVER OF NUCLEAR MATERIAL
390: #	ICR, MBR, PIL	CONCISE NOTE INDICATOR
391: #	ICR, MBR, PIL	TEXT OF CONCISE NOTE
407: 3 #	ICR, PIL	KEY MEASUREMENT POINT CODE
411: RD #	ICR, MBR	TYPE OF INVENTORY CHANGE, TYPE OF ACCOUNTING ENTRY
412: 20141215 #	ICR	DATE OF INVENTORY CHANGE
430: B / Q / 1 / G #	ICR, PIL	MATERIAL DESCRIPTION CODE
435: / #	ICR, PIL	OPERATOR'S MATERIAL DESCRIPTION CODE
436: #	ICR, PIL	OPERATOR'S MATERIAL DESCRIPTION (TEXT)
445: #	ICR, MBR, PIL	NON-LATIN ALPHABET IDENTIFICATION
446: 045C8 #	ICR, PIL	BATCH NAME
447: #	ICR, PIL	SHIPPER'S BATCH NAME
469: N / / #	ICR, PIL	MEASUREMENT IDENTIFICATION CODE
470: 1 #	ICR, PIL	NUMBER OF ITEMS IN BATCH

WEIGHT DATA:

600: #	610: #	620: #	630: 163257G #	640: #
650: #	660: #	670: 1306G #	680: #	690: #
700: 1498G #	710: #	720: #	730: #	740: #
750: #	760: #	800: #		

The reported data string for the above receipt domestic would be:

001:OI/NN;15#002:1/1#003:20150124#006:NAME, I#010:I#015:20141201/20141231#207:NND-#307:NND1#309:N#
370:NN/NN-B#372:NN/NND1#407:3#411:RD#412:220141215#430:B/Q/1/G#446:045C8#469:N#470:1#
630:163275G#670:1396G#700:1498G#

Figure 5-4: Example of Code 10 labeled format. Image from IAEA

Table 5-1: Data elements calculated from simulated data alone

Label	Name	Description
001	Reference Number	Uniquely identifies a report for filing, processing, sorting and reference purpose
002	Entry Number / Total Number of Entries	Numbers the specific entry within the set of accounting entry
015	Reporting Period	The period covered by the report
411	Type of Inventory Change (ICR), Type of Accounting Entry (material balance report (MBR))	Defines the type of transaction reported or a material balance item respectively
412	Date of Inventory Change	Date on which an inventory change occurred or was established
436	Operator's Material Description Text	Unformatted description of the batch in free text
446	Batch Name	Uniquely identifies a portion of nuclear material handled as a unit for accounting purposes

Table 5-2: Data elements relying on information from the MBA and/or MDC files

Label	Name	Description
003	Report Date	The date on which the report was produced
006	Encoder's Name	The name of the official responsible for the report
207	Facility Code	Identifies the reporting facility
307	MBA Code	Identifies the reporting material balance area
370	Shipper of Nuclear Material	Identifies the shipper of the nuclear material
372	Receiver of Nuclear Material	Identifies the receiver of the nuclear material
407	Key Measurement Point Code	Key Measurement Point Code
430	Material Description Code	Code from the set of Agency-defined codes that describe the physical and chemical form of a batch of material, its packaging/containment and its irradiation status/quality

Table 5-3: Fixed data elements

Label	Name	Description
010	Report Type	I for Inventory Change Report
309	Entry Status and Cross-Reference Code	N for new entry
469	Measurement Identification Code	Indicates when and where the batch was last measured
470	Number of Items in the Batch	1 item

Table 5-4: Weight data implemented

Label	Name	Unit
610	Natural uranium	kg
620	Depleted uranium	kg
630	Enriched uranium	g
640	U-233 isotopic content	g
670	U-235 isotopic content	g
700	Plutonium	g
800	Thorium	kg

A tool to convert nuclear fuel cycle simulations to accounting reports

5.1 CNTAUR input data

Three files are needed to run CNTAUR and generate synthetic nuclear material accounting reports. The output of a CYCLUS simulation, an MBA file, and a MDC file. More detailed information can be found in the CNTAUR User Guide. The bulk of the data needed to run CNTAUR is contained within a stanard CYCLUS output file.

The MBA file is where the user provides the bridge to connect their simulation agents to the nuclear material accounting structures of MBAs and KMP. This file is also where users note which MBAs the user wants accounting reports to be generated for. ?? that don't generate reports can be used to represent other countries, and transactions to or from those locations will be unmatched, or only show up on one end. Individual agents can also be tagged as ghosts, which will not generate reports and transactions to or from those locations will not be recorded at all. This technique can be used to clean up parts of the simulation that handle accounting-irrelevant non-nuclear material like the extra fluorine produced after deconversion, or to test out nefarious "undeclared" actors. Below is an example snippit of a MBA file declaration for a simplified nuclear reactor, represented only by a reactor and a spent fuel pool. Designating a reactor agent, in this case `Reactor_Id15`, ensures that nuclear loss and production are calculated upon final discharge of the fuel. The CYCLUS input file that corresponds to this simple example can be found in Appendix C.

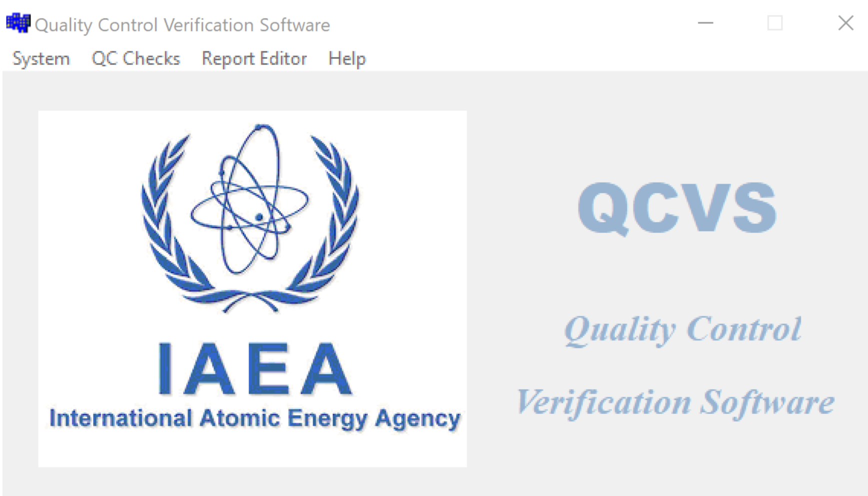
```
<MBA>
    <name>AA01</name>
    <country>AA</country>
    <facility>AA01</facility>
    <agents>
        <agent>Reactor_Id15</agent>
        <agent>SpentFuelPool_Id16</agent>
    </agents>
    <reactor_agent>
        <agent>Reactor_Id15</agent>
    </reactor_agent>
    <inventory_KMPs>
        <KMP>
            <name>A</name>
            <agent>Reactor</agent>
        </KMP>
        <KMP>
            <name>B</name>
            <agent>SpentFuelPool_Id16</agent>
        </KMP>
    </inventory_KMPs>
    <generate_reports>True</generate_reports>
</MBA>
```

A material description code file is also needed because CYCLUS currently lacks a flag to designate how nuclear material as being contained, in the context the IAEA requires. Nuclear materials in a simulation are also only designated by mass, without any necessary density measurements that could be used to determine the relevant volumes. The current version of CNTAUR uses an additional user-input file to that links relevant material description code to the CNTAUR material description management of commodities, however future updates to CYCLUS may be able to automate away the need for an additional MDC file. Below is an example snippet of a MDC file, linking user-defined commodity names in Cyclus corresponding to a particular type of nuclear material to their physical form, chemical form, containment and irradiation status/quality codes.

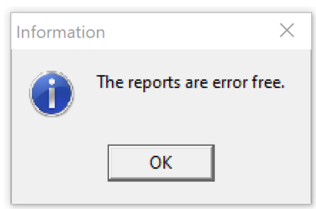
```
{
    "fuel": "B/Q/2/F",
    "spent_fuel": "B/Q/1/G",
    "spent_fuel_cooled": "B/Q/3/G"
}
```

5.2 Validation

We have acquired a version of the IAEA developed software to validate nuclear material accounting reports against the Code 10 style for format compliance, called Quality Control Verification Software (QCVS). Along with our Code 10 subject matter expert (SME), QCVS was used to ensure that Cntaur was replicating nuclear material accounting reports correctly. Fictitious countries that do not share country codes with any existing countries were developed to use in all test cases. Cntaur-created test data, including the simplified single-facility country described above, have successfully passed QCVS quality control checks.



(a) IAEA developed Quality Control Verification Software



(b) Quality Control checks verify whether a snippit is compliant

6 Advanced algorithms

Anomaly detection is a widely developed field in data analytics and is used in a number of areas, including finance [8], network security [9], and various other fields. Unsupervised learning algorithms are often used on data streams to compare past data to current and to identify when some feature in the data is not consistent with past behavior [10]. Fuel cycles should follow regular cadences of operations, which is the impetus for the advanced algorithm portion of this project. By studying how a disruption affects the cadence of operations, it is would be possible to develop an autonomous system that could flag any disruption to the IAEA analyst for further follow up.

As part of this project, we looked into possible existing algorithms that are supported in Python libraries [11]. Online learning [12] is the best option from these tools as it represents a method for a background system to continually review State reporting data and flag any outliers that need further analysis. Online learning allows the model to adapt to legitimate changes to the cadence of operations such as a new reactor coming online, while also being able to continue to flag anomalies in the system. Tests were run with single- and multivariate-data from other sources [13], confirming that the commercial off-the-shelf algorithm worked quickly and accurately on desktop computers. Figure 6-1 shows the result of a random forest advanced algorithm identifying outlier events in NYC taxi riding. The green regions in the top are events that would affect taxi ridership, for example blizzards. The bottom plot shows the outlier scores for two options of random forest modelling.

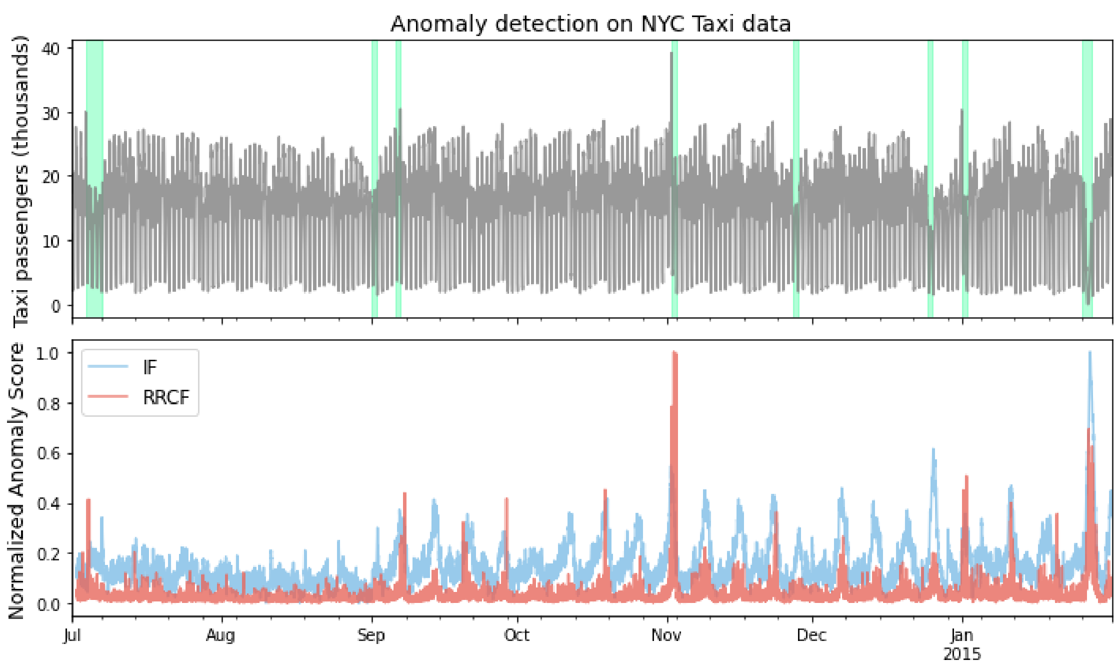


Figure 6-1: A random forest anomaly detection technique example on NYC taxi cab data.

The current state of the Cyclus simulations means there is perfect time symmetry once any steady state develops. This means that the disruptions are both easily visible for humans as well as almost degenerate for a machine algorithm that is following the pattern of data while looking for any possible outliers. Figure 6-2 shows the anomaly detection on one part of a fuel cycle with a major disruption due to a reactor core being emptied early. The period of time outside of the

Advanced algorithms

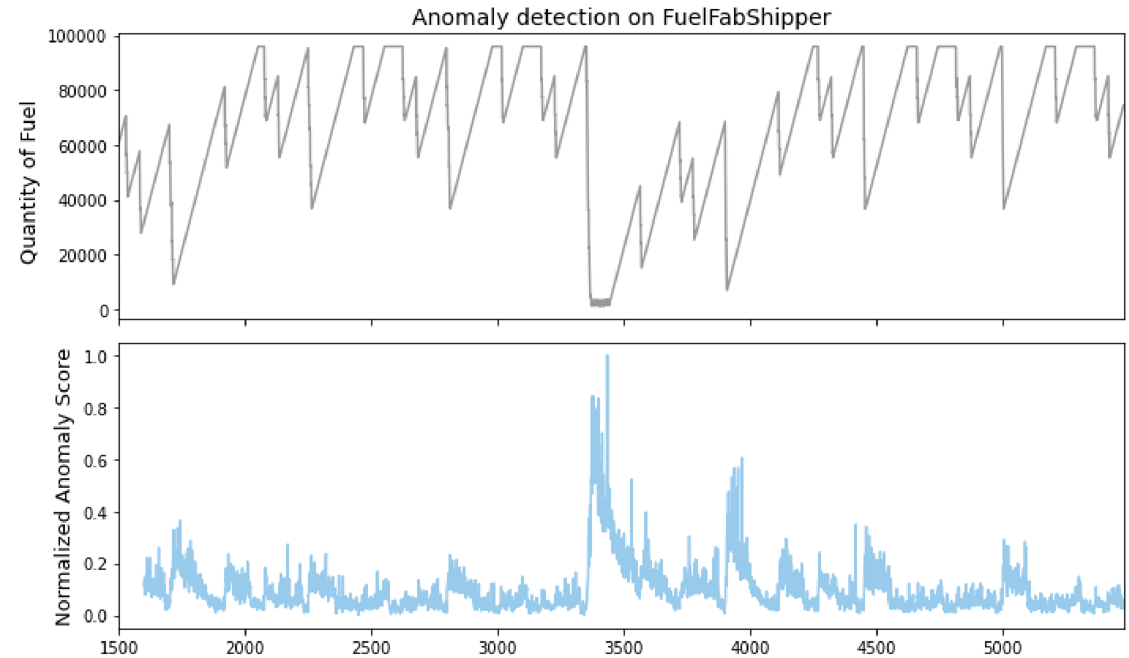


Figure 6-2: The nominal inventory of the Fuel Fabrication Shipper in a Cyclus simulation with a major disruption shows a period of increased anomaly score before the regular cadence of operations resumes

steady state is easily identifiable by the increased anomaly score. Future work on Cyclus will include "noise", which would make disruptions easier to obscure and provide a challenge for the advanced algorithms. Nevertheless, the actual techniques of multivariate anomaly detection have been developed for other fields and should easily be adaptable to advanced fuel cycle analysis.

With future developments in Cyclus to include noise, it will be possible to create millions of diversion scenarios and test the sensitivity of outlier detection methods to the different scenarios. Including multivariate analysis would allow for the combinations of disruptions at various points to identify the source of the initial disruption that has propagated through the network. This would provide a major advantage to the IAEA, as it would reduce the needs of analysts to examine reports except when the algorithm has flagged that the cadence of operations has changed.

7 Summary

This project developed a new capability to reflect nuclear fuel cycle simulations as Code 10-style IAEA nuclear material accounting reports. Coupled with the ability to model diverse and complex nuclear fuel cycles using the CYCLUS nuclear fuel cycle simulator, this presents a powerful new tool that can generate large amounts of synthetic data. Any nuclear fuel cycle that can be modeled in CYCLUS can be used to generate synthetic nuclear material accounting reports.

This capability can be used as a virtual test bed to accelerate the development of novel data processing techniques to more efficiently and effectively analyze State nuclear material accounting reports. By generating data based on the format and content that States with CSAs submit to the IAEA, data scientists and analysts can focus on leveraging advancements in data processing techniques for safeguards evaluation. Because our tool builds on the CYCLUS ecosystem, future developments of nuclear process and facility models for the CYCLUS ecosystem can also be used to generate synthetic nuclear material accounting reports for advanced fuel cycles.

8 Additional information

We have identified related work being undertaken at the IAEA, specifically the Nuclear Solar System (NSS) project which was introduced at the 2022 Institute for Nuclear Materials Management (INMM) Annual Meeting[14]. It is very encouraging to find complimentary work being undertaken at the IAEA, as it both reinforces the mission relevance of our projects and helps us refine our effort to align with the IAEA's most up-to-date R&D priorities.

After careful review of the presentation and paper content, we believe Advanced Algorithms for Scrutiny of Mandatory State's Declarations is complementary to the NSS project but also has an additional scope beyond the NSS's aims. The NSS, as it was presented, is an effort to statistically scale and anonymize real State accounting reports and inspector verified data as training data for a machine learning algorithm, which then generates synthetic but similar reports that could be used in statistical methodology development. Our process, conversely, generates the flow of nuclear materials from models that represent the nuclear and/or chemical processes that occur throughout the nuclear fuel cycle. We are therefore not limited by the constraints of having to model a simple enough system that real data can be sufficiently anonymized and that matches existing nuclear fuel cycles.

To date, we are not aware of any opportunities to access the NSS project outside the IAEA. But, since we have the ability to recreate a system similar to the three-State "universe" developed by the NSS, we would benefit from the opportunity to validate our capabilities by comparing our model to the synthetic Universe data. If the opportunity arises to use facility models from the NSS project, they could be encapsulated as agents and used directly in a simulation to produce much more realistic data.

In FY22, A briefing was given to the IAEA on the previous SG Policy project "Analysis of States' Cadence of Nuclear Operations: Detecting Safeguards Relevant Anomalies" and the then-current status of this project along with our future goals for the project.

NNSA Nuclear Nonproliferation and International Safeguards (NNIS) graduate fellow and project co-investigator Kathryn Mummah has worked on this project while conducting a long-term laboratory assignment at Los Alamos National Lab. This project is ending before Kathryn has finished the chapter of her dissertation that encompasses this work. After Kathryn defends her dissertation "State-Level Nuclear Fuel Cycle Simulations for Safeguards Applications" for the degree of PhD, Nuclear Engineering and Engineering Physics from the University of Wisconsin–Madison, she will ensure a copy of her dissertation with additional detail on this project will be made available to the Concepts and Approaches and Policy (which funded the previous project mentioned above) subprograms. Kathryn intends to defend in calendar year 2024.

References

[1] "Enhancing capabilites for nuclear verificaiton," International Atomic Energy Agency, Vienna, Austria, Tech. Rep. STR-399, Jan. 2022.

[2] "Safeguards Implementation Practices Guide on Provision of Information to the IAEA," en, International Atomic Energy Agency, Vienna, Austria, Safeguards Implementation Practices (SIP) Guide IAEA-SVS-33, 2016. [Online]. Available: https://www.iaea.org/publications/11083/safeguards-implementation-practices-guide-on-provision-of-information-to-the-iaea (visited on 06/30/2022).

[3] "Model Subsidiary Arrangement Code 10 Contents, Format and Structure of Reports to the Agency," International Atomic Energy Agency, Vienna, Austria, Tech. Rep. SG-FM-1172, Jan. 2017.

[4] K. D. Huff, M. J. Gidden, R. W. Carlsen, *et al.*, "Fundamental concepts in the Cyclus nuclear fuel cycle simulation framework," *Advances in Engineering Software*, vol. 94, pp. 46–59, Apr. 2016, ISSN: 0965-9978. DOI: 10.1016/j.advengsoft.2016.01.014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0965997816300229 (visited on 07/31/2016).

[5] M. J. Gidden, "An Agent-Based Modeling Framework and Application for the Generic Nuclear Fuel Cycle," English, ISBN: 9781321656183, Ph.D. The University of Wisconsin - Madison, United States – Wisconsin, 2015. [Online]. Available: http://search.proquest.com/docview/1671784496/abstract/A58332D062B04DB4PQ/1 (visited on 01/15/2021).

[6] S. E. (Skutnik, "ORIGEN-based Nuclear Fuel Inventory Module for Fuel Cycle Assessment: Final Project Report," English, Univ. of Tennessee, Knoxville, TN (United States), Tech. Rep. DOE/NEUP-13-5415, Jun. 2017. DOI: 10.2172/1364128. [Online]. Available: https://www.osti.gov/biblio/1364128/ (visited on 07/20/2021).

[7] G. T. Westphal and K. D. Huff, "PyRe: A Cyclus Pyroprocessing Facility Archetype," in *Proceedings of the 2018 Advances in Nuclear Nonproliferation Technology and Policy Conference*, Orlando, FL: American Nuclear Society, Nov. 2018.

[8] M. Ahmed, A. N. Mahmood, and M. R. Islam, "A survey of anomaly detection techniques in financial domain," *Future Generation Computer Systems*, vol. 55, pp. 278–288, 2016, ISSN: 0167-739X. DOI: https://doi.org/10.1016/j.future.2015.01.001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X15000023.

[9] I. Fosić, D. Žagar, K. Grgić, and V. Križanović, "Anomaly detection in netflow network traffic using supervised machine learning algorithms," *Journal of Industrial Information Integration*, vol. 33, p. 100 466, 2023, ISSN: 2452-414X. DOI: https://doi.org/10.1016/j.jii.2023.100466. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2452414X23000390.

[10] J. Audibert, P. Michiardi, F. Guyard, S. Marti, and M. A. Zuluaga, "Usad: Unsupervised anomaly detection on multivariate time series," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '20, Virtual Event, CA, USA: Association for Computing Machinery, 2020, pp. 3395–3404, ISBN: 9781450379984. DOI: 10.1145/3394486.3403392. [Online]. Available: https://doi.org/10.1145/3394486.3403392.

References

[11] Y. Zhao, Z. Nasrullah, and Z. Li, "PyOD: A Python Toolbox for Scalable Outlier Detection," *Journal of Machine Learning Research*, vol. 20, Jun. 2019, arXiv:1901.01588 [cs, stat]. DOI: 10.48550/arXiv.1901.01588. [Online]. Available: http://arxiv.org/abs/1901.01588 (visited on 09/20/2023).

[12] R. Adams and D. MacKay, "Bayesian online changepoint detection," *Arxiv*, no. 0710.3742v1, 2007.

[13] *TLC Trip Record Data*, 2023. [Online]. Available: https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page (visited on 09/20/2023).

[14] E. Brayfindley, S. Cormon, and C. F. Normon, "The Nuclear Solar System—a synthetic data generation framework for safeguards," Virtual, Jul. 2022.

A Model fuel cycles

A.1 State A Fuel Cycle

State A is a resource-rich State with a relatively young nuclear fuel cycle.

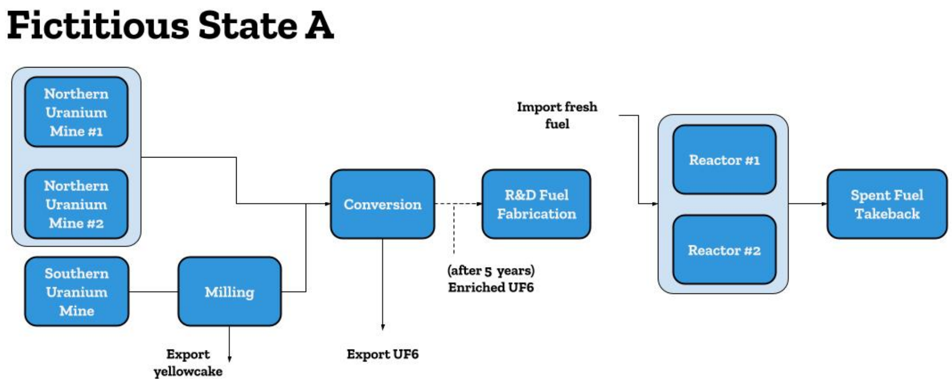


Figure A-1: Model fuel cycle A

It originally was a uranium exporting State only, and still provides Yellowcake and UF6 to other States. The State has two PWRs at a single power facility near its capital, with long-range plans for a third PWR unit at a separate location in the country. All PWRs are LEU UO2 fueled, the technology for which was provided by supplier State C. None of State A's uranium output goes directly to its own power reactor fuel, as it has only a pilot LEU fuel fabrication plant. State A has plans for a full-scale UO2 fabrication plant in the future, but these have not been realized. It receives fresh power reactor fuel from supplier State C under its original power reactor supply agreement. State A does not have a permanent spent fuel repository, so it stores spent fuel in pools at the power reactor facility. It has a fuel take-back agreement with supplier State C, and has begun returning spent fuel to State C for final disposition in that State. Fuel cycle elements of State A include

- Uranium Mining
- Uranium Milling
- Uranium Refining/Conversion
- Import of Power Reactor Fuel Elements
- Power Reactors
- Spent Fuel Storage
- Transfer of Spent Fuel for Final Disposal

A.2 State B Fuel Cycle

State B is a developed State with well-established fuel cycle capabilities.

It imports UF6 from State A and with its relatively new gas-centrifuge enrichment plant (GCEP) enriches uranium to LEU fuel grade. It produces fresh fuel elements for its own 8 PWRs, as well as for export to other States. The State's two research reactors are TRIGA designs and obtain 19.9% LEU fuel from a separate supplier State. State B does not have permanent spent fuel storage facilities, and it stores spent fuel in pools at the power reactor facility. It has a separate agreement

Model fuel cycles

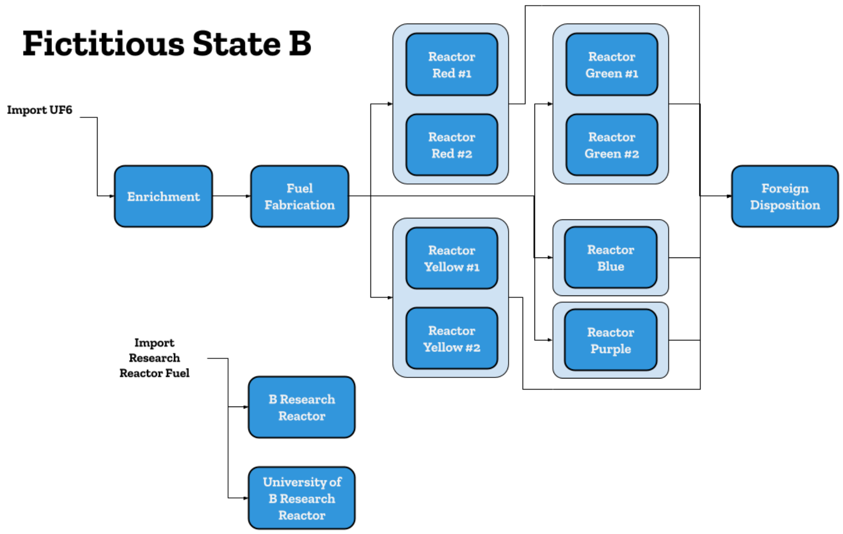


Figure A-2: Model fuel cycle B

with supplier State C for spent fuel disposition and returns a significant percentage of its spent fuel to State C for final disposition in that State. The agreement also allows for State C to reprocess State B's spent fuel for plutonium separation. Fuel cycle elements of State B include

- Import of UF6
- Import of TRIGA Fuel Elements
- Uranium Enrichment (GCEP)
- Fuel Fabrication (including conversion from UF6 to UO2)
- Power Reactors
- Research Reactor
- Spent Fuel Storage
- Export of Spent Fuel for Long-term Dry Storage and/or Reprocessing

A.3 State C Fuel Cycle

State C is a long-time advanced fuel cycle State and serves as a world supplier of nuclear fuel-cycle and power reactor capabilities.

It has no uranium reserves and must import U3O8 and UF6 from State A and historically has done so from other States. State C provided two LEU PWRs to State A under a long-standing supplier agreement, which also provides fresh fuel elements for the PWRs and allows for take-back of spent fuel from State A for final disposal. State C has over 20 LEU PWRs of its own design and several older units from a separate nuclear supplier State. The State's several research reactors are TRIGA and custom Chinese designs, which obtain 19.9

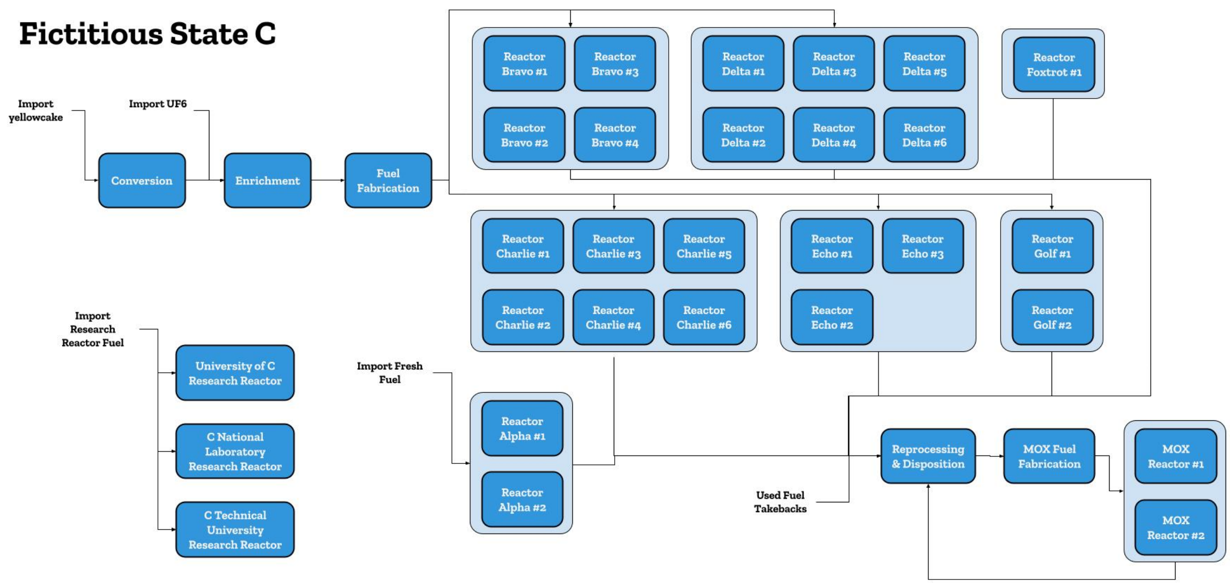


Figure A-3: Model fuel cycle C

B Code 10 labels not implemented

Table B-1: Weight elements not implemented in CNTAUR

Label	Name	Unit
600	Unified uranium	g
650	U-234 isotopic content	g
660	U-233 + U-235 isotopic content	g
680	U-236 isotopic content	g
690	U-238 isotopic content	g
710	Pu-238 isotopic content	g
720	Pu-239 isotopic content	g
730	Pu-240 isotopic content	g
740	Pu-241 isotopic content	g
750	Pu-242 isotopic content	g
760	Pu-239 + Pu-241 isotopic content	g
770	Natural uranium fissile content	g
780	Depleted uranium fissile content	g

Table B-2: Data elements not implemented in CNTAUR

Label	Name	Description
099	Concise Note Reference	Provides the country, facility, MBA, report as a whole or entry to which the Concise Note refers
310	State Accounting System Record Identification	Identifies the corresponding information in the State accounting system
390	Concise Note Indicator	Calls attention to a Concise Note attached
391	Text of Concise Note	Other unformatted information
436	Operator's Material Description Code	The code used by the operator to identify the type of nuclear material
445	Non-Latin Alphabet Identification	A code to indicate that a non-Latin alphabet was used in the report and to identify that alphabet
447	Shipper's Batch Name	Identifies the shipper's batch name in the reporting of a receipt

C Example

A minimal working example of CNTAUR is presented in this Appendix. The country in question is fictitious country AA, with single reactor facility AA01. The reactor's parameters have been simplified to demonstrate the capability of CNTAUR.

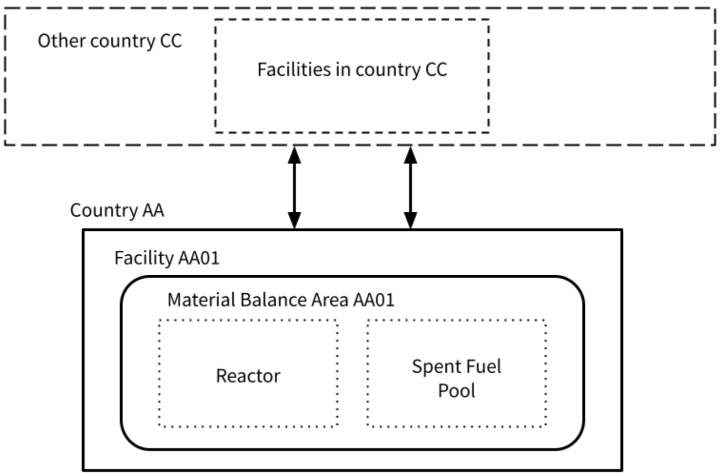


Figure C-1: Accounting structure of example state AA

Input files

Cyclus *input file*

```
<simulation>
<control>
    <dt>86400</dt>
    <duration>121</duration>
    <startyear>2020</startyear>
    <startmonth>1</startmonth>
</control>

<archetypes>
    <spec><lib>agents</lib><name>NullRegion</name></spec>
    <spec><lib>agents</lib><name>NullInst</name></spec>
    <spec><lib>cycamore</lib><name>Source</name></spec>
    <spec><lib>cycamore</lib><name>Reactor</name></spec>
    <spec><lib>cycamore</lib><name>Sink</name></spec>
    <spec><lib>cycamore</lib><name>Storage</name></spec>
</archetypes>

<facility>
    <name>Source</name>
    <config>
        <Source>
```

Example

```
            <outcommod>fuel</outcommod>
            <outrecipe>fuel_recipe</outrecipe>
        </Source>
    </config>
</facility>

<facility>
    <name>Reactor</name>
    <config>
        <Reactor>
            <fuel_incommods>
                <val>fuel</val>
            </fuel_incommods>
            <fuel_inrecipes>
                <val>fuel_recipe</val>
            </fuel_inrecipes>
            <fuel_outcommods>
                <val>spent_fuel</val>
            </fuel_outcommods>
            <fuel_outrecipes>
                <val>spent_fuel_recipe</val>
            </fuel_outrecipes>
            <assem_size>100</assem_size>
            <cycle_time>30</cycle_time>
            <n_assem_core>3</n_assem_core>
            <n_assem_batch>1</n_assem_batch>
        </Reactor>
    </config>
</facility>

<facility>
    <name>SpentFuelPool</name>
    <config>
        <Storage>
            <in_commods>
                <val>spent_fuel</val>
            </in_commods>
            <out_commods>
                <val>spent_fuel_cooled</val>
            </out_commods>
            <residence_time>14</residence_time>
        </Storage>
    </config>
</facility>

<facility>
    <name>Sink</name>
    <config>
        <Sink>
            <in_commods>
                <val>spent_fuel_cooled</val>
            </in_commods>
```

```
      </Sink>
    </config>
  </facility>

  <recipe>
    <name>fuel_recipe</name>
    <basis>mass</basis>
    <nuclide>
      <id>922350000</id>
      <comp>0.0265</comp>
    </nuclide>
    <nuclide>
      <id>922380000</id>
      <comp>0.8442</comp>
    </nuclide>
    <nuclide>
      <id>80160000</id>
      <comp>0.1186</comp>
    </nuclide>
  </recipe>

  <recipe>
    <name>spent_fuel_recipe</name>
    <basis>mass</basis>
    <nuclide><id>922340000</id><comp>0.00012</comp></nuclide>
    <nuclide><id>922350000</id><comp>0.00456</comp></nuclide>
    <nuclide><id>922360000</id><comp>0.00311</comp></nuclide>
    <nuclide><id>922380000</id><comp>0.73401</comp></nuclide>
    <nuclide><id>942380000</id><comp>0.00017</comp></nuclide>
    <nuclide><id>942390000</id><comp>0.00384</comp></nuclide>
    <nuclide><id>942400000</id><comp>0.00197</comp></nuclide>
    <nuclide><id>942410000</id><comp>0.00080</comp></nuclide>
    <nuclide><id>942420000</id><comp>0.00051</comp></nuclide>
    <nuclide><id>80160000</id><comp>0.11852</comp></nuclide>
    <nuclide><id>10010000</id><comp>0.13239</comp></nuclide>
  </recipe>

  <region>
    <name>SingleRegion</name>
    <config>
      <NullRegion/>
    </config>
    <institution>
      <name>SingleInstitution</name>
      <config>
        <NullInst/>
      </config>
      <initialfacilitylist>
        <entry><prototype>Source</prototype><number>1</number></entry>
        <entry><prototype>Reactor</prototype><number>1</number></entry>
        <entry><prototype>SpentFuelPool</prototype><number>1</number></entry>
        <entry><prototype>Sink</prototype><number>1</number></entry>
```

Example

```
            </initialfacilitylist>
        </institution>
    </region>

</simulation>
```

Country file

This country file is the same one shown in Section 5.1.

```
<data>
    <MBA>
        <name>CC01</name>
        <country>CC</country>
        <facility>CC1–</facility>
        <agents>
            <agent>Source_Id14</agent>
            <agent>Sink_Id17</agent>
        </agents>
        <generate_reports>False</generate_reports>
    </MBA>

    <MBA>
        <name>AA01</name>
        <country>AA</country>
        <facility>AA01</facility>
        <agents>
            <agent>Reactor_Id15</agent>
            <agent>SpentFuelPool_Id16</agent>
        </agents>
        <reactor_agent>
            <agent>Reactor_Id15</agent>
        </reactor_agent>
        <inventory_KMPs>
            <KMP>
                <name>A</name>
                <agent>Reactor</agent>
            </KMP>
            <KMP>
                <name>B</name>
                <agent>SpentFuelPool_Id16</agent>
            </KMP>
        </inventory_KMPs>
        <generate_reports>True</generate_reports>
    </MBA>
</data>
```

Material description code file

The three types of nuclear materials in this simulation include fresh nuclear fuel, imported from another country, spent fuel discharged from the reactor, and spent fuel "cooled" and ready to ship to another facility (in this simulation, to a fuel take-back out of country). This MBA file is the same one shown in Section 5.1.

```
{
    "fuel": "B/Q/2/F",
    "spent_fuel": "B/Q/1/G",
    "spent_fuel_cooled": "B/Q/3/G"
}
```

Cntaur Output

The following Code 10 reports were produced

Report 1: January

```
001:OI/AA;1#002:1/6#003:20200302#006:TEST,TEST#010:I#015:20200101/20200131#207:AA01
    #307:AA01#309:N#370:CC/CC#372:AA/AA01#407:1#411:RF#412:20200101#430:B/Q/2/F
    #436:fuel#446:13#447:13#469:N#470:1#630:88011.7G#670:2678.7G#
001:OI/AA;1#002:2/6#003:20200302#006:TEST,TEST#010:I#015:20200101/20200131#207:AA01
    #307:AA01#309:N#310:1#370:CC/CC#372:AA/AA01#407:1#411:RF#412:20200101#430:B/Q
    /2/F#436:fuel#446:15#447:15#469:N#470:1#630:88011.7G#670:2678.7G#
001:OI/AA;1#002:3/6#003:20200302#006:TEST,TEST#010:I#015:20200101/20200131#207:AA01
    #307:AA01#309:N#310:2#370:CC/CC#372:AA/AA01#407:1#411:RF#412:20200101#430:B/Q
    /2/F#436:fuel#446:17#447:17#469:N#470:1#630:88011.7G#670:2678.7G#
001:OI/AA;1#002:4/6#003:20200302#006:TEST,TEST#010:I#015:20200101/20200131#207:AA01
    #307:AA01#309:N#310:4#370:AA/AA01#372:AA/AA01#407:A#411:LN#412:20200131#430:B/Q
    /1/G#436:spent_fuel#446:76#447:76#469:N#470:1#630:13831.7G#670:2222.7G#
001:OI/AA;1#002:5/6#003:20200302#006:TEST,TEST#010:I#015:20200101/20200131#207:AA01
    #307:AA01#309:N#310:3#370:CC/CC#372:AA/AA01#407:1#411:RF#412:20200131#430:B/Q
    /2/F#436:fuel#446:83#447:83#469:N#470:1#630:88011.7G#670:2678.7G#
001:OI/AA;1#002:6/6#003:20200302#006:TEST,TEST#010:I#015:20200101/20200131#207:AA01
    #307:AA01#309:N#310:4#370:AA/AA01#372:AA/AA01#407:A#411:NP#412:20200131#430:B/Q
    /1/G#436:spent_fuel#446:76#447:76#469:N#470:1#700:729.0G#
```

Report 2: February

```
001:OI/AA;2#002:1/1#003:20200330#006:TEST,TEST#010:I#015:20200201/20200228#207:AA01
    #307:AA01#309:N#310:5#370:AA/AA01#372:CC/CC#407:2#411:SF#412:20200215#430:B/Q
    /3/G#436:spent_fuel_cooled#446:76#447:76#469:N#470:1#620:74.18K#700:729.0G#
```

Report 3: March

```
001:OI/AA;3#002:1/4#003:20200430#006:TEST,TEST#010:I#015:20200301/20200331#207:AA01
    #307:AA01#309:N#310:7#370:AA/AA01#372:AA/AA01#407:A#411:LN#412:20200303#430:B/Q
    /1/G#436:spent_fuel#446:145#447:145#469:N#470:1#630:13831.7G#670:2222.7G#
```

Example

```
001:OI/AA;3#002:2/4#003:20200430#006:TEST,TEST#010:I#015:20200301/20200331#207:AA01
    #307:AA01#309:N#310:6#370:CC/CC#372:AA/AA01#407:1#411:RF#412:20200303#430:B/Q
    /2/F#436:fuel#446:152#447:152#469:N#470:1#630:88011.7G#670:2678.7G#
001:OI/AA;3#002:3/4#003:20200430#006:TEST,TEST#010:I#015:20200301/20200331#207:AA01
    #307:AA01#309:N#310:7#370:AA/AA01#372:AA/AA01#407:A#411:NP#412:20200303#430:B/Q
    /1/G#436:spent_fuel#446:145#447:145#469:N#470:1#700:729.0G#
001:OI/AA;3#002:4/4#003:20200430#006:TEST,TEST#010:I#015:20200301/20200331#207:AA01
    #307:AA01#309:N#310:8#370:AA/AA01#372:CC/CC#407:2#411:SF#412:20200318#430:B/Q
    /3/G#436:spent_fuel_cooled#446:145#447:145#469:N#470:1#620:74.18K#700:729.0G#
```

Report 4: April

```
001:OI/AA;4#002:1/4#003:20200530#006:TEST,TEST#010:I#015:20200401/20200430#207:AA01
    #307:AA01#309:N#310:10#370:AA/AA01#372:AA/AA01#407:A#411:LN#412:20200403#430:B/
    Q/1/G#436:spent_fuel#446:214#447:214#469:N#470:1#630:13831.7G#670:2222.7G#
001:OI/AA;4#002:2/4#003:20200530#006:TEST,TEST#010:I#015:20200401/20200430#207:AA01
    #307:AA01#309:N#310:9#370:CC/CC#372:AA/AA01#407:1#411:RF#412:20200403#430:B/Q
    /2/F#436:fuel#446:221#447:221#469:N#470:1#630:88011.7G#670:2678.7G#
001:OI/AA;4#002:3/4#003:20200530#006:TEST,TEST#010:I#015:20200401/20200430#207:AA01
    #307:AA01#309:N#310:10#370:AA/AA01#372:AA/AA01#407:A#411:NP#412:20200403#430:B/
    Q/1/G#436:spent_fuel#446:214#447:214#469:N#470:1#700:729.0G#
001:OI/AA;4#002:4/4#003:20200530#006:TEST,TEST#010:I#015:20200401/20200430#207:AA01
    #307:AA01#309:N#310:11#370:AA/AA01#372:CC/CC#407:2#411:SF#412:20200418#430:B/Q
    /3/G#436:spent_fuel_cooled#446:214#447:214#469:N#470:1#620:74.18K#700:729.0G#
```

QCVS configuration

In order to verify the working example above using QCVS, the tool was configured with the data of the the fictional countries. Because country AA is not required to know the facility and MBA codes of their trading partner country CC, those were left blank.

File	Data
Countries	AA, CC
DomesticFac	AA01
DomesticMBA	AA01
ForeignFac	
ForeignMBA	

Table C-1: Configuration of QCVS

The reports above passed the QCVS quality control checks.