WM2013 Conference, February 24 – 28, 2013, Phoenix, Arizona USA

Expert System for Building TRU Waste Payloads – 13554

Heather Bruemmer and Bryant Slater
Information Systems Laboratories, 2235 East 25th Street, Suite 100, Idaho Falls, ID 83404
hbruemmer@islinc.com, bslater@islinc.com

ABSTRACT

The process for grouping TRU waste drums into payloads for shipment to the Waste Isolation Pilot Plant (WIPP) for disposal is a very complex process. Transportation and regulatory requirements must be met, along with striving for the goals of shipment efficiency: maximize the number of waste drums in a shipment and minimize the use of empty drums which take up precious underground storage space. The restrictions on payloads range from weight restrictions, to limitations on flammable gas in the headspace, to minimum TRU alpha activity concentration requirements. The Overpack and Payload Assistant Tool (OPAT) has been developed as a mixed-initiative intelligent system within the WIPP Waste Data System (WDS) to guide the construction of multiple acceptable payloads. OPAT saves the user time while at the same time maximizes the efficiency of shipments for the given drum population. The tool provides the user with the flexibility to tune critical factors that guide OPAT's operation based on real-time feedback concerning the results of the execution. This feedback complements the user's external knowledge of the drum population (such as location of drums, known challenges, internal shipment goals). This work demonstrates how software can be utilized to complement the unique domain knowledge of the users. The mixed-initiative approach combines the insight and intuition of the human expert with the proficiency of automated computational algorithms. The result is the ability to thoroughly and efficiently explore the search space of possible solutions and derive the best waste management decision.

INTRODUCTION

The mission of the Waste Isolation Pilot Plant (WIPP) is the safe disposal of the nation's defense-related transuranic radioactive waste in a deep geological repository in southeastern New Mexico. The Waste Data System (WDS) is custom software that was created in 2008-2009 to manage the significant quantity of data and complexity of regulations involved in the process of characterizing, certifying, transporting and emplacing the waste. The WDS replaced the previous legacy system (the WIPP Waste Information System, or WWIS) with modern, flexible software written in a modular fashion that allows for adaptability in the face of the increasing challenges faced by the waste shipping sites. The WDS is web-based software written in Java that incorporates all previous functionality of the legacy systems, insight gained from software written for the Department of Energy (DOE) sites, and added planning, forecasting, and reporting tools. The process of creating and maintaining the WDS strictly adheres to the software lifecycle development guidelines found in the IEEE Software Engineering standards [1-8], and meets or exceeds the quality assurance demands of ASME NQA-2 [9-10]. The WDS project is managed by Nuclear Waste Partnership LLC (NWP), with support from the software team at Information Systems Laboratory (ISL).

One of the significant improvements included in the first release of the WDS in 2009 was a planning interface for assembling overpack containers and payloads. Payloads are waste container assemblies placed in a Type B shipping container for shipment, with payload configurations (number and organization) varying by payload container type and Type B shipping container type (TRUPACT-II, HalfPACT, TRUPACT-III, and 72-B). Overpack containers are simply intermediary filtered payload container layers used to assemble damaged or otherwise hard-to-ship containers prior to placement in the shipping container. When building a payload or an overpack container, the user must consider multiple data points that factor into compliance with the Contact-Handled Transuranic Waste Authorized Methods for Payload Control (CH-TRAMPAC) regulations [11-12]. The factors to consider when combining drums into these groupings include: the gross weight of container and contents; the total fissile gram equivalent (FGE) of the contents; and the gas generation properties of the waste and waste packaging. Overpack containers must meet additional container-level constraints imposed by the Transuranic Waste Acceptance Criteria for the WIPP (WAC) [13], such as a minimum TRU Alpha Activity Concentration value and ^{239}Pu equivalent Curie (PE-Ci) limits. In the legacy WWIS application, the interface for creating overpacks provided no immediate visual access to the container data needed to support these decisions, and the payload interface provided limited access to this data (see Figure 1).

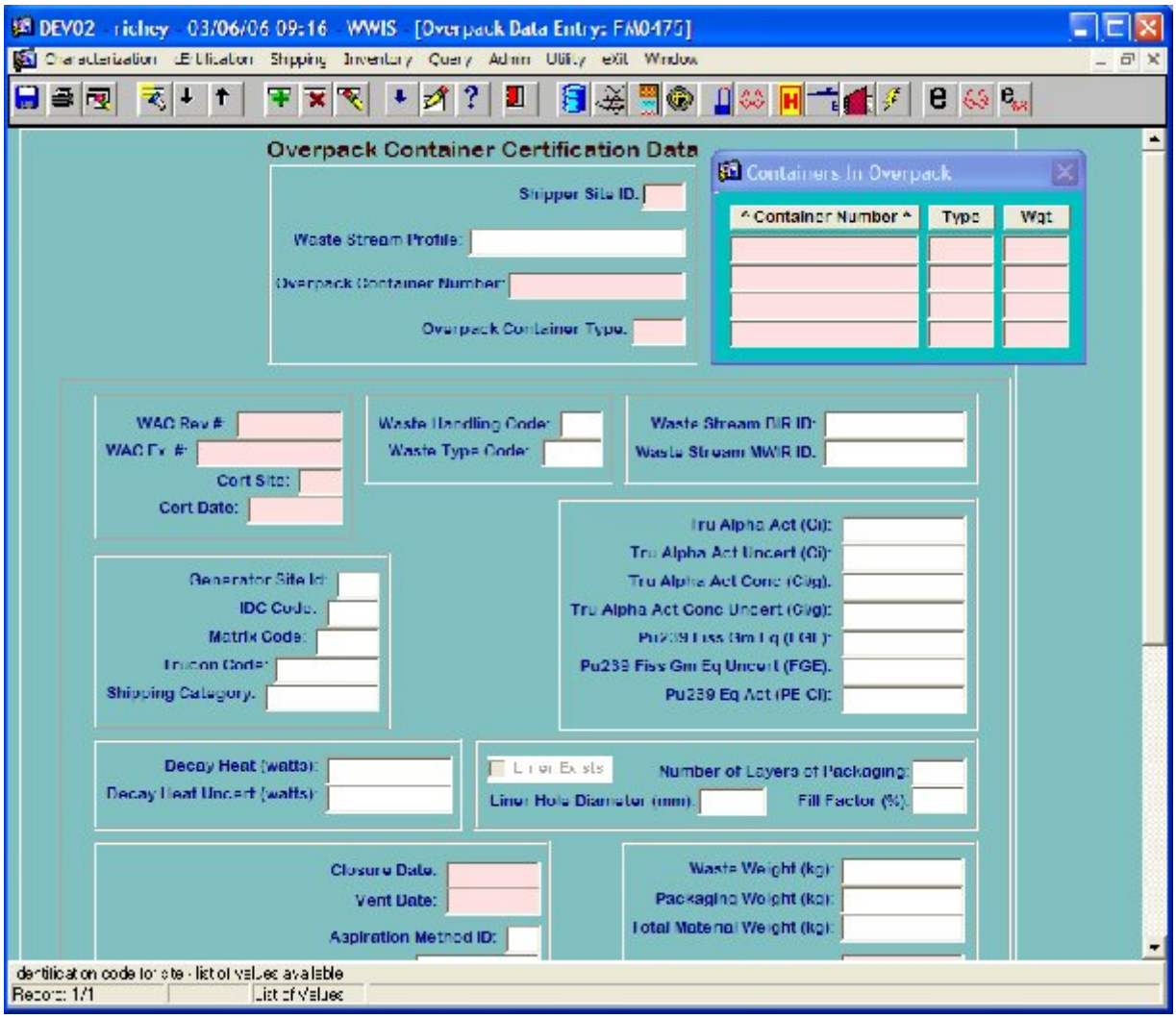

Figure 1. Screenshot of the legacy WWIS system Overpack Screen shows the limited information provided concerning overpack inner containers.

The overall goal of the user building overpacks and payloads is the same goal of the WIPP mission – the safe and efficient disposal of TRU waste. From a software standpoint, safety is enforced by the implementation of the guiding regulations. For efficiency improvements, the focus is on maximizing waste volume per shipment, which in turn generally implies minimizing the use of dunnage containers (empty containers). Minimal use of dunnage containers in assemblies also meets the goal of maximizing usage of the underground storage space. To accomplish these goals, the user needs ready access to all relevant input data that affects whether or not dunnage containers must be used. Dunnage containers are used to complete configurations when the

addition of the remaining required containers would cause weight, FGE, gas generation or other limits to be exceeded. The overpack and payload planning screens within the WDS provide query and sorting tools for the available certified containers for a given site, and immediate feedback concerning configuration totals and availability of the software evaluation components enforcing each regulation.

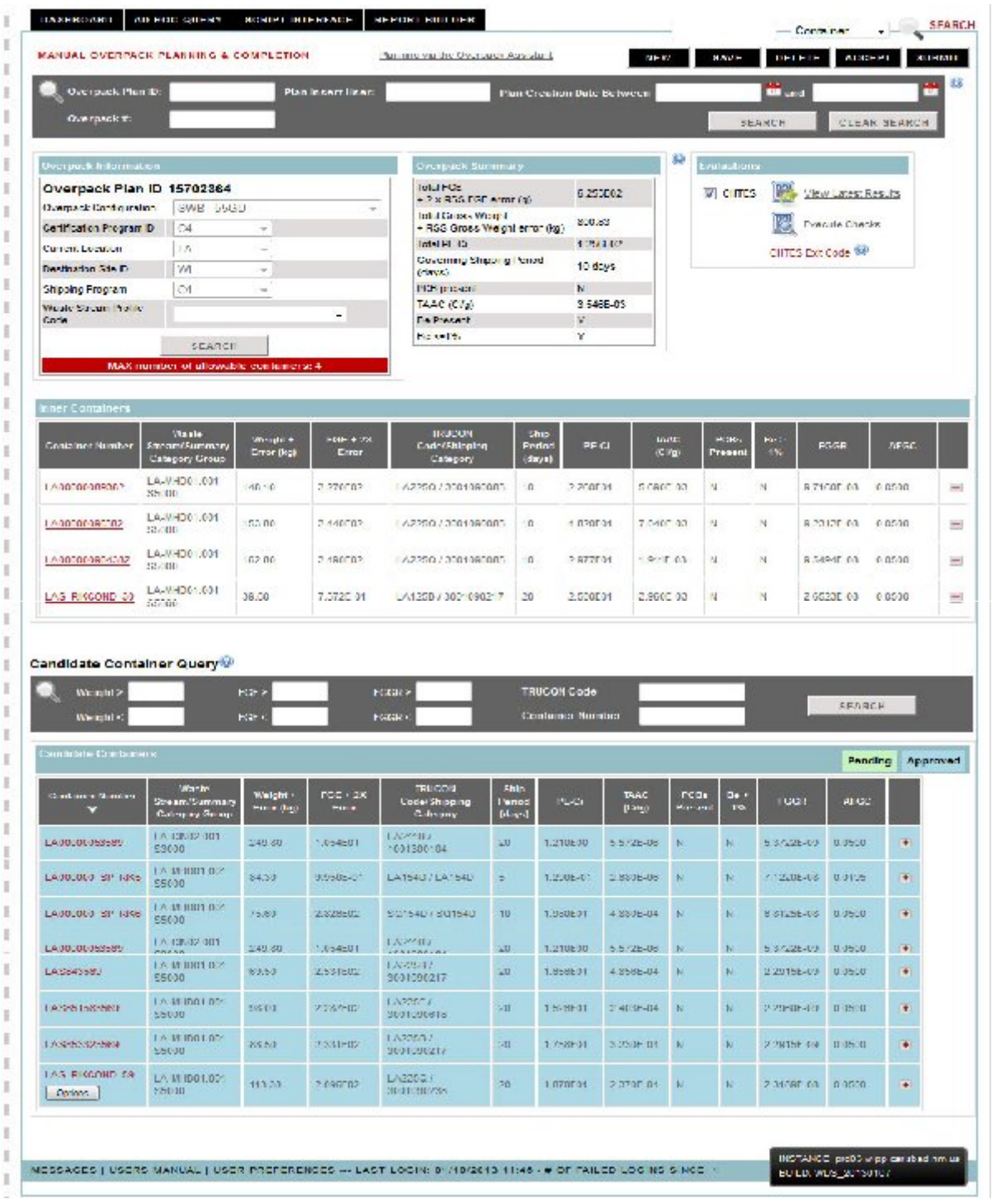

Figure 2. Screenshot of the WDS Manual Overpack Planning Screen highlights the improved user interface provided in the WDS over the WWIS, with increased data display, sorting features, and query tools.

While the new WDS planning screens provide increases in efficiency in terms of building "maximal" configurations and certainly significant time improvements over the legacy system, the system designers considered whether or not it would be possible to automate the process of building overpacks and payloads. The process of manually building a single overpack or payload can take several hours, and several users may be performing the process simultaneously. While a user may be focusing on creating a single maximal configuration, it is difficult for a single user, no less a group of users, to keep the "big picture" of maximally shipping the available waste population in mind. Automation of the process would provide significant time benefits, along with potentially providing overall grouping efficiency since the entire available population could be considered in a single execution.

The concept of a software "assistant" to automate the building of overpacks and payloads was introduced in the early versions of the WDS, and the tool provided the ability to select a candidate pool and have the software build compliant payloads and overpacks from the selection, with controls provided that manage the number of dunnage containers allowed. The early versions of the assistant were heavy on automation, and very light on user control. User feedback indicated that they needed flexibility in terms of how candidate containers were provided to the system. Additionally, they needed the ability to tailor the software's applied algorithms based on known properties of the candidate waste. Finally and perhaps most significantly, they needed greater feedback from the system to guide their decisions in terms of refining the candidate inputs, algorithm planning selections, and even procedural decisions concerning whether or not "difficult" waste drums (containers close to exceeding one or more regulatory limit) should attempt to be shipped as-is or re-packaged/processed.

Utilizing this feedback, the WDS designers introduced a mixed-initiative systems approach to overpack and payload building called the Overpack and Payload Assistant Planning Tool (OPAT). "Mixed-initiative systems integrate human and automated reasoning to take advantage of their complementary reasoning styles and computational strengths" (IJCAI-03 Workshop on Mixed-Initiative Intelligent Systems [14]). In simple terms, these systems allow the human to do what he is good at (using specialized, experiential knowledge of the problem), and allow computers to do what they are good at (performing multiple, complex computations or following numerous logical paths to find an optimal solution). The problem of assembling containers into compliant overpacks or payloads is an ideal problem for a mixed-initiative system: the automated aspect of the tool can easily assemble and test thousands of potential configurations within minutes, while the human user guides the system with their special knowledge of the waste characteristics. In the following discussion of the details of the system, the focus is on the overpack portion of the system, which contains the full complexity of the system.

METHODS

OPAT is a software tool within the WDS that takes a candidate pool of eligible waste containers and builds the maximum number of TRAMPAC-compliant overpack or payload plans from the selected pool. Each portion of the overpack version of the tool will be described below, including the applicable algorithms, with special consideration given to both the efficiency provided by the software system and the tool interface parameters that provide the user with the ability to accurately direct the algorithm.

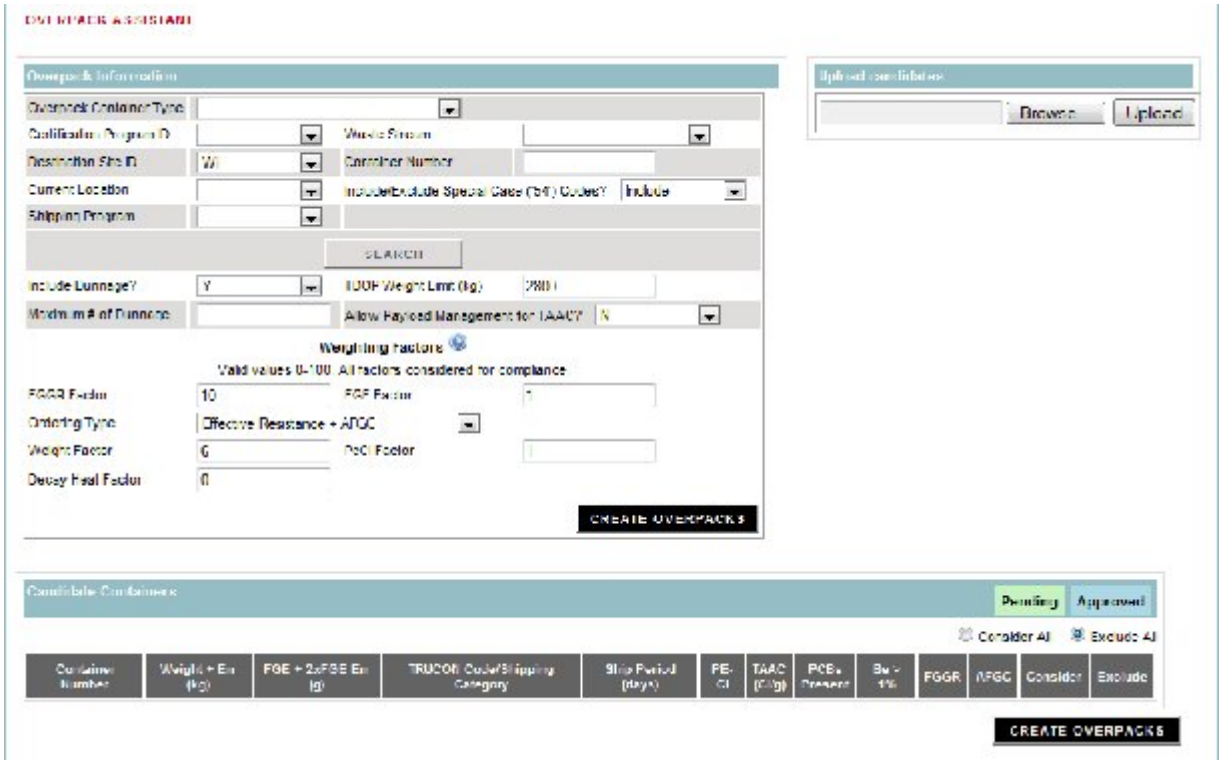

Figure 3. Screenshot of the WDS Overpack Assistant Screen displays the tool interface parameters.

Obtaining Candidates – Search and Upload

The basic options for candidate search can be seen in the screenshot above (see Figure 3). Candidates can be narrowed down by the factors displayed on the top left portion of the screen, and the query executes among all certified containers currently available in the database. The candidates appear in the "Candidate Containers" portion of the screen upon completion of the query, and all relevant data is displayed. An early improvement to this interface involved adding the ability to individually check containers for consideration by or exclusion from the algorithm – this was needed as some containers may not be available for physical or other procedural

considerations. Over time, individually checking off perhaps 50 out of 150 containers was found to be inconvenient, and the file upload functionality was introduced (see top right of screen). This function accepts a basic text file, with one eligible container listed per line. The Waste Certification Officials (WCOs) who build the overpacks often receive such lists from the individual waste sites containing the containers that are currently available, so this functionality directly fits with the existing system processes and provides a significant user interface efficiency benefit.

Organizing the Candidates for Algorithmic efficiency: Weighting Factors, Assessments, and Ordering the Container List

The goal of calculating a composite assessment for each container is to end up with a list that is ordered optimally with respect to consideration by the algorithm. Determining what order is "optimal" is a considerably difficult problem, due to the fact that there are several parameters which could potentially play into the calculation. The following assessments are calculated for each container:

- Weight Assessment: Computed as the ratio of the gross weight plus error to the mean weight limit (weight limit for the overpack type/ max # of containers for the overpack type). Ratios > 1 indicate heavier containers that must be balanced with lighter containers.

- FGE Assessment: Computed as a ratio of the FGE plus two × FGE error to the mean FGE limit (highest FGE limit for the overpack/ max # of containers for the overpack type). Ratios > 1 indicate containers with high FGE that must be balanced with containers with lower FGE.

- PE-Ci: Computed as a ratio of the PE-Ci value to the mean PE-Ci limit for the overpack.

- TAAC: Computed as a ratio of the (minimum TRU Alpha Activity Concentration limit (100 nCi/g) divided by the max # of container for the overpack type) to the TAAC for the container. Since this is a MINIMUM limit rather than a maximum, the ratio is reversed so that containers with a high value for the ratio indicate a relatively LOW TAAC value (low TAAC value makes a container relatively harder to place).

- Decay Heat: Computed as a ratio of the decay heat plus error to the mean packaging design decay heat limit, which is determined based on the selected package type (always TRUPACT-II in the assistant) and waste type of the containers.

- Flammable Gas: The flammable gas assessment determination is the most complex. There are three options for this assessment based on the value selected for "Ordering Type" by the user:

- Effective Resistance: When this value is selected, the assessment is calculated as the ratio of the effective resistance from the shipping category divided by the upper bound for the total resistance currently used by the defined shipping categories (10^7).

- Effective Resistance + AFGC: When this value is selected, the calculation takes the Effective Resistance assessment, and multiplies it by the ratio of .05 (the standard AFGC) to the AFGC for the container (AFGC = allowable flammable gas concentration – it is a value that considers the effects of VOCs on the allowable amount of hydrogen).

- Consider only FGGR factor: In this case, none of the assessment values are used outside of the flammable gas value. The software calculates the Effective Resistance type assessment, but utilizes it in a different ordering algorithm (explained as an alternative below).

Composite Assessment

The composite assessment for a container is calculated by summing the product of each individual assessment with its designated 'weighting factor' (WF):

$$\text{Composite Assessment} = \sum \text{WF}_i \times \text{Assessment}_i$$

Where i = container parameter

The user can enter values from 0-100 for each WF. A value of zero for a WF effectively eliminates the associated parameter from the composite assessment.

WFs can be used to focus the assistant in terms of container ordering on the factor(s) most relevant to the data set. For example, if the most significant factor affecting the production of complete plans is gross weight, then the assistant should order the container list from heaviest to lightest. This can be accomplished by setting the WF for gross weight to some value >=1, and setting all other WFs to zero.

Note: WF values do NOT affect the creation of plans that meet ALL the requirements – they only affect the ordering of the container list.

Ordering the Container List

There are two choices for ordering the container list. The default choice utilizes the composite assessment. This approach will order the list from highest to lowest values calculated for the composite assessment and considers all factors (as set by the user-entered parameters). The second approach is implemented when the "Consider only FGGR Factor" is selected for ordering

type. This approach is used when the gas generation properties are by far the most limiting factors in the waste population, and the algorithm maximizes the container ordering by first grouping the containers based on the total packaging resistance (as represented by their assigned shipping category), and then sorting the containers within those groupings from lowest to highest AFGC. This ordering is optimal for creating the maximum number of overpacks that comply with the TRAMPAC flammable gas limits.

Creating the Plans

There are multiple steps to creating a plan, involving several iterations over the container set. The primary goal of the algorithm is to build a full plan (i.e., utilizing the maximum number of containers allowed in the overpack type) that meets all transportation and regulatory requirements.

The operation begins with the creation of a plan using the first container in the list, and iterates down the ordered container list, attempting to add each successive container to the plan. At this stage, the algorithm is operating as a 'greedy algorithm'. At each step (addition of a container to the plan), the algorithm determines if the plan is 'feasible' prior to the addition of the container. We know that due to the ordering of the container list per the entered weighting factors, that the selection of container at each step is 'locally optimal' (as optimal has been defined for the execution by the user-entered values for the WFs). Once a container is deemed acceptable to the overall plan, its addition (during this operation) is irrevocable (see [15] for a definition of 'greedy algorithm'). The majority of the feasibility analyses are straightforward comparisons of the total value for the given factor and the applicable limit. The flammable gas feasibility analysis is the most complex, and involves modeling the combined gas generation properties of the configuration via system of linear equations (payload calculations models based on the CH-TRAMPAC Payload Appendix 2.4 [11]).

If the plan is full and meets all limits, then the algorithm accepts the plan. If the plan is NOT full or does not meet the TAAC (we know at this point it meets all other requirements), then we would like to see if we could do 'better' with that first container and a different set of containers from the candidate list. At this point, the algorithm begins iterating through each position in the plan (other than the first and last positions), and for each position, iterates through the remaining candidate list, attempting to build a more 'optimal' plan (optimality determined by the new plan being larger than the existing best try AND meeting all restrictions). The algorithm stops at any point when a full plan meeting the requirements is built. The algorithm retains the largest, compliant plan built as it proceeds through the iteration.

The plan creation is an "iterative improvement" algorithm [15]. The algorithm begins by establishing a feasible candidate (the first plan built), then proceeds to improve upon that by re-attempting to build the plan by successively removing each of the containers currently at each

subsequent position in the plan (following the first one) and attempting to re-build a full plan from that position. At all times, it maintains the *most* optimal solution considered up until that point (i.e., the fullest passing plan), returning this solution when all possibilities have been exhausted.

Providing Feedback on Limiting Factors

An added feature of the upgrades to the Overpack Assistant is the tracking of the parameters on which attempted overpack plans failed and then providing a summary of this information to the user as feedback. The purpose of this feedback is to help guide the user in setting values for the weighting factors, as well as guiding selection of containers for consideration.

As an example, the image below (see Figure 4) shows the feedback popup displayed after an execution of the Overpack Assistant. The user who receives this message may decide to modify the 'ordering type' to be 'Consider only FGGR factor' and re-execute the assistant to see if a better plan proposal is returned (i.e., more full plans).

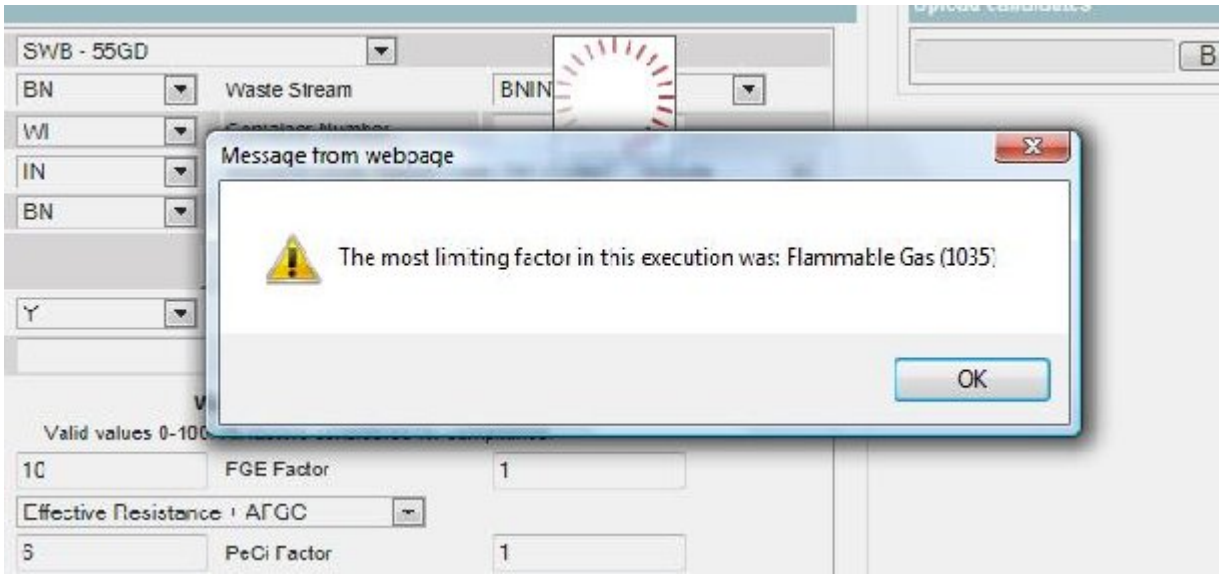

Figure 4. Screen shot of example showing limiting factor results from an execution of OPAT. The number in parentheses, 1035, indicates that 1035 overpack plans were attempted but failed due to exceeding flammable gas limits.

When a failure is encountered in a feasibility check, the failure is recorded (e.g., a potential plan that failed the feasibility check for FGE). At the end of the algorithm execution, the failures are added into the running totals and the values are translated into a message posted to the user in the popup.

RESULTS

The time improvement by using OPAT vs. the manual overpack building process is readily apparent (see Table I below). Even with minor rework for physical or procedural limitations that crop up that would prevent some of the automatically created overpacks from being feasible, there is still an overwhelming time benefit to using the automatic tool. The interface is straightforward to use and requires very little training. OPAT is designed to generate compliant results, so the user can essentially not fail to use it correctly.

Table I. Potential time efficiency gained from utilizing OPAT.

Overpack Type	Population Size	Potential Number of Overpacks	Time for Manual Building (Assumes average 1 hour/overpack)	Time for OPAT
Standard Waste Box (4 drums/overpack)	160	40	40 hours	< .25 hours
	40	10	10 hours	<.1 hours
Ten-drum Overpack	110	11	11 hours	<.25 hours
	40	4	4 hours	< .1 hours

The benefits of using OPAT in terms of creating the "ideal" number of compliant overpacks with minimal use of dunnage containers in terms of efficiently shipping and emplacing the pool of eligible candidate containers as a whole is a little bit more difficult to measure. We have recently started tracking the use of OPAT to create plans versus the use of the manual system, and preliminary results show that just over 25% of overpacks are being built using OPAT. The OPAT-created plans utilize approximately 50% less dunnage on average than those created through the manual process.

Factors beyond our control, such as smaller available candidate pools, limit the application of the tool in production. However, the hope is that over time, as the waste populations become increasingly difficult to ship, OPAT will ease the burden on the user to create compliant overpacks in a safe, time-effective manner.

CONCLUSIONS

OPAT was developed to address the increasingly difficult problem of creating multiple compliant payload configurations in an overall time and mission-efficient fashion: maximizes waste shipped, minimizes use of dunnage containers. The software design was created over time based

on interaction with the user community, starting with a simple approach and building on the user interface features and complexity of the algorithm as the process matured. User satisfaction was greatly improved by switching from a mainly *automated* approach with little user input or system flexibility, to a *mixed-initiative* approach that allowed the user to provide guidance to the automated portion of the system. The system designers expect to continue to hone the system's efficiency in future releases. Specifically, the goal is to minimize the rework needed on overpacks produced by the system, so that the user can smoothly move from use of the tool directly to shipment of the generated configurations with few or no rejected products.

The process of iteratively constructing software solutions using mixed-initiative control that combines automation with human expertise has the potential to benefit a multitude of waste management problems, where systems are complex and safety and efficiency are of critical importance.

REFERENCES

[1] IEEE Recommended Practice for Software Design Descriptions, IEEE Std. 1016-1998

[2] IEEE Recommended Practice for Software Requirements Specifications, IEEE Std 830-1998

[3] IEEE Standard for Quality Assurance Plans, IEEE Std 730-2002

[4] IEEE Standard for Software Configuration Management Plans, IEEE Std 828-2005

[5] IEEE Standard for Software Maintenance, IEEE Std 1219-1998

[6] IEEE Standard for Software Project Management Plans, IEEE Std 1058-1998

[7] IEEE Standard for Software Reviews, IEE Std 1028-1997 (R2002)

[8] IEEE Standard for Software Verification and Validation, IEEE Std 1012-2004

[9] ASME NQA-2A, Subpart 2.7 (1989.1990), Sections 3.2 "Design Phase" and 6.3 "Software Design and Implementation Documentation"

[10] ASME NQA-2A, Subpart 2.7 (1989.1990), Section 6.2, "Software Requirements Documentation"

[11] CH-TRU Payload Appendices, Rev. 2, February 2009.

[12] Contact-Handled Transuranic Waste Authorized Methods for Payload Control (CH-TRAMPAC), Rev. 3, February 2009.

[13] Transuranic Waste Acceptance Criteria for the Waste Isolation Pilot Plant,

DOE/WIPP-02-3122, Rev 7.2, June 13, 2011.

[14] Iterative Improvement. Luay Nakhleh, Department of Computer Science, Rice University. Spring 2012 semester notes. http://www.clear.rice.edu/comp182/class/IterativeImprovement.pdf

[15] "Workshop on Mixed-Initiative Intelligent Systems." *IJCAI-03 Workshop on Mixed-Initiative Intelligent Systems*. N.p., 2003. Web. 09 Nov. 2012. <http://lalab.gmu.edu/miis/>.