Model Traceability Procedure for an Automotive-Safety Virtual-Testing Framework

Marcy A. Edwards, Jason L. Forman, Bronislaw Gepner, Pablo Gracia Cemborain, Adrian Caudillo-Huerta, Daniel Perez-Rapela, Corina Klug, Alexander Gromer, Masahiro Okamura, Nobuhiro Taki, Jessica S. Jermakian

Abstract Building safety robustness in modern vehicles will require safety assessment agencies to expand the scope of their evaluations to include more types of crashes, occupants and injuries. Virtual testing, using computational models of human surrogates and vehicles, provides an opportunity to expand the scope of physical tests. This study aimed to develop a feasible method for model traceability in virtual testing where trusted simulations can add value to assessment programmes. Discussions with industry, stress testing existing solutions and refining were used to outline the requirements for a virtual testing framework where simulations are run by automakers. A prototype finite-element solver-based tool was developed to test if the requirements could be achieved. The requirements, process and tool together provide a procedure for model traceability that ensures the information reported to the assessment agency comes from the inputs and outputs run by the solver, the simulation outputs come from the simulation inputs and no inappropriate changes have been made between validation and assessment load cases in the simulation, while accommodating necessary changes and protecting the intellectual property of automakers and suppliers. By adhering to these requirements, assessment agencies can expand physical assessments with computational tools, like human body models, while maintaining consumer trust.

Keywords Virtual testing, Human body models, Consumer information programmes, Simulation, Fingerprinting

I. INTRODUCTION

Beginning with the introduction of the seat belt, vehicles have been on a trajectory of continuous improvement to eliminate deaths and reduce injuries in motor vehicle crashes [1-3]. With the introduction of holistic restraint systems and improved vehicle safety designs, the rate of crash deaths per 100,000 people has decreased by 38% from 1992 to 2022 [3]. Vehicle safety assessment programmes worldwide have helped encourage improved vehicle safety designs despite limitations to the crash scenarios and human surrogates used in physical crash testing. For example, the Insurance Institute for Highway Safety (IIHS) frontal moderate overlap crash test, where 40% of the width of the vehicle impacts a barrier, approximates many frontal vehicle-to-vehicle crashes that occur in the real world. In this test, a human surrogate with 50th percentile male anthropometry is used to best represent the large range of drivers injured in frontal crashes. Though both the crash conditions and human surrogate are approximations of what happens in real-world crashes, vehicle improvements driven by crash testing have benefited all occupants [4]. A 2006 study found that drivers of good-rated vehicles in the IIHS moderate overlap test were 46% less likely to die in a frontal crash than drivers of poor-rated vehicles [5].

However, to continue building safety robustness in vehicles, it's important for assessment programmes worldwide to expand their capability to evaluate various crash, occupant, and injury types observed in the field. Though large reductions in injuries in crashes have been made using ATDs, they are designed to balance the demands of durability and biofidelity and are significantly limited in the types of humans and human injuries they can represent. Computational models of humans (human body models) have been in development for over 30 years, and are becoming increasingly more sophisticated in their ability to predict human injury [6-9]. Human

M. A. Edwards (medwards@iihs.org) is a Senior Research Engineer and J.S. Jermakian is a Senior Vice President of Vehicle Research at the Insurance Institute for Highway Safety in Virginia, USA. J. Forman is an Associate Professor (jlf3m@virginia.edu), B. Gepner is a Senior Scientist (bg5ah@virginia.edu), A. Caudillo-Huerta (afc5b@virginia.edu) is a Research Engineer and P. Gracia Cemborain (fmx7jf@virginia.edu) is a Research Engineer at the University of Virginia Center for Applied Biomechanics in Virginia, USA. D. Perez-Rapela (dperezrapela@hic-research.com) is the Owner and Consultant at HIC Research in Virginia, USA. C. Klug (corina.klug@tugraz.at) is an Associate Professor at Graz University of Technology in Graz, Austria. A. Gromer (alexander.gromer@ansys.com) is a Senior Manager Application Engineering at DYNAmore, an ANSYS Company, Munich, Germany. M. Okamura (okamura.masahiro@jsol.co.jp) is a Senior Computer-Aided Engineering (CAE) specialist and N. Taki (taki.nobuhiro@jsol.co.jp) is a CAE engineer at JSOL Corporation in Tokyo, Japan.

body models also have the potential to represent a broader range of humans than ATDs, providing a means to assess the robustness of protection provided by safety systems with consideration for person-to-person variability. Offering better tools for human injury prediction and the ability to broaden the scope of safety assessments without increasing the testing burden makes virtual testing a vital tool to continue the trend of understanding injury and advancing safety.

Virtual testing is currently used in the European New Car Assessment Programme's (Euro NCAP's) pedestrian impactor test to determine the head contact times for hardware evaluations of deployable devices such as pop-up hoods [10]. Several NCAP programmes and IIHS have announced plans to expand their assessment programmes using virtual testing, first using computational models of physical ATDs to validate vehicle models, then expanding to human body models (HBMs) where more occupant sizes and injury types can be assessed [11]. Currently, Euro NCAP's workflow involves simulation load cases that could potentially be validated by auditing with a matched physical test [11]. IIHS plans to include virtual tests for the first time in rear impact assessments. Initially, these rear impact assessment tests will only include ATDs as human surrogates to allow IIHS to use physical test auditing as an oversight mechanism for simulation results. In the future, however, as ratings programmes expand to include more virtual tests that cannot be physically validated, it is crucial to have a procedure for ensuring that simulation outcomes are trustworthy and meaningful. In particular, as human body finite-element (FE) models are considered for virtual testing programmes, it will be impossible to use matched physical test auditing as an oversight mechanism for simulation results. Instead, the virtual assessment framework would require that the environment model be validated in a scenario that is physically testable (e.g., with ATD tests), and then frozen for use in scenarios that are not physically testable (e.g., with human body models). While manual tracking and reporting processes are the standard today, confidence can be further built through implementing model version control and tracking mechanisms to verify that the models used for the untestable assessment simulations are, in fact, the same as the models that were used for the validation simulations.

Model tracking and verification for virtual testing introduces challenges that do not exist in physical tests. While safety assessment agencies can purchase and test a physical vehicle, their computational equivalents exist only within the automaker. Further, since computational models are discretized approximations of their physical counterparts, they require an appropriate validation procedure and are more prone to mistakes and manipulation that may cause the models to behave in a physically unrealistic manner when used in scenarios outside of their validation regime. Overcoming these challenges requires simulation-specific tools in a virtual testing framework that protects automaker intellectual property and respects their simulation workflow. Model fingerprinting and version control tools are an incremental step that can be used to verify that models have not changed from validation to assessment in a manner that does not require automakers to pass sensitive model information to the assessment agency. Fingerprinting is a method of generating a unique identifier to represent a dataset, like FE model files, to allow a third party to confirm that the data contents have not changed without having access to the contents of the data files. In 2022, [12] presented the first concept and corresponding prototype tool to generate hashes (i.e., a type of fingerprint which transforms data to a fixed-length string) of simulation input files and then link them with hashed output files in a signed report. The aim of the current study was to build on this work through stress testing the existing tool and then refining the requirements, processes and tools needed for effective model traceability in a virtual testing framework. The procedure for model traceability presented in this study leverages fingerprinting and other cryptographic technologies (e.g., secure signatures) to ensure proper model traceability and outlines the information exchange needed to build trust in simulation results while remaining practical for automakers.

II. METHODS

First, automakers were surveyed on their experience and tools for using simulation in occupant safety evaluations. Automakers who expressed interest then provided more detailed feedback on using virtual testing in automotive safety assessments. From this feedback and discussion with industry experts, goals and tasks were

outlined to develop the appropriate oversight procedure for virtual testing. Task 1 focused on developing guidelines for model traceability in virtual testing (VT). Task 2 focused on developing the rules and requirements for the model traceability process. Task 3 tested the ability of software tools to adhere to these guidelines and requirements. In the first iteration of Task 3, the existing Python wrapper tool presented in [12] was stress tested against the guidelines. After this exercise, the goals were refined to recognize that not all misconduct can be prevented, and the guidelines and traceability rules were refined to ensure proper functionality of software tools. Next, guidelines and traceability rules were finalized. Last, prototype software developed by ANSYS that supports the execution of the model traceability within the LS-DYNA FE solver was stress tested to confirm its ability to support the finalized guidelines and traceability rules.

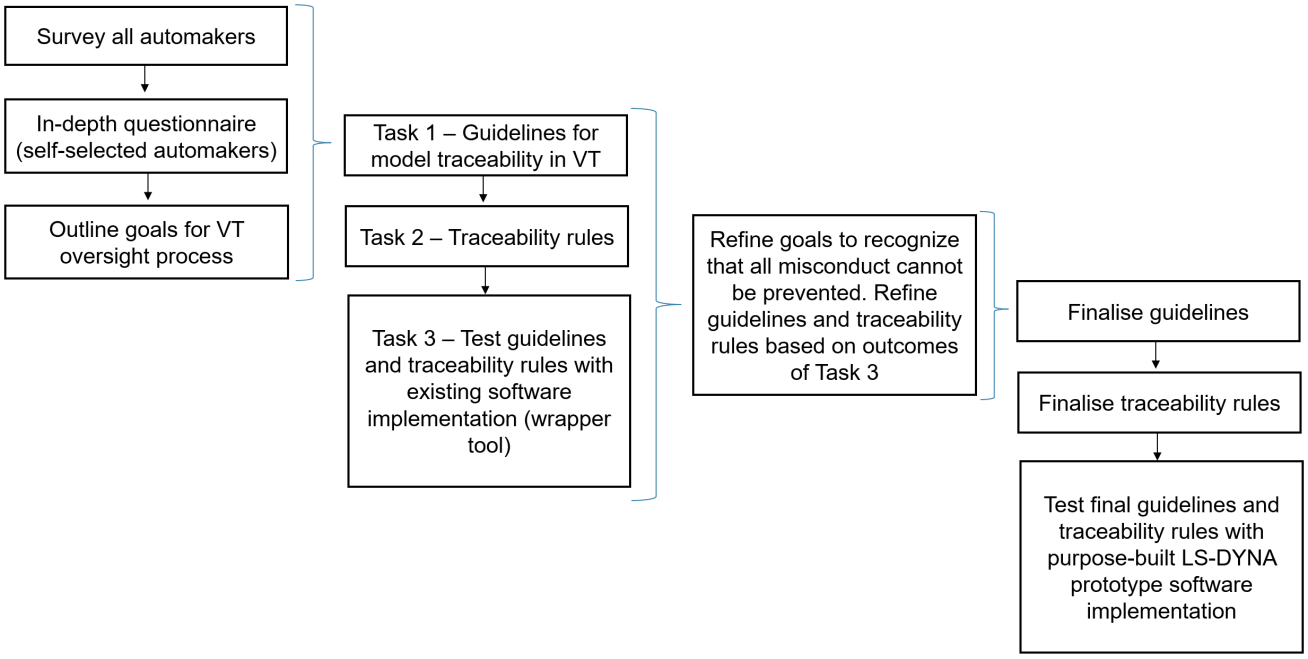

Fig. 1 Workflow of methods for developing a model traceability procedure in virtual testing framework

Automaker Survey

As part of a larger survey on computational modeling experience and best practices, automakers were queried for their feedback on their experience and tools for using simulations in occupant safety evaluations. Questions included:

- Does your organisation use computational modeling for occupant safety evaluations (e.g., using ATD models and/or human body models)?
- Approximately how many years has your organisation used computational modeling for safety evaluations?
- What solver(s) (e.g., LS-DYNA, MADYMO, Radioss, Pamcrash) do you use?

Aggregated Feedback From Industry Questionnaire and Discussions

Automakers who expressed interest were invited to complete a more in-depth questionnaire on their perspectives on virtual testing in automotive safety assessments. Eight automakers completed the questionnaire and participated in follow-up meetings. From the questionnaire and follow-up meetings, questions and feedback were organised into the following categories:

- Overall challenges with virtual testing in safety assessments
- Automaker recommendations on the key components for a successful virtual testing framework
- Given the option to perform an evaluation either physically or virtually, factors for internal consideration when making that decision
- Perspectives on a procedure that automatically reports the results with no or minimal user interaction
- Benefits and hurdles for assessment by a third party or a ratings agency in virtual testing (i.e., where simulations are performed by a third party or ratings agency and not the automaker)

- Benefits and hurdles for virtual testing programmes with automaker run simulations (i.e., where the simulations are performed by the automaker and not the rating agency).

Development, Testing and Refining the Model Traceability Procedure

Development

From the feedback gathered, initial goals were outlined to identify the minimum amount of information and oversight mechanisms necessary to ensure that results from an automaker's simulations come from trusted models, where trusted models meet the following requirements:

- the FE model passed specified validation cases
- the model was not manipulated to artificially alter its response
- the assessment and validation simulations were conducted with the same, unaltered model.

In parallel, goals were defined to identify the methods and tools needed for fingerprinting to trace models from validation to assessment and to investigate tools to check inputs and outputs from the simulations for proper methods and to control changes.

Three tasks were outlined to achieve these goals:

1. develop high-level guidelines describing what model traceability should accomplish
2. outline content for traceability rules
3. outline software performance requirements for functionality.

Stress Testing the Wrapper Tool

As a first stress-testing exercise, the Python wrapper tool presented in [12] was tested. The wrapper runs the LS-DYNA simulation and attempts to read all include files (plain text files with ASCII characters), creates a hash [13] of each include file and, under the normal termination condition, reads LS-DYNA binout files, extracts hashed output .csv files and writes hashes to a cryptographically signed .pdf file. Hashing is one type of fingerprinting process that transforms data to a fixed length, typically alphanumeric, string as a way to quickly confirm that file contents have not changed [13]. Cryptographic signatures provide a method for verifying the source of a file and that the file has not been altered [14]. To explore solutions for model traceability, this project exercised this tool to identify strengths and weaknesses in the methods. Eight scenarios were explored to test the functionality of the tools against developed guidelines and rules for traceability, including:

- Modifying input files and methods
    - Check if include files part of *INCLUDE_RELATIVE_PATH misidentify inputs
    - Check if include files defined under the *COMMENT section lead to incorrect include hashing
    - Check if unidentified restart files lead to unreported input hashes
    - Check if multiline inputs (i.e., adding the "+" sign at the end of the line) are hashed properly
    - Check if multiple calls of the same include can be used with *INCLUDE_TRANSFORM
- Modifying output files and methods
    - Check if code can identify changes in the binout file names when using the *CASE keyword
- Modifying execution of the script
    - Check if code can be modified to replace simulation outputs after termination, but before triggering postprocessing
    - Check if adding "Normal termination" to the file triggers the postprocessing routine before solver termination.

Refining

Following stress testing, an inspection of existing virtual testing protocols helped identify further potential for vulnerabilities in the virtual testing process and outline feasibility requirements for the model traceability procedure. From this inspection, the goals were refined so that instead of aiming to eliminate all potential vulnerabilities in the virtual testing process, which could prove impossible, they aimed to raise the threshold for vulnerabilities. Additionally, requirements were identified to address the need to control components of the simulations that impact model validation (e.g., model material and element definitions) between validation and

assessment load cases, while allowing others like dummy nodal coordinates (for different occupant positions) and crash pulse to change.

Model Traceability Procedure

From this work, a procedure for model traceability was developed that allows vehicle safety assessment agencies to trust results from virtual tests in a workflow that is feasible for automakers and their suppliers. The procedure allows simulations to be run by the automaker while relying on the FE solver software to track, trace and report simulation inputs, outputs and content fingerprints using secure methods that cannot be modified before submission to the assessment agency. To support the implementation of this functionality across the multiple FE solvers used in the automotive industry, platform-agnostic requirements for the FE-solver functionality were outlined.

Stress Testing the LS-DYNA Prototype

To implement and test the refined requirements, functionality was added to a LS-DYNA development version (version DEV-113612 and newer). This functionality allowed the solver to

- allow custom definition of inputs (keyword files) as locked (should not change) or open (appropriate changes allowed),
- calculate hash of locked content,
- calculate hash of output files,
- lock and sign a report containing these fingerprints,
- provide a report containing an overview of open content, and
- supply a public key allowing external parties to view, but not edit, the report.

This functionality was stress tested with both encrypted and unencrypted FE human surrogates and vehicle environments. In rear-impact simulations, a series of isolated cases were analysed from the hashing of determined parts of the inputs to the appropriate extraction of results. The Open VT seat and the DYNAmore FE version of the BioRID v4.0 were evaluated in a simulated rear-impact scenario using the following scenarios:

- hashing of individual and combined keywords
- hashing of full include files
- hashing of encrypted portions of the models
- ensuring the change of reported hashes between different load cases
- extracting information from simulation output files
- checking that the reported hashes are the same as the ones obtained from an external tool
- checking GNU Privacy Guard (GPG) [15] signature (encrypted digital signature using open-source GnuPG encryption) to ensure file origin
- tracking hashing changes from validation to assessment scenarios.

This functionality was also stress tested with the WorldSID v8.0 FE ATD from DYNAmore alone, with the LSTC WorldSID FE ATD in a far-side sled simulation and with the THUMS AM50 V4.0.2 HBM in a full-overlap frontal simulation. With these simulations, the LS-DYNA functionality was tested to confirm whether it could

- hash different parts of the model structure (e.g., materials, damping),
- return identical hashes for locked portions,
- report different hashes when the comment section is modified,
- work with a preprocessor (PRIMER v21.1) and not result in incorrect definitions of the hashing portions, and
- ensure that the appropriate hashes change when modifying the inputs, including crash pulse, seat, dummy and seat belt.

III. RESULTS

Automaker Survey

Fourteen automakers responded to the survey on computational modeling experience and best practices. Of the fourteen, 13 indicated that they used computational modeling for occupant safety evaluations, ranging in

experience from 2 to 40 years. Eleven automakers reported having experience with LS-DYNA, seven reported experience with MADYMO, four reported experience with Pamcrash and one reported experience with Radiosss.

Aggregated Feedback From Industry Questionnaire and Discussions

Eight automakers responded to the in-depth questionnaire and participated in follow-up meetings to communicate their perspectives on virtual testing in automotive safety assessments.

Automakers communicated that the overall challenges with virtual testing in safety assessments are the protection of intellectual property while sharing the necessary information with the assessment agency and expectations from the assessment agency on the readiness of computational modeling tools for virtual testing. Additionally, several automakers felt strongly that when simulations are conducted by a third party or the assessment agency, the modeling practices that make the models the most reliable would not be transferred with the vehicle model, limiting the accuracy of the simulation outputs and leading to possible use outside of its design constraints. Though automakers acknowledged that third party or assessment-agency run simulations benefit from standardisation by reducing errors, they proposed simulations run by automakers as the most feasible path forward, with appropriate transparency of model development and validation and oversight of the process to prevent misconduct.

When choosing whether to perform physical or virtual evaluations, automakers responded that both methods have value, depending on the test purpose and conditions and part availability. Physical tests provide more certainty, but virtual tests can include more conditions. When queried for their perspectives on a procedure that automatically reports the results with no or minimal user interaction, automakers responded that it would be challenging to develop a tool that could manage the differences in automaker workflow while having complete confidence that models that have been validated are also used for assessment.

Feedback on the potential hurdles for a virtual testing programme where simulation are run by automakers included variations in model accuracy among the automakers, the influence of software differences on model accuracy, trust in simulation, and development of a sufficient oversight mechanism to trust simulations.

The feedback from industry oriented this work towards a virtual testing framework where simulations are run by automakers with sufficient oversight to ensure that assessment agencies can trust that simulation results are representative of physical reality and real-world outcomes.

Development, Testing and Refining the Model Traceability Procedure

Development

Feedback from industry and the computational modeling community helped develop high-level guidelines describing what a procedure for model traceability in virtual testing should accomplish. The procedure should:

1. Confirm that the information reported to the assessment agency comes from the inputs and outputs run by the solver.
2. Confirm that the simulation outputs come from the simulation inputs.
3. Confirm that no inappropriate changes have been made between validation and assessment load cases in the simulation.
4. Accommodate necessary changes to the simulation inputs between validation and assessment load cases (e.g., nodal coordinates for a different seat position).
5. Protect the intellectual property of automakers and suppliers.

Additionally, it was identified that the rules and software used in the model traceability procedure be technology and platform agnostic and freely available for anyone to implement or commercialize. The assessment agency must also be able to freely view the report, confirming model traceability without requiring a software license.

Stress Testing the Wrapper Tool

Stress testing of the Python wrapper tool presented in [12] included trialing the wrapper in several simple scenarios with varied syntax and structure, seeking to identify the strengths and weaknesses of the hash-generation wrapper (Table I). The tool achieved the aim of being able to track model changes automatically by hashing individual include files. Several of the test scenarios, however, showed that implementing this functionality external to the FE solver introduced vulnerabilities like not recognising the use of a restart file to initialise the simulation, source code modification and substitution of solver output files. These vulnerabilities undermine the value of the cryptographically signed report because they allow tampering that is not identified prior to the initialisation of the oversight process and generation of the report. This exercise also highlighted the sensitivity of solver syntax and the potential burden for third-party tools to update their code to properly interpret solver syntax.

TABLE I
SUMMARY OF PYTHON WRAPPER TOOL TEST SCENARIOS AND OUTCOMES

Test category	Method	Result
Modifying input files and methods	Check if include files part of *INCLUDE_RELATIVE_PATH misidentify inputs	Fail, defining include files as part of *INCLUDE_RELATIVE_PATH result in misidentification of inputs
	Check if include files defined under the *COMMENT section lead to incorrect include hashing	Fail, include files defined under the *COMMENT section as actual input includes result in incorrect include hashing
	Check if unidentified restart files lead to unreported input hashes	Fail, unidentified restart files result in unreported hashes
	Check if multiline inputs (i.e., adding the "+" sign at the end of the line) are hashed properly	Fail, multiline inputs were not hashed properly
	Check if multiple calls of the same include can be used with *INCLUDE_TRANSFORM	Fail, the tool did not allow for multiple calls of the same include
Modifying output files and methods	Check if tool can identify changes in the binout file names when using the *CASE keyword	Fail, the tool could not identify changes in binout file names when *CASE keyword is used
Modifying execution of the script	Check if tool can be modified to replace simulation outputs after termination, but before triggering postprocessing	Fail, code was able to be inserted in the tool that could replace simulation outputs before triggering post processing
	Check if adding "Normal termination" to the file triggers the postprocessing routine before solver termination	Fail, adding "Normal termination" to the tool's code was successful in triggering post processing before the actual termination of the solver

Refining

Stress testing and an inspection of existing virtual testing protocols identified that simulations run by automakers without appropriate oversight will be vulnerable to the following: different model versions being used between validation and assessment, results being reported that don't come from the validated models, and intentionally deceptive features being added to circumvent model-tracking tools and secure reports. Some of these vulnerabilities also increase the potential for mistakes, such as poor version control leading to inadvertently mixing up models or mislabeling results. Many of these vulnerabilities and avenues for mistakes can be mitigated with tools that can track and trace model changes and link the inputs to the outputs. Since it would be impossible

to prevent all vulnerabilities, the goals for the model traceability procedure were refined to raise the level of intent that would be needed to exploit a vulnerability from low to high. Vulnerabilities with a low level of intent are those that could conceivably be a simple oversight or mistake (such as poor version control leading to a mixing up of model files). Vulnerabilities with a high level of intent are those where an error could not conceivably be a simple mistake (such as intentionally altering the fingerprint of a file or including features that have no legitimate modeling purpose). Table II provides several examples of identified vulnerabilities. Through this procedure, we seek to provide tools and processes to help avoid legitimate mistakes and easy-access tampering or manipulation to improve confidence in simulation results through the implementation of secure and verifiable documentation and model tracing.

TABLE II
VULNERABILITY EXAMPLES

Vulnerability	Explanation	Example
Outputs modified before reporting	The manual postprocess of the simulation data allows the users to mistakenly or intentionally modify the outputs before reporting	Postprocessing step executed on the wrong simulation data Report manipulated before being sent to the assessment agency
The reported results did not come from the simulation outputs	The manual postprocess allows for the reported data to be generated outside of the solver	Test data is used to generate the reported results
Inputs inconsistent with the load case	The inputs do not represent the load case conditions	Wrong pulse is used in the simulation
Inconsistent input changes between load cases	The input is modified beyond what's strictly needed to represent the changes in load case	Global damping is selectively added to some load cases

Model Traceability Procedure

After goal refining and testing, model traceability requirements were categorised into four groups: model fingerprinting, automatic linking of models to simulation results, file security protocols and model interrogation.

Model Fingerprinting
The procedure requires that a fingerprint or a unique identifier (e.g., checksum, hash) be reported for the whole model (or for the compilation of files that comprise the model) to link the model to reported outputs. The model must also be able to be parsed into sections that can be either locked or open. Locked portions must not change between validation and assessment, and there must be a unique identifier reported for each locked portion. Open portions are those portions of the models that must be allowed to change between validation (e.g., audit) and assessment. Open portions need to be accessible by the assessment agency for inspection and may possibly need to be hashed for portions that must be open to the assessment agency but include encryption, like a dummy model.

Automatic Linking of Models to Simulation Results
The procedure needs to provide a secure digital link between locked portions of the model, open portions of the model, and the results, which outputs to a secure summary file and executes without intermediary human interaction. The assessment agency should also have access to software that can ensure that the outputs come from the reported simulation by checking a unique identifier (e.g., hash) from digital outputs of the solver against those provided by the automaker. This functionality should also allow the assessment agency to check that locked portions of the model have not changed from validation to assessment (by comparing the model identifiers).

File Security Protocols

In the model traceability procedure, model fingerprinting, results fingerprinting and summary file security protocols must be accomplished with publicly available protocols. Therefore, the tools must not require proprietary encryption protocols or software for the assessment agency to archive and view results from the reported validation and assessment. The requirements for this procedure also need to be technology and platform agnostic, using publicly available tools to verify traceability. These requirements should also be freely available to implement and commercialize.

Model Interrogation

The procedure should allow the assessment agency sufficient access to the validation and assessment simulations for a basic check of test conditions including 3D data, energy results, results used for assessment and other solver output files. In addition, the assessment agency should have enough visibility of the model and simulation to examine for consistency and intentionally deceptive actions, including lists of keywords included in the open and locked portions of the model and control specifications used for validation and assessment simulations.

Table III outlines the model traceability procedure to achieve these requirements in a virtual testing framework where simulations are run by automakers; it provides the detailed steps and tools used by the automaker and assessment agency. Fig. 2 provides a high-level version of this same procedure.

TABLE III
MODEL TRACEABILITY PROCEDURE

Party	**Tool**	**Function**
Automaker	FE Solver	1. Identifies inputs (keyword files) as locked (should not change) or open (appropriate changes allowed) 2. Calculates fingerprint of locked content 3. Calculates fingerprint of output files 4. Locks and signs a report containing these fingerprints 5. Provides a report containing an overview of open content 6. Supplies a public key allowing external parties view-only access to the report
	Assessment agency upload portal	1. Provides the assessment agency the following: simulation results, a keyword summary of open content and a locked report containing fingerprints for locked content, output files, open content and all files
Assessment agency	Assessment agency tool (cryptography methods matching the FE solver)	1. Calculates (or acquires from a third-party) fingerprints for the output files and the input files of qualified parts (e.g., dummy) 2. Compares calculated fingerprints to those provided in the locked report 3. Reviews the keyword summary to confirm that changes are appropriate. 4. Confirms consistent model use across simulations by comparing the fingerprints for validation and assessment

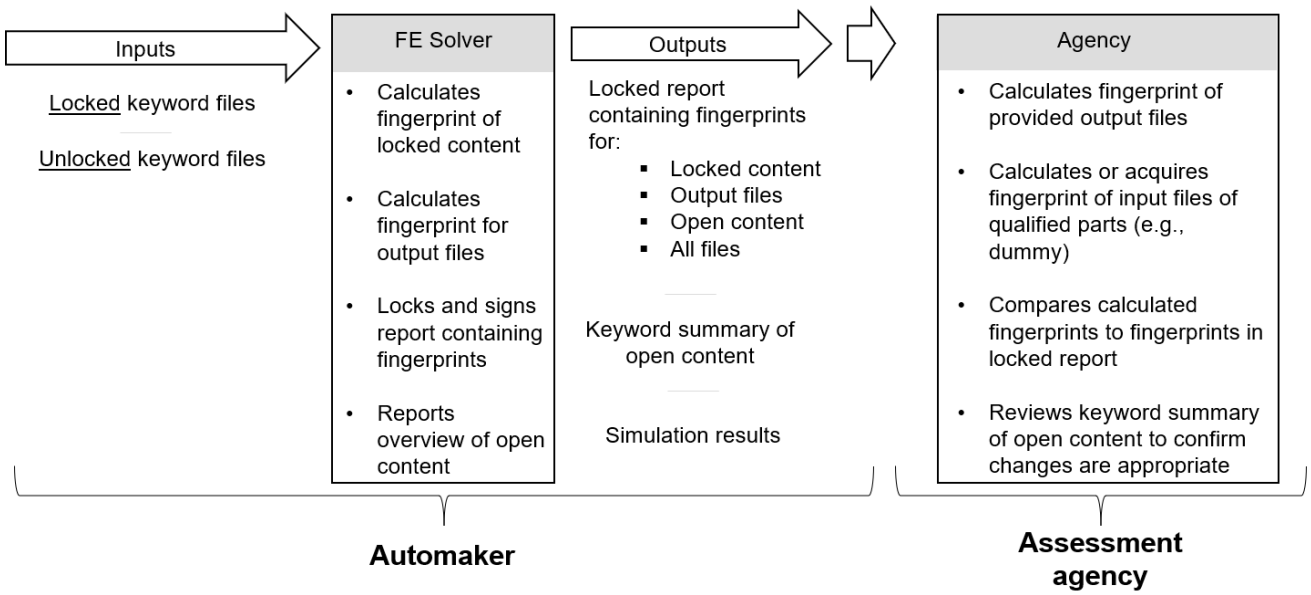

Fig. 2 Workflow for model traceability procedure in virtual testing framework where simulations are run by automakers

Stress Testing the LS-DYNA Prototype

The wrapper tool introduced in [12] generates hashes of simulation input files and then links them with hashed output files in a signed report. However, since the tool was external to the FE solver, it was unable to monitor what happened in the solver between the inputs and outputs, which allowed vulnerabilities like not recognising the use of a restart file to initialize the simulation and allowing changes to the outputs prior to hashing. To address these vulnerabilities, ANSYS added a new feature to a LS-DYNA development version to support fingerprinting of input and output data. Solver generation of the fingerprints and report significantly reduces the opportunities for manipulation because the solver process and source code are closed to the user and the solver company is a trustworthy source for the digital signature of the report. The feature is designed to work in current computational-modeling production environments and with existing workflows. It allows the user to specify "locked" portions of an input through the newly introduced *HASH and *HASH_END keywords, which act as brackets to mark the blocks for which LS-DYNA creates a hash. A hash (unique identifier) is created for the content between the defined brackets. Any change of a "locked" input section results in a different hash identifier.

Once a simulation is finished, LS-DYNA writes the hash identifiers of the "locked" input blocks and the output files (e.g. binouts, d3plots, etc.) at the end of the d3hsp file. Finally, to protect the hash identifiers, the d3hsp file is hashed and its identifier is encrypted (digital signature) using asymmetrical public-key cryptography. With asymmetrical public-key cryptography, the sender (LS-DYNA) locks the file with a private key and the recipient (automaker and assessment agency) can open and view, but not edit, the files with a public key [14]. For hashing, LS-DYNA uses the open-source sha256 hash algorithm [13] and, for the encrypted digital signature, it uses open-source GnuPG (GPG) [15] encryption, which allows third parties to verify the hash identifiers and the digital signature independently.

This tool was stress tested using both encrypted and unencrypted FE human surrogate models BioRID, WorldSID and THUMS. With the BioRID FE model and the Open VT seat, it was confirmed that the tool can hash individual and combined keywords, full include files and encrypted portions of the models. It was also confirmed that different hashes would be reported if something was changed between load cases. The hashes were able to be successfully extracted from the binout and d3hsp files and confirmed with an external tool. The GPG signature was confirmed to originate from the d3hsp file. Note that due to the hash nature of the solution, the smallest difference in the inputs will result in a mismatch of the resulting hashes.

With the DYNAmore WorldSID ATD, LST WorldSID ATD and THUMS AM50 v4.0.2 HBM, the tool worked as

expected where any modification to the inputs, including changes to the pulse, comments, seat, human surrogate or seat belt were reflected in a modification to the final hashes. As expected, the exercise found that preprocessors are currently not compatible with the hashing feature.

The wrapper introduced by [12] and the LS-DYNA integrated tool have some similarities in functionality, but the LS-DYNA tools increases the security of the process to address vulnerabilities found when stress testing the external wrapper. Table IV provides a summary of the functional requirements for model traceability in an automotive virtual testing framework, the methods for implementation and details on how each prototype software solution executes this feature.

TABLE IV
FUNCTIONAL REQUIREMENTS FOR MODEL TRACEABILITY AND SOFTWARE TOOL IMPLEMENTATION

Functional requirements for virtual testing traceability	**Feature**	**Galijatovic [12] wrapper**	**LS-DYNA integrated tool**
1. Confirmation that the information reported to the assessment agency came from the inputs and outputs run by the solver	Secure reporting	View-only report	View-only report
		Signed with private key of the user or the hardware	Signed with private key of LS-DYNA, where authenticity can be proven with a public key
		Generated by the wrapper	Generated by the solver to minimise opportunities for manual manipulation
		Tool is external software which could be manipulated	Source code for tool is inaccessible and not editable (integrated into LS-DYNA)
2. Confirmation that simulation outputs come from the simulation inputs	Fingerprint inputs	Input read and hashed by wrapper independent of LS-DYNA	Input read by LS-DYNA and directly hashed
	Fingerprint outputs	Independent output hashing outside of the solver after termination	Solver level output hashing at simulation completion before termination
	Secure reporting	Vulnerable transitions from one tool to the other	Explicit link between input and output reporting within the solver
3. Confirmation that no inappropriate changes have been made between validation and assessment load cases in simulation	Fingerprint inputs	Hashes generated for all inputs	Hashes generated for all inputs
	Secure reporting	Contains hashes for the inputs	Contains hashes for the inputs and keyword summary content

	Keyword summary of open content	Open include files need to be shared together with the report	Reported in d3hsp
4. Accommodation for necessary changes to the models between validation and assessment (e.g., nodal coordinates for a different seat position)	Fingerprint structure	Hashed each include file, which requires splitting files up into frozen and load-case dependent files	Hashes can be generated for portions of include files for frozen and open content, using a bracket scheme independent of file structure
	Fingerprint summary of open content	Hashes open include files	Hashes open content with segmentation
	Secure reporting	Contains hashes for the inputs	Contains hashes for the inputs and keyword summary content
	Keyword summary of open content	Open include files need to be shared together with the report	Reported in d3hsp
5. Protect the intellectual property of automakers and suppliers	Fingerprinting	Critical building block for secure reporting	Critical building block for secure reporting
	Secure reporting	Allows the process to move forward without the need to access input files	Allows the process to move forward without the need to access input files
	Keyword summary of open content	Open include files need to be shared together with the report	Keyword summary allows access to critical aspect of the inputs without sharing IP

IV. DISCUSSION

Though other vehicle safety agencies are considering virtual frameworks where the automaker shares the encrypted vehicle model or the agency develops a parametrizable generic model, the virtual testing framework proposed here allows the automakers to run their own safety simulations to be used for assessment. This runs counter to how safety assessment organisations have gained consumer trust in the past, through independent evaluation of vehicle safety. Automakers have used computational modeling in designing for occupant safety for decades, throughout which they have built considerable expertise in the use of sophisticated models to predict safety outcomes for their vehicles. Leveraging automaker knowledge and sophisticated vehicle models with the advanced human injury prediction of human body models offers the opportunity to advance safety, but requires a virtual testing framework that respects automaker privacy and model integrity while also being trustworthy, effective and practical. Conversations with automakers and their suppliers revealed that the computational models they build for vehicles and vehicle components contain proprietary information, like material characteristics and modeling methods, in which automakers have heavily invested to make their models as predictive as possible. Further, proper use of these models lies in institutional knowledge that requires company-specific training. Together, these reasons make sharing models with an assessment agency or a third party prohibitive and prone to error. Maintaining consumer trust with a virtual testing framework that allows

automakers to run their own simulations requires accountability. When the assessments can be checked with physical tests, random auditing by the assessment agency can provide accountability. However, when the assessment matrix includes tests that cannot be physically audited, like those with human body models, two key components are necessary for accountability — physical validation tests to confirm the predictive abilities of the automaker models and an oversight mechanism to ensure that no changes are made to the verified models that would alter their predictive abilities.

This work developed a model traceability procedure as an oversight mechanism that securely reports fingerprints of models and simulation output files to the assessment agency to provide confirmation that models have not changed between validation and assessment load cases and to securely link the models to the outputs. FE analysis computational modeling is a complicated process involving many software tools including model assemblers, preprocessors, FE solvers, postprocessors and output file generators. Stress testing of model traceability tools external to the FE solver revealed significant vulnerabilities, showing that an oversight mechanism within the FE solver provided the most secure solution. The prototype software tool developed by ANSYS for LS-DYNA operates within the solver and provides hashes for the model components that can be securely transferred to the assessment agency, preventing mistakes or intentional editing. The primary advantage of a solver-based solution is the heightened level of security in reporting. While the Python wrapper tool [12] was able to generate hashes for both the inputs and outputs and report them in a view-only file, the opportunities for manipulation were significant. When the hashes are generated by the solver, as in the LS-DYNA tool, the opportunities for manual manipulation are minimised, the source code cannot be edited to alter the simulation inputs or outputs and there is oversight of the process between the input and output stages. A solution native to the solver also avoids the overhead required for a third-party solution to maintain accurate syntax and functionality to properly interact with the solver.

The LS-DYNA prototype was stress tested with both ATD and HBM FE models and showed that the procedure works with both types of human surrogates and accommodates parsing the simulation components into locked and unlocked portions. However, stress testing of the LS-DYNA prototype tool identified some items for improvement, including the ability to customise the data contained in the input information summary (currently the whole d3hsp is hashed and reported for this function) and the ability to report the hashes in an independent text file (currently included at the end of the d3hsp). Additionally, subsequent feedback from automakers requested functionality in the tools to pare down the data shared with the assessment agency in the output files to only the information required by the assessment agency. Further improvements could include eliminating hash variability due to changes in comments. Alternate tools that build on the ANSYS functionality and solver-based hashing could also provide additional solutions for model traceability and may be explored in the future.

Though the test cases for this work focused on ensuring that that models have not changed between validation and assessment load cases and are securely linked to the outputs for consumer information programmes, the tools have broader application in the field of automotive safety assessments. The procedure and tools could also be used to ensure that the dummy and HBM do not change after being qualified or certified. They could also be used to expand the scope of physical tests in type approval or regulation by providing the necessary oversight to allow modeling to confirm compliance of all vehicle variants.

This model traceability procedure was developed so that similar workflows could be emulated by other solvers used in industry like VPS and Radioss. The procedure with the LS-DYNA prototype provides a feasible workflow for automakers, including functionality that accommodates appropriate changes. For example, the automaker’s model can be validated with the physical test of a dummy and the framework allows for changes to the surrogate so that the validation can be applied to human body models. It also executes the oversight process within software already used in the automaker's simulation process. The exercises in this study made it clear that preventing all vulnerabilities in the virtual testing process was not practical. Instead of aiming to eliminate all vulnerabilities, this procedure was developed to raise the threshold for vulnerability so that mistakes and bad practice would not go unnoticed and intentional misconduct would require significant effort and a clear intent to deceive the assessment agency.

V. CONCLUSION

After several iterations of stress testing and refining, a procedure for model traceability in virtual testing was developed where simulations are run by the automakers and the FE solver is employed to confirm that the information reported to the assessment agency comes from the inputs and outputs run by the solver, the simulation outputs come from the simulation inputs, and no inappropriate changes have been made between validation and assessment load cases in the simulation, all while accommodating necessary changes and protecting the intellectual property of automakers and suppliers. By verifying that no inappropriate changes have been made to models and outputs while protecting automaker intellectual property, this procedure provides a feasible path forward for assessment agencies to encourage more robust safety solutions by broadening the scope of assessments. While this procedure cannot prevent all intentional misconduct, it raises the level of intentionality required to manipulate the models, reducing the risk of changes being undetected and increasing the trustworthiness of virtual testing. Expanding vehicle assessments with virtual testing is an important next step in addressing occupant safety for all occupants, and this model traceability procedure enables the inclusion of simulations that cannot be directly validated, like those with human body models.

VI. ACKNOWLEDGEMENTS

The work presented here was developed as a collaboration of IIHS, the University of Virginia Center for Applied Biomechanics, HIC Research, Graz University of Technology, JSOL and ANSYS. IIHS provided funding for portions of this work. The authors would like to thank DYNAmore, an ANSYS company, for the use of the BioRID FE model.

VII. REFERENCES

[1] Mackay, M. Engineering in accidents: vehicle design and injuries. *Injury*, 1994. 25(9): p. 615–621

[2] Kahane, C.J. Effectiveness of pretensioner and load limiters for enhancing fatility reduction by seat belts. 2013, National Highway Traffic Safety Administration: Washington, DC.

[3] Insurance Institute for Highway Safety. "Fatality Facts 2022: Yearly snapshot" Internet https://www.iihs.org/topics/fatality-statistics/detail/yearly-snapshot. [cited 2025 March 21].

[4] Forman, J., Poplin, G.S., et al. Automobile injury trends in the contemporary fleet: Belted occupants in frontal collisions. *Traffic Injury Prevention*, 2019. 20(6): p. 607-612

[5] Insurance Institute for Highway Safety. Major change in frontal crashworthiness evaluations. *Status Report*, 2006. 41(3): p. 1–7

[6] Huang, Y., King, A.I., and Cavanaugh, J.M. Finite element modeling of gross motion of human cadavers in side impact. *SAE Transactions*, 1994: p. 1604–1622

[7] Matsuda, T., Kobayashi, N., Fujita, N., and Kitagawa, Y. Development of a human body model (THUMS Version 7) to simulate kinematics and injuries of reclined occupants in frontal collisions. *Proceedings of 27th ESV Conference*, 2023. Yokohama, Japan

[8] Gayzik, F.S., Moreno, D.P., Vavalle, N.A., Rhyne, A.C., and Stitzel, J.D. Development of a full human body finite element model for blunt injury prediction utilizing a multi-modality medical imaging protocol. *Proceedings of 12th International LS-DYNA Users Conference*, 2012. Detroit, MI

[9] Pipkorn, B., Östh, J., et al. Validation of the SAFER human body model kinematics in far-side impacts. *Proceedings of IRCOBI Conference*, 2021. Online

[10] Klug, C., Feist, F., et al. Development of a certification procedure for numerical pedestrian models. *Proceedings of 26th ESV Conference*, 2019. Eindhoven, The Netherlands

[11] Klug, C., Schachner, M., et al. Euro NCAP virtual testing-crashworthiness. *Proceedings of 27th ESV Conference*, 2023. Yokohama, Japan

[12] Galijatovic, E., Eichlseder, M., Heindl, S.F., and Klug, C. Integrity of virtual testing for crash protection. *Frontiers in Future Transportation*, 2022. 3

[13] Veness, C. "Movable Type Scripts SHA-256 Crytographic Hash Algorithm" Internet https://www.movable-type.co.uk/scripts/sha256.html. [cited 2025 March 12].

[14] Diffie, W. and Hellman, M.E., "Democratizing Cryptography: The Work of Whitfield Diffie and Martin Hellman", pages 365-390, 2022

[15] The GnuPG Project. "The GNU Privacy Guard" Internet https://gnupg.org/. [cited 2025 March 12].