Advanced AI driven Exploration Possibilities to Link Model Changes and Effects

Dr. D. Steffes-lai, Prof. Dr. J. Garcke, Dr. R. Iza-Teran, T. Klein, M. Pathare
(Fraunhofer SCAI, Germany);

G. N. Devdikar
(Stellantis-Opel, Germany)

Abstract

The rapid advancement of artificial intelligence (AI) presents significant opportunities for the field of computer-aided engineering (CAE), particularly in data management and engineering design processes. This research work explores AI-driven techniques for visualizing and analysing CAE model evolution, emphasizing methods for intelligent data retrieval and inference. We introduce enhancements to our Laplace-Beltrami Shape Feature Approach (LBSFA), enabling engineers to assess deformations and mesh function changes more efficiently. Additionally, we examine the application of Retrieval-Augmented Generation (RAG) with structured data representations to enhance exploration and preselection of key components. Finally, we discuss the potential of AI-driven systems to learn measure-effect relationships, optimizing CAE data analysis and decision-making.

1. Introduction

The CAE field is facing enormous innovative opportunities offered by the rapid development of artificial intelligence (AI), particularly in text and image processing, if the unique challenges of complex CAE data are addressed. CAE engineers foresee AI as a potential tool not only for data management but also for enhancing complex engineering design processes across development stages. This work examines how AI could transform CAE with novel data representation and exploration techniques. We focus on AI-driven methods for visualizing and exploring model evolution and linking it to resulting effects, enabling inference and intelligent data retrieval. Implementing AI effectively in CAE requires transparent learning algorithms capable of handling limited but complex data, like mesh functions and geometries. Furthermore, effective product development relies on seamless integration between model changes and their effects in analysis and optimization. However, many tools used for design and simulation operate as standalone systems, limiting collaboration and efficiency [1,2,3]. While some analysis tools include built-in modeling capabilities, designers and engineers often rely on separate software for creating models and evaluating their performance [4,5]. Additionally, many modeling systems offer support for automated optimization, requiring iterative design-evaluate-redesign loops but are expensive and difficult to integrate in real product development workflows

[6]. This adaptation is time-consuming, prone to errors, and becomes increasingly complex as the number of variables grows. Despite technological advancements, industries such as automotive, aerospace, and manufacturing still primarily depend on expert- and simulation-based product design and optimization methods, lacking fully integrated solutions that link model updates with performance analysis.

Automated design optimization, where model changes are seamlessly connected to evaluation tools, is essential for improving product quality and accelerating development. While commercial design and simulation software have made progress in integration [6], most solutions remain constrained within proprietary ecosystems and require costly upgrades. An alternative is to link model changes with analysis and optimization tools using common scripting and programming languages, reducing time and costs. Some organizations have developed in-house solutions for optimizing product designs, yet a standardized framework for integrating model changes into the optimization process remains underexplored.

To address these challenges, an AI framework integrating model changes with evaluation and exploration of results is proposed. By leveraging dimension reduction in a RAG framework for model changes and simulations, this approach streamlines complex product development and optimization processes, making them more efficient and accessible for designers and engineers.

The outline of this article is as follows. First, we present the latest extensions of our Laplace-Beltrami shape feature approach (LBSFA) [7] to investigate similarities or exceptions in deformations and mesh functions, called events, in many simulations. The combined representation of model changes in geometry and input parameters, and results allows to set the model changes into context with the insights from simulation results analysis. This provides a fundamental basis for further post processing of results, like sensitivity and correlation analysis, up to envisioned learning of relationships. Details on the exploration of model changes as well as the pairwise comparison results are given in Section 2. However, the engineer still needs to select mesh functions and components of interest out of a ranked list of components that are influenced by the model changes overviewed in Section 3.a. To address this, we have investigated some more sophisticated filter and grouping methods to ease the decision on which component to look first. On the one hand, a method that filters out all parts without deformations but just influenced due to rigid body motion has proven to be very effective is outlined in Section 3.b. On the other hand, focusing on jumps in a development tree identifies the measures with the highest impact which is presented in Section 4. In this sense, a jump means that the corresponding result jumps from one behavioral mode to another.

Second, we explore how LLMs can be leveraged for CAE data. We present a proof of concept of using a Retrieval-Augmented Generation (RAG) method together with our structured data representation to enable easy exploration of relationships within the data of a development project in Section 5. The RAG approach uses a knowledge base outside the training database and thus offers an extension or specialization in domain-specific knowledge. With this approach we could reach an automatic knowledge driven preselection of important components. Further on, fast metamodels can be generated based on this selection which are an efficient alternative for data-intensive convolutional neural networks (CNNs).

The use case on which we demonstrate our developed methodology is shortly presented in Section 6. Finally, we outline the different modules needed for an AI support system which learns measure-effect relationships in the conclusion in Section 7.

This contribution illustrates how far our AI driven solutions specialized for CAE applications can already reach to support the complex simulation data analysis tasks giving the engineers more resources for interpretation of results and decision making.

2. Comparing two simulation results and detection of model changes

In virtual product development, engineers aim to improve the car behavior with respect to specific crash requirements. To achieve that, they apply design measures with respect to a predecessor design to obtain a successor configuration. The effects of those measures are analysed after the corresponding simulation is computed. For the entire development tree, that is all predecessor and successor simulation pairs within the development process, the full output data, namely displacements and functions on meshes, must be compared. Our approach analyses functions on meshes rather than scalar data or sensor / curve data. More precisely, the impact of design changes can be analysed in terms of data functions either on the nodes (such as displacements or nodal mass) or elements (such as plastic strains, stresses, or failed elements) depending on the analysis objective. For analyzing displacements, rigid body motion can be extracted by setting corresponding anchor points. Thereby, detailed insight in local influences can be evaluated with respect to a certain design change. Several distance measures are available to focus on global or local influences. As basic distance measures the maximum norm, the L2 norm as well as the L1 norm are implemented. To allow the comparison of the difference in a part in relation to all other parts, the metric values are normalized to [0,1]. The maximum norm detects events with emphasis on parts with very high deviations on single nodes / elements between the two models. In contrast the L1 metric sums over the deviations (in elements or nodes) of a part. Thus, applying this metric, events are detected that show deviations on larger areas of the selected part, it puts more emphasis on the entirety of parts. The L2 metric is kind of mixture of both aspects, that is due to the sum of squared deviations single very high deviations are weighted stronger. All differences are stored node- / elementwise for later evaluation.

The comparison results, that is the local events found, are stored partwise in structured JSON files that support further post processing. For each selected part, the part-ID, part-name, metric value, as well as minimum and maximum deviation in the part is stored.

For interactive exploration, we extended the above methodology with some interactive functionality to visualize all node- and elementwise deviations on both models and highlight interesting parts [8]. This simplifies the detection of local hotspots. The whole vehicle or the relevant parts are shown to get an overview of design change impacts.

Design measures include changes in geometry, in material parameters, or in welds. For example, a modification of the B-Pillar might include a geometrical change of a structural member, e.g., a cut or extension, which could require changes in the weld connections or of material thickness. To appropriately deal with all these changes we have developed and implemented a corresponding structured digital representation [9]. Each model can be compared with respect to the respective predecessor or a reference model. A special challenge is the parametrization of geometrical changes to be used with machine learning algorithms, which is one of our ongoing research topics. A simple way to parametrize geometrical changes is by using a mapping to the reference and comparing the differences as a mesh function. To provide an overview of model changes across several iterations, the results of the comparison of many models can be saved in a (graph) database keeping track of model information using parts and their connections [13,14]. Furthermore, the database can be queried easily to summarize changes or to search for specific measures applied. Ongoing work is to further extend the semantics of this representation, including components as groups of parts, and on making it available for filtering and searching (semantic) design measures.

3. Advanced similarity analysis of many simulations results

The Laplace-Beltrami shape feature approach (LBSFA) [7,10] has been used successfully for the compact representation of deformations in car crash simulations [10, 11]. In addition, this Fourier representation has been used as low dimensional features to compute a surrogate model in [12]. The considered geometry in this approach is a surface mesh, so that a Laplace operator can be computed on it with the property of being invariant to isometric deformations. From this operator, its eigenvectors can be computed to derive a basis representation. The mesh as well as all deformations on it can be projected onto the basis to obtain a Fourier decomposition. In addition, also functions on the mesh can be projected to the basis. Especially for functions computing the difference with respect to a reference mesh, a very compact representation can be achieved with only few coefficients [7]. This representation can be reverted to obtain the corresponding function with sufficient accuracy for many applications.

The projection coefficients on the eigenvectors of such a “Fourier-decomposition for geometries” allows a clustering of many simulations based on their different behavioral modes using these “geometry-aware Fourier-modes”. Formally, for any mesh function f of a simulation result corresponding projection coefficients α_j into the spectral basis are computed. This allows to write f as a linear combination of eigenvectors φ_j:

$$f = \sum_{j=1}^{N} \alpha_j \varphi_{j,} \;\; \alpha_j = \langle f, \varphi_j \rangle \qquad (1)$$

where N is a chosen number of coefficients, up to the number of grid nodes. Note that usually only a few coefficients in this surface-aware decomposition are needed to represent relevant variations, resulting in a dimensionality reduction. In the case presented, we use 100 coefficients. The first coefficients with respect to the order of the variance can be used for a 3D representation which captures similarity of many simulations. For details see [7].

This allows an interactive exploration of the results that can be combined with the results of the input model changes. The described approach is implemented in our exploration tool SimExplore.

a. Interactive exploration of model and results

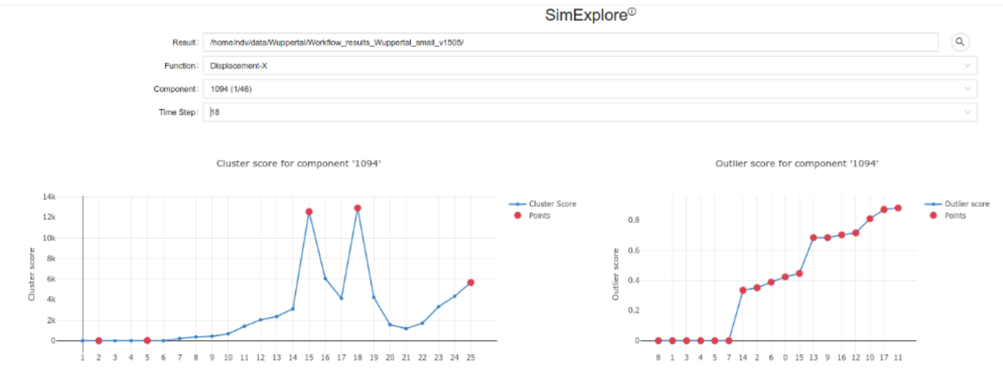

Figure 1: Demonstration of a web-based user interface which visualizes the results of the analysis by interactively highlighting important parts and time steps.

Figures 1 and 2 show the developed web-based user interface for interactively exploring multiple simulation results. For each analysis, the interface guides the user through the selection of interesting components and time steps in order to investigate events in the overall dataset. The function to be analysed can be chosen, which enables the selection of components, sorted by the component score, see Figure 1 (top). The component score ranks each component based on the difference in function values seen between simulations. With this help, the user can select a meaningful component for further investigation. Plots of the cluster score and outlier score are displayed for the selected function and component, see Figure 1. The user can select a time step for further exploration based on the cluster score plot. To guide the selection of a meaningful time step, the cluster score evaluates how clear the simulations are split into clusters and outliers, see Figure 1 (left). To get an overview of how the simulation behaves over all time steps the outlier score plot can be used, Figure 1 (right). It ranks each simulation based on how different the simulation is compared to the simulations which behave similar.

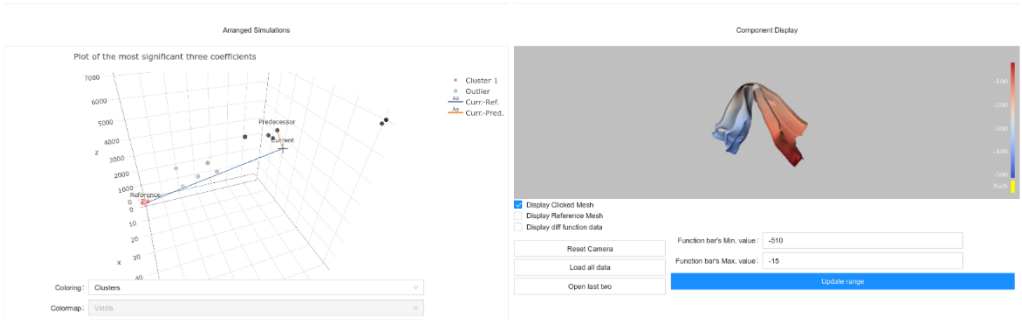

Figure 2: After selecting a part and time step, see Figure 1, an overview of the behavior of the simulations is given by grouping similar simulations into clusters and identifying outliers which behave very differently. Interactively, the corresponding deformation for each part can be visualized by clicking on the embedding point of the simulation. In addition, the corresponding detected model changes can directly be visualized to link the model changes with the resulting effects.

Figure 2 shows a three-dimensional plot where each point reflects a simulation result. If points are closely together, they show similar behavior, while points that are far apart behave differently. By clicking on the points, the user can get a 3D view of the component clicked and can compare it interactively to the other simulation. In addition, the predecessor and successor simulations of the development tree are highlighted and a report of the detected model changes between these two simulations is shown. To see a direct correlation between the position of the simulation in the latent space and the corresponding input changes the user can change the coloring from clusters to any of the input parameters that are varied in the model, such as thickness.

b. Filtering of components

The part identification still faces the issue of classifying real deformations with respect to almost rigid body motions. For that effect some filter capabilities have been investigated. The issues here are parts that do not deform, but which are affected by the rigid body motion. In this case, if follower points are used to subtract this motion, the corresponding parts are highlighted as being changed by our LBSFA algorithms described above. If the user does not filter out such parts beforehand, they will disturb the ranking of the parts. Therefore, a method to detect parts without deformation to filter out such parts has been developed.

The resulting filtering of such parts for our use case example is shown in Figure 3.

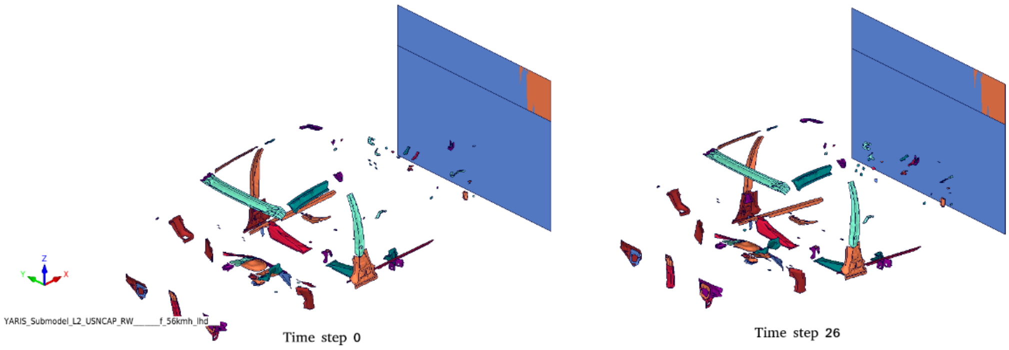

Figure 3: Filtering of components that identify parts not having deformations. It shows parts that mainly have rigid body motion and little to no deformations. Left image is at time step 0 and the right image is at time step 26 (last time step).

An overview of the data analysis pipeline of SimExplore is given in Figure 4.

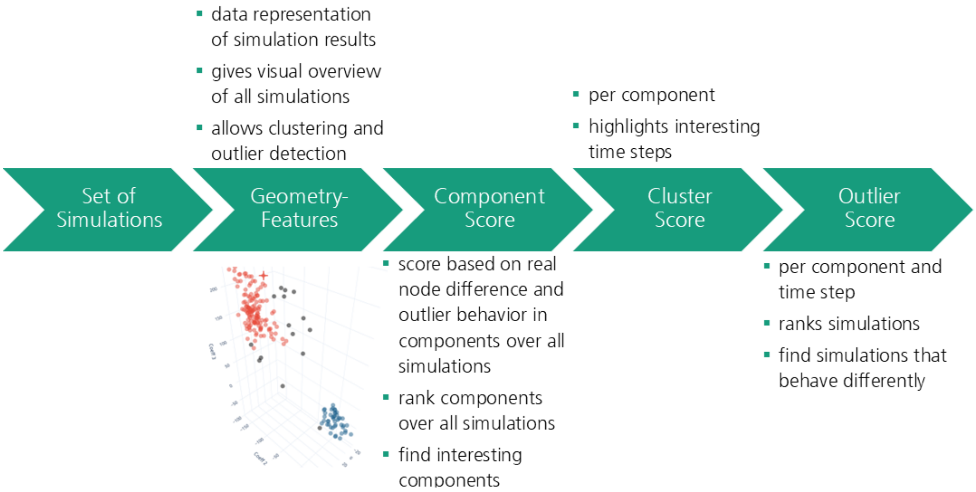

Figure 4: The modules of the analysis pipeline, consisting of data processing and filtering steps.

4. Jump score

In product development, engineers need to apply design measures to obtain a particular outcome respecting safety requirements and constraints. The comparison between a predecessor and successor design pair is effectively achieved e.g. with the methods from [8] but comparing over all simulations in the development tree is still an appealing objective. With this goal in mind, we have developed an innovative Jump score to identify the measures with the highest impact regarding a behavioral change. In this sense, a jump means that the corresponding result jumps from one behavioral mode to another.

The Jump score computation comprises two main steps:

1. Evaluation of a score to get important predecessor–successor pairs
2. Per identified pair computation of PID grouping using clustering

An example visualization of the first computing step is shown in the upper part of Figure 3, the x-axis in this plot represents the predecessor–successor pairs and the y-axis is the score. Selecting one pair from it, allows step 2 as represented in the middle part of Figure 3. For this pair several time dependent curves are shown, that is one for each PID of the car structure with colors representing clusters grouping those parts. Selecting a specific cluster of curves allows the visualization of all the PIDs in the cluster at the maximum time step of the curves which is shown in the lower part of Figure 5.

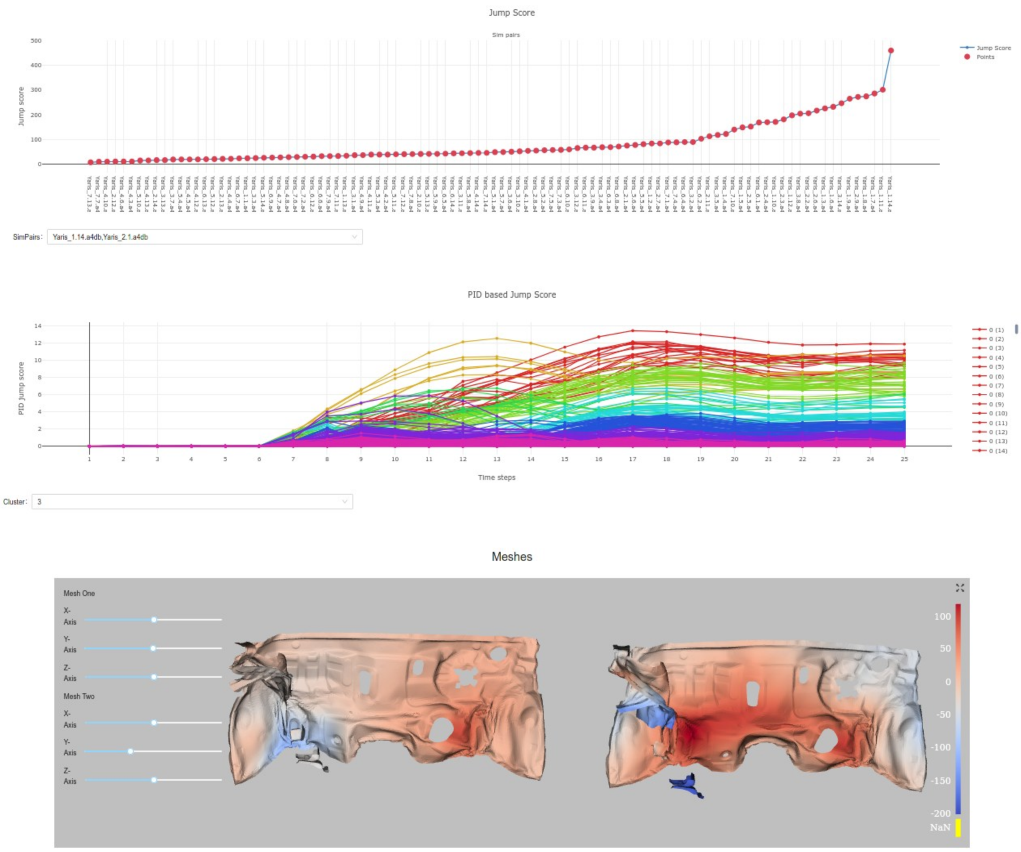

Figure 5:Overview of the web-based UI for the analysis of the Jump score. The first part of the image shows the ranking of simulation pairs due to the computed Jump score, the middle part shows the corresponding curves for the components of a selected simulation pair, while the bottom part gives a detailed overview of the mesh and functions on the selected component.

The workflow for computing and evaluating the Jump score still requires a series of evaluations that requires an adequate complex workflow and a graphical user interface. In this work we benchmark the flexibility of a RAG system to achieve this evaluation process.

The system contains a backend that interfaces with a graph database, but it can also use a file-based storage. A React based web frontend is extended in a modular way that has been developed for the data analysis evaluation of CAE data in [14] so that different ui components can be added as needed. The backend is python based and it uses a Django interface that communicates with the frontend via a server.

The chat system will call specific functions of the system based on the required task. In essence a user interaction is mimicked to interface with the already available results from the Jump score computation (see Figure 6).

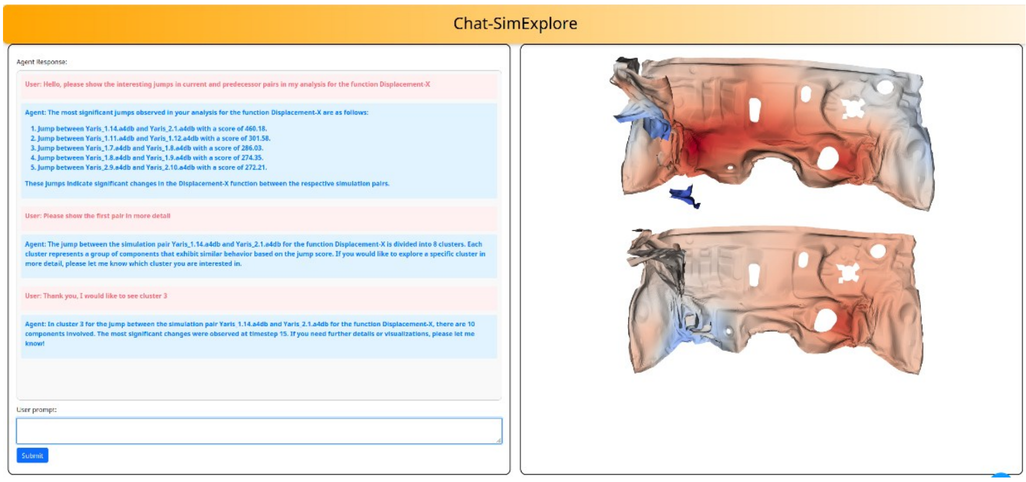

Figure 6:Example use of the web-based chat interface for the Jump score analysis.

5. RAG Approach to link model input (measures, sources) and effects

The emergence of Large Language Models (LLMs) has given further boost in developing AI based support assistant system for the design exploration and analysis in the virtual product development workflow in engineering. LLM's such as OpenAI's GPT-4o can be used in linking and automating different downstream analysis procedures, with the aim to provide relevant responses to the engineer's query. Moreover, the process becomes generalized, which means that independent on how the engineer frames the query, the support assistant can respond to the query by "reading in between the lines" so to say.

To effectively utilize the LLM in an engineering support assistant system, we consider two primary requirements:

1. There should be a way in which we are able to define and regulate LLM responses.

2. The proprietary data should be protected, and the data security regulations should be fulfilled.

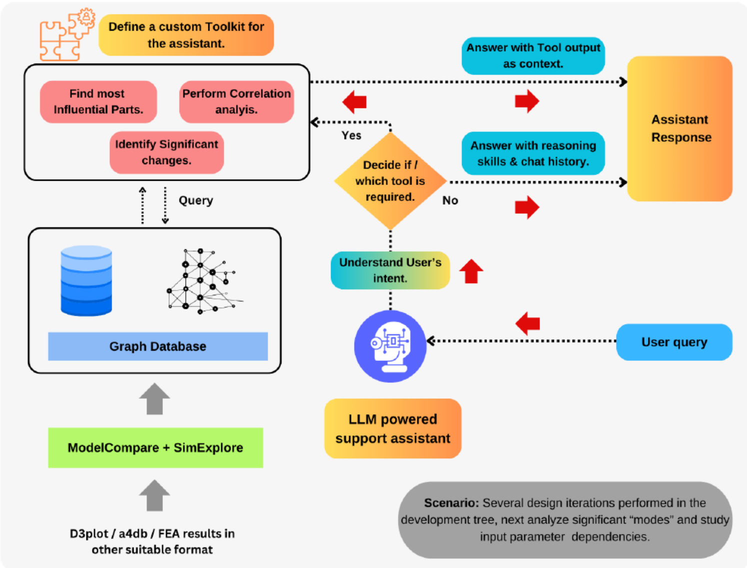

Figure 7:Overview of RAG approach utilizing LLM as agent to call specific tools to answer user's question (Fig 5.1 from [13]).

Considering above requirements, we propose an approach of designing a support assistant system in which the LLM is placed between the engineer (user) and our Fraunhofer CAE analysis tools [13]. One of the example designs adopting this approach is shown in Figure 7. With structured parametric prompt designs, we define the role LLM plays in different scenarios considering the product development stage at hand. Next, by utilizing function calling capabilities of LLM and with specialized python-based code structure, we have developed a custom toolkit with which analysis procedures using Fraunhofer CAE analysis tools can easily be automated after getting a LLM function call.

For the design scenario mentioned in Figure 7, the engineer is interested in analyzing multiple design iterations. Prior to the operation of the support assistant, in the considered design above, the simulations are analyzed with the LBSFA to extract meaningful low dimensional features. These features are stored in a structured format in a graph database with a vision to create a knowledge graph. During the operation of the support assistant, for a particular analysis procedure when called by the LLM based on the engineer's query, the analysis is performed at the backend automatically by the custom tool kit. The selected tool is utilized to retrieve essential proprietary data and relevant features stored in the graph database and conduct a detailed analysis. The analysis result is filtered and made more specific to answer the engineer's query and provided back to the LLM which receives further analyzes based on previous conversation and summarizes to the engineer in a suitable format. Moreover, the graph database can be updated with each conversation to facilitate fast retrieval of already performed analysis. Our specialized parametric prompt designs and python based “custom tool kit” has proven to show an effective approach to control LLM responses and define analysis procedures for different scenarios thereby reducing “hallucinations”. Consequently, the data used for analysis is not transmitted to the LLM, ensuring that all computations are performed locally by python functions. Moreover, the LLM has no direct control over data retrieval systems beyond invoking a python function within the custom toolkit responsible for data acquisition, thereby fully complying with the data security requirements outlined in the second point above.

This approach, where the LLM acts as a mediator—understanding and reasoning about the engineer’s query while determining the necessity of a specific tool—is inspired by the Retrieval-Augmented Generation (RAG) methodology. The parametric prompt design incorporates additional relevant information specific to the engineering analysis task, enabling the LLM to generate a more detailed and contextually accurate response.

6. Experimental Results for a Toyota Yaris frontal crash analysis

We demonstrate our developed RAG approach to identify jumps between behavioral modes in many simulation results on a frontal crash scenario. Our study is based on the Toyota Yaris study for a Euro NCAP front crash with a velocity of 56 km/h against a rigid wall which was set up and presented in [15, 16] and already investigated with our explorative ML-based approach in [10]. The study is restricted to the body in white parts of the vehicle, changes are applied to main structural beams adding beads as well as changing the position of them, whereas weight reduction is obtained by reducing the thickness of 4 main structural members, see [15]. Results of our explorative analysis, the web user interface and the RAG approach are shown in Figures 1-4 embedded in the algorithmic description above for better readability.

7. Conclusion

In this work we have presented an AI-driven methodology for the analysis and visualization of many CAE models which includes identification of design measures and the simulation outcomes. Concentrating on the standard evaluation of simulations in a development tree, the Jump score computes the most affected pair of simulations in a development tree and given it, groups the components as per similarity of time dependent outcomes.

A RAG system based on a LLM with implemented agents interacts with developed modules to enable the evaluation of the results of the Jump score. A modular software infrastructure with a backend and web frontend is used to demonstrate the approach. The RAG based system can access the information about design changes and simulation outcomes in a graph database and this in turn can be used to save input-outcome relationship providing the basis for a learning and inference framework for assisting engineers in the development process. Saving and learning this relationship is challenging but we think that the provided framework will enable us to further investigate this task in a more systematic and flexible way.

8. References

[1] F. Bianconi, P. Conti, N. Senin, D.R. Wallace, "CAE systems and distributed design environments", *In: XII ADM international conference*. Italy: Sept, 2001.

[2] H. S. Park, G.B. Lee, H. Kim, "Collaborative engineering system supporting product development process", *In: ASME design engineering technical conferences 2003*", Chicago, USA, Vol. 1, Part B. pp. 873–80, 2003.

[3] J. Sun, J.H. Frazer, T. Mingxi, "Shape optimisation using evolutionary techniques in product design", *Computer & Industrial Engineering*, 53, pp. 200-205, 2007.

[4] Q. Liu, W. Liu, F. Ruan, H. Qiu, "Parameters' automated optimization in sheet metal forming process", *Journal of Materials Processing Technology*, 187–188, pp. 159-163, 2007.

[5] R. Roy, S. Hinduja, R. Teti, "Recent advances in engineering design optimisation: challenges and future trends", *CIRP Annals*, 57, Issue 2, pp. 697-715, 2008, https://doi.org/10.1016/j.cirp.2008.09.007.

[6] M. Cordova, Dassault Systèmes the 3DEXPERIENCE Company, "Optimizing Product Development with model-based systems engineering", white paper, ad-optimizing-product-development-mbse-value-whitepaper-en.pdf, pp. 1-29, 2021, download from https://www.andea.com/media/24ej2lk0/ad-optimizing-product-development-mbse-value-whitepaper-en.pdf on 01/04/2025.

[7] R. Iza-Teran and J. Garcke, "A Geometrical Method for Low-Dimensional Representations of Simulations," *SIAM/ASA Journal on Uncertainty Quantification*, vol. 7, no. 2, pp. 472–496, Jan. 2019.

[8] D. Steffes-lai, M. Pathare, and J. Garcke, "Towards a Framework for Automatic Event Detection for Car Crash Simulations", in *NAFEMS World Congress 2021*, Online, 2021.

[9] J. Garcke, M. Pathare, and N. Prabakaran, "ModelCompare", *in Griebel M., Schüller A., and Schweitzer M.A., editors, Scientific Computing and*

Algorithms in Industrial Simulations: Projects and Products of Fraunhofer SCAI, pp. 199–205. Springer International Publishing, Cham, 2017.

[10] D. Steffes-lai, R. Iza-Teran, M. Pathare, T. N. Klein, J. Garcke, „Structured representation of the relationship between model changes and findings from the simulation results analysis using an explorative ML-based approach", in *NAFEMS DACH Konferenz 2024*, Bamberg, 2024.

[11] J. Garcke, R. Iza-Teran, M. Pathare, P. Schwanitz, D. Steffes-lai, „Identifying similarities and exceptions in deformations and mesh functions comparing many simulation results automatically", in *SIMVEC Kongress, Baden-Baden 2022*, VDI Berichte 2407, pp. 277-288, ISBN: 978-3-18-092407-6, 2022.

[12] R. Iza Teran, D. Steffes-lai, and L. Morand, "Geometrical Shape Learning as Basis for Compact Microstructure Representations and Microstructure-Properties Linkages", *European Journal of Materials*, 1–9. https://doi.org/10.1080/26889277.2025.2454654

[13] G. Devdikar, "Application of Large Language Models in Automotive Product Development*", Master's Thesis, Technische Universität Dresden, Fraunhofer SCAI*. Sankt Augustin, Germany, 2024.

[14] A. Pakiman and J. Garcke, "Graph modeling in computer assisted automotive development", *In 2022 IEEE International Conference on Knowledge Graph (ICKG)*, pp. 203–210. 2022.

[15] H. Singh, A. Schumacher, et.al., „Hierarchical Multi-Level-Optimization of crashworthy structures using automatic generated submodels", in *11th European LS-DYNA Conference 2017*, Salzburg, Austria, 2017.

[16] D. C.J. Falconi, A.F. Walser, H. Singh, A. Schumacher, "Automatic Generation, Validation and Correlation of the Submodels for the Use in the Optimization of Crashworthy Structures", in *Schumacher A., Vietor T., Fiebig S., Bletzinger KU., Maute K. (eds): Advances in Structural and Multidisciplinary Optimization. WCSMO 2017*. Springer, Cham., https://doi.org/10.1007/978-3-319-67988-4_117, 2018.