A Next Generation Software Platform for LS-DYNA Modeling and Configurable Vertical Application Development

H. Ouyang, Tim Palmer, Q. He

Engineering Technology Associates, Inc.,

Troy, Michigan, USA

Summary:

Finite element modeling tools have undergone a transformation in recent years. These tools have been made easier to use, configurable for vertical applications and able to interact with outside applications. Such software needs to not only efficiently create models, but are also expected to interact with other software tools, be configurable and anticipate the future needs of users by being able to extend the users ability to assimilate models and other data efficiently into an LS-DYNA environment.

Using the heritage of a stable, well-featured finite element modeling software, ETA has developed a new software platform to create finite element modeling software that meets the challenges of today's users, both from an infrastructure and a data management standpoint. This will offer users an opportunity to create, share and manage models using standardized interfaces, scripting tools and both standardized and user-defined processes.

This paper will present the opportunities that this new platform will offer users and the future of modeling tools that empower users of optimization and synchronous design tools.

Keywords:

FE Modeling, Database architecture, Vertical Applications, VPG, DYNAFORM, Drop Testing

1 Introduction

Finite element modelling software has long been the 'face' of the finite element solver. The user interacts with the solver through a graphical interface which aids them in creating the necessary data structures to simulate their engineering problem.

As solvers matured, so did the modelling tools. They have gone from the monochrome monitors of the 1980's, with rudimentary 3D graphics to fully interactive systems which engage the users and help them quickly and efficiently generate their models and query the results. This increase in productivity has allowed engineers to spend more time engineering and less time as data entry operators. This capability has also allowed engineers to consider ever more complex simulations, some of which are very unique. This uniqueness then highlights the inability of the generic finite element modelling tools to efficiently build these one-of-a-kind models.

Creating special purpose modelling tools was once the norm. Most large companies had specialized finite element modelling applications which met their specific needs. The costs and overhead associated with the development of these tools and the fact that creating tools is not their core business, has all but eliminated the specialized modelling tool development at large multi-nationals. There is a need for general purpose finite element modelling tools with the flexibility to be configured for unique simulation processes for today's simulation engineer to remain effective and efficient.

2 Introducing a New Software Architecture

ETA has developed a new software architecture which will allow developers to create pre-packaged modules which provide vertical applications; specialized user interfaces for a specific type of simulation, and also user configured vertical applications. This new architecture also allows the user to have the modelling tools interact with other software tools, to pass data to these applications, or allow those applications to influence the finite element model.

3 Vertical Application Development: The Productivity Boost

A vertical application is a toolset or set of processes which is focused on a narrow set of simulations. These usually are aligned with a specific industry or product type, for instance vehicle crashworthiness, metalforming or electronics shock and vibration. Development of vertical application toolsets within a software's architecture allows users to use pre-configured process sets, or develop their own vertical application toolsets. These can be built around a similar physical testing process, allowing for a means for direct correlation of simulation results. With the many tools built in to solvers such as LS-DYNA which are used by only a subset of the overall users, there comes challenges on how to best use them.

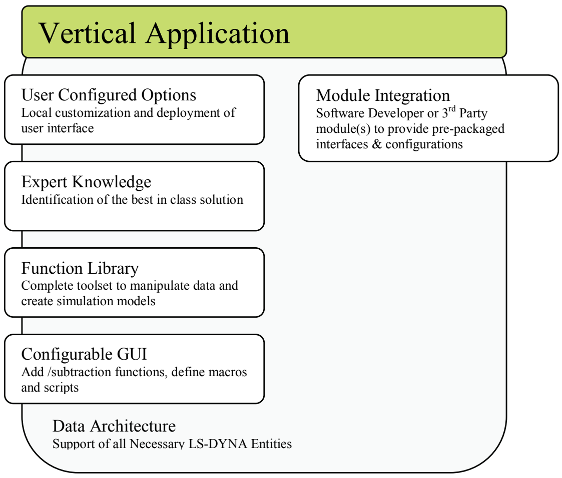

Figure 1: Vertical Application Scheme

It is surprising sometimes to speak with users and learn that they are not aware of some of the latest methods available to them within a solver. How can the user even find these new features and assemble them into a process? It may be necessary for an outside developer or expert to identify the newest features in a software and deploy these through pre-packaged modules.

To provide the greatest productivity boost, the user must feel as if the software was designed for them, personally, to do their specific job. That's where the capture of expert's knowledge and the subsequent development of vertical applications add immense value to the user.

The requirements for a software platform to implement these types of features are:

- Complete function set
- Configurable User Interface
- Ability to capture processes
- Interaction with external databases and software tools
- Open access to functions, scripting access

3.1 A Complete Function Set

As finite element modelling software have matured, they have provided users with many different functions to accomplish a goal: build a finite element model for use in a simulation. Sometimes these functions have become more automated as computer processor power and graphics engines have progressed.

As a product family the latest ETA software architecture delivers an enormous function set. Historically, you can find these functions in previous generations of the software product, dating back to the mid 1980's. These types of functions will always be linked with finite element modelling; node creation and element creation. As the generations of software progress, we now automatically generate these entities in a way that the casual user never knows that these functions were once a series of individual operations, each of which had their own 'art' in executing.

A User Manual that once was 100 pages, now is over 500 pages and growing with each release. This explosion of features means that users have multiple ways to do the same process, and that specialized features have been developed to provide tools for virtually every situation the user may encounter. This also means that behind the scenes, process automation is happening without the user comprehending it.

When a vertical application is being studied for automation, it is necessary for all of the possible functions and tools to be available. Having an extensive set of tools for this work is vital for the success of the application. Guiding a user 90% of the way through a process and requiring them to manually complete the last 10% will result in less adoption of the toolset and limit the effectiveness due to widely varying user experience levels and process variability.

3.2 Configurable User Interface

As the options and functions available to users grow at a high rate, the user interface's ability to display the range of functions and guide the user through the process becomes problematic. User interface design must be flexible enough to be able to display all of the options available to a user at a specific point in the modelling process. It must also be configurable to allow the user to streamline the toolset and hide unwanted or unnecessary items from view.

Allowing the user to have a set of tools that are focused on their process is critical in user efficiency and satisfaction. Imagine if a mechanic had to carry all of his toolchest with him even if he just needed to adjust a setting which needed a screwdriver and some lubricant. While he would probably be able to perform his job, he would be burdened by the extra work and confusion of too many tools and the chance of selecting the wrong tool. He would certainly like access to all his tools if necessary, but after he has performed a process successfully, he knows he only needs a small selection of tools to complete his job.

The ETA architecture allows the user to add/remove and reorganize the functions to streamline the interface. However, just like that mechanic, the toolchest is still easily accessed to get that tool needed for a unique or unforeseen situation.

A simple example of a configured user interface is that of a finite element modeller who only constructs solid element meshes. This engineer could streamline the menus by removing all modelling & model checking tools, material types and section cards that are exclusively used in shell meshing. This would potentially remove over 75% of the total menu options improving the interface ergonomics and thereby improving the efficiency of the user.

These configurations could be stored as user profiles (for instance named 'solid meshing'), which the user could load for a specific task. A typical user may create and save several such profiles allowing them to configure the product for the tasks at hand. This customisation of the toolset is available at three levels: the default, full set of functions, a user profile set of functions, and a project level configuration.

3.3 Ability to Capture Processes

The highest value and most specialized engineering applications today are not CAE Tools; they are knowledge tools. They contain validated engineering knowledge, ability to capture and retain new knowledge, classify and contextually understand the knowledge.

The term "process capture" has been used recently to describe the ability of a software to replay a series of commands, or to make the series of commands a more general process by which a repetitive task can be executed with minimal user intervention. Process capture also allows the user to save his knowledge and pass it on to a colleague or build a knowledge base which becomes a part of the company's intellectual property.

Process capture in the ETA software architecture is realized through a service which enables the user to capture not only the process but also the product knowledge through a parameterised geometry engine, which can place limits on the size and shape of an object, and parametrically define its relationship to other entities in the model. This, along with the standard process capture, allows process knowledge to be captured and combined with product knowledge.

3.4 Interaction with Other Software Tools

Engineers may use many different tools to develop a product. Input may come not only from design performance, but cost models, material utilization, production constraints, or some other product design consideration that may be either contained in a database or based on another calculation.

The ETA software architecture not only allows for interaction with external software and database applications, but allows for the reuse of other finite element models. The ability to import and reuse the product knowledge contained in ADAMS, NASTRAN, RADIOSS, PAM-CRASH or ABAQUS models may be imported and converted automatically into LS-DYNA models. This underlying capability of the ETA software architecture is easily updated through a template system.

3.5 Open Architecture / Scripting Access

The ETA software architecture allows the user to capture the process, replay it and modify it. This process capture is based on a basic capability of the software to execute any function through a scripting interface.

To be truly open and accessible for outside software tools and configurable for processes, each function within the software must be accessible through a scripting access. Passing variables or data between software products provides the foundation for parametric design optimisation, where the FE modelling software reacts to the results or direction dictated from external software, and automatically updates the model. In this case the design may be 'driven' by outside software.

This type of capability, combined with powerful, robust and accurate modelling functions promises to change the way that engineers develop products, and will unlock the potential of optimisation software for the use in everyday problems.

4 Examples of Vertical Application Toolsets

When vertical applications are identified, they require a set of specific knowledge be placed in a toolset. The resulting toolset must have the ability to be robust and applicable to not only a specific problem, but a class of problems or a genre of engineering problems.

Capturing the knowledge of experts and allowing others to effectively utilize that knowledge is one of the main aims of automating processes. Working with experts in specific simulation domains allows software developers to create effective vertical application modules.

4.1 Standard Testing Procedures: Drop Testing

One such example of a vertical application toolset is the drop testing menus. Drop testing is commonly performed on electronic products and their packaging. This type of testing is used to assess the product's resistance to shock and vibration, and assures the product reaches the customer in working order.

The implementation of the Drop test vertical application menu within the product allows users to have a smaller, more specialized toolset for their specific type of simulation. This series of menus collects the common modelling tools needed to set up and execute a series of drop test simulations.

The drop test vertical application is built around the procedures found in the corresponding laboratory tests. In these tests, specimens are dropped on rigid surfaces, in various orientations, and from various heights. They may also be mounted on shaker tables or impacted with indenters of varying sizes.

Users who make use of simulation for drop test are quite specialized, and do not typically perform other types of simulations such as crashworthiness or metalforming. Their modeling needs may be quite extensive, but the boundary conditions that are used are quite limited. They need to calculate impact velocities due to drop height, gravity vectors, rigid walls and need only a few outputs. Having a well defined set of parameters make this an ideal case for process automation.

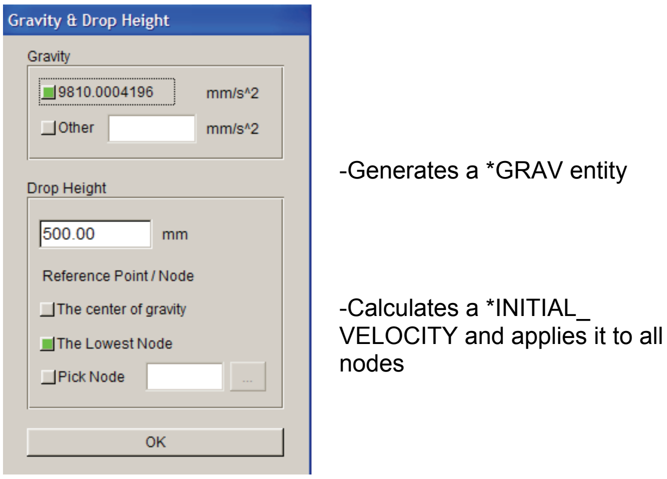

Figure 2: Drop Test Interface Example

To accomplish this, terminology used by the specific user community should be adopted. Many users of application tools are not experts in the terminology of LS-DYNA. Many users are also the test engineers, who are concerned with physical tests and the terminology associated with testing. Having simple terms such as 'drop height' and 'drop angle' may seem simple, but if other, more LS-DYNA specific terms were used, they would be confusing.

The ETA software architecture allows the developers to create a specialized set of menus that use the terminology consistent with the test environment. The combination of functions in the software provides the user with a series of commands that are executed in the background, and only the resulting operation is displayed for the user.

For instance, when the user specifies the drop height, the software will calculate the drop time and resulting free fall velocity, based on the height and gravity constant. Knowing this information, the software applies an initial velocity on each of the nodes in the model. This is a series of separate operations that, to the user, looks like one function.

4.2 Common Processes and Unique Needs: Vehicle Safety Simulation

Vehicle safety has many different types of tests and various evaluation criteria. Each of these tests has a prescribed testing device, test setup procedure and specific quantities to measure. The results of the tests are typically compared to a set of established criteria, and the response pass/fail is easily identifiable.

Since there are specific test devices, specific responses, and clear evaluation criteria, it qualifies nicely as a process to standardize.

Specific tasks which are common to most engineers working in the area of vehicle crashworthiness and occupant safety can be identified. For example dummy positioning and barrier positioning are two common tasks performed by most all vehicle safety engineers. Creating a developer-supplied set of templates and process tools is quite natural for these tasks. However, as vehicle products are possibly quite unique with respect to corporate standards and testing norms, the user may have additional needs to develop their own, more customized processes. An example of this type of process may be a rollover test simulation where there are no legislated test procedures, but companies may have their own proprietary testing process.

Together, these can form a vertical application which meets the needs of an overall industry and the specific needs of a particular company.

4.3 Industry Specific Terminology: DYNAFORM

To make the toolset usable by an industry, it must make use of common nomenclature found in that industry. An excellent example of how this has been implemented is the eta/DYNAFORM product. It has made use of the common terms used by those in the tooling industry, the vast majority of which do not correspond to any terms used in the LS-DYNA finite element software.

Users need to understand the terms and allow the software to translate that into the language of the solver. Knowing that a punch speed definition is translated in to a *INITIAL_VELOCITY keyword, is not of much interest to most tooling engineers, but that the software reliably translates their needs into the proper LS-DYNA input is a basic requirement.

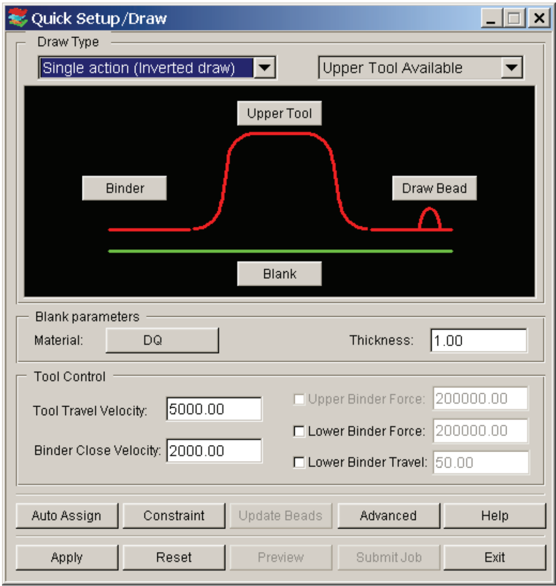

Figure 3: DYNAFORM Quick Setup Interface

5 Future Applications of Configurable Menus and Interactive Applications

The combination of various vertical applications is promising to transform the product design and development process. By combining a series of toolsets, the user could leverage automated modeling, simulation and optimization tools to identify optimal designs.

Starting with tools to parameterize a finite element model and its underlying geometry, the results of simulations, database queries, combined with design goals, has already proven to automate the process of refining design concepts. This process, called a "CAE-Centric Process" has beem successfully implemented in vehicle design studies. Early CAE-Centric processes used semi-automated processes and specialized functions to perform the tasks. Starting only with an initieal design concept, the ETA software architecture has allowed the continuous loop of simulate/optimize/modify to be carried out in a hands-off environment.

This provides a platform for multi-disciplinary, multi-objective optimization on a large scale. Utilizing the ability of the software to communicate with outside applications to drive the design changes, automatically re-mesh and export and execute input files and post-process and evaluate the results, the next horizon of CAE is quickly approaching.

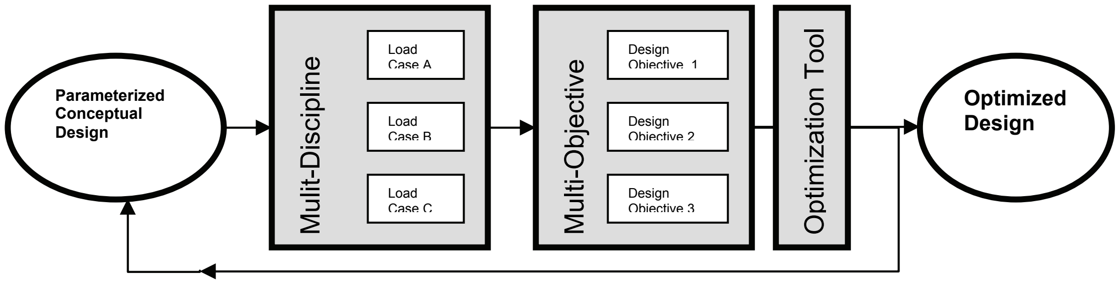

Figure 4: CAE-Centric Process Utilizing Scripting, Process Capture and Interaction with Multiple Software Products

6 Summary

New software architecture has been developed which allows user to interact with a familiar style of user interface, configure it to their specific needs, and capture and deploy their corporate knowledge. The availability of such tools will allow the expanded use of software tools such as LS-DYNA, and allow users to take advantage of the many possible applications of the software.

The next generation of CAE tools will leverage these basic capabilities to evolve into a CAE-Centric process which drive product innovation through automated design processes.

7 References

Additional information may be found in these publications:

[1] Flemings, T.J.: "NAFEMS Regional Summit", CAE 2020 Roundtable Presentation, 2008