CS6750 Project – Summer 2021

Patrick Kriengsiri
pk@gatech.edu

Abstract—Femap is one of the leading software packages for structural analysis in the aerospace and defense industry. In continuous development for nearly forty years, it contains a mature and diverse feature set that is required by engineers; however some features and interaction paradigms were developed long before emphasis on HCI was an active part of software development. This paper explores strengths and shortcomings of one of the most common interfaces – entity selection, as well as proposes a user-centric redesign that users expect out of any modern software package.

1 INTRODUCTION

1.1 Introduction

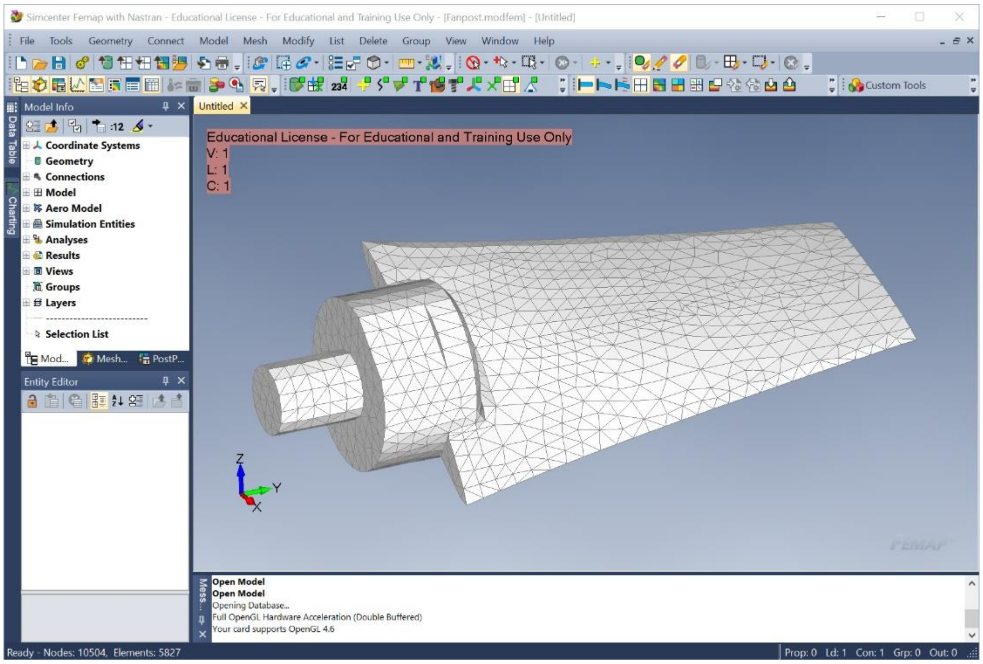
Femap is a finite element pre-processor and post-processor used for structural analysis in a variety of industries, including aerospace, defense and marine. The role of pre-processing and post-processing software is to build a finite element model that is then exported for numerical analysis via an external solver (e.g. Nastran, Abaqus, Ansys), the results of which are then imported for further analysis.

In continuous development since 1984, Femap has a broad customer base that includes long-time engineers as well as an emerging base of novice users. Having been in development for so long has both benefits and drawbacks. While the feature set is broad and mature, many of the interaction paradigms were developed in a time before HCI and user-centric-design we considered as primary motivators for software development. As a result, some aspects of the program have been remained unchanged for fear of alienating an existing user base at the expense of usability and learnability, especially for new users.

I was a software developer for that program for over a decade and was a user for many years prior. I recently changed jobs so am no longer involved with the product is an attempt to apply concepts of HCI and user-centric-design on a specific aspect of the program (entity selection) in a holistic and (hopefully) unbiased way now that I'm no longer affiliated with the program.

1.2 Interface Description

In finite element analysis, a finite element model is comprised of finite element entities (such as nodes and elements) that are created by an external program (pre-processor) such as Femap. These finite element entities can either be created individually or created as a mesh from geometric entities (points, curves, surfaces and solids). The primary user interface for Femap is shown in Figure 1.



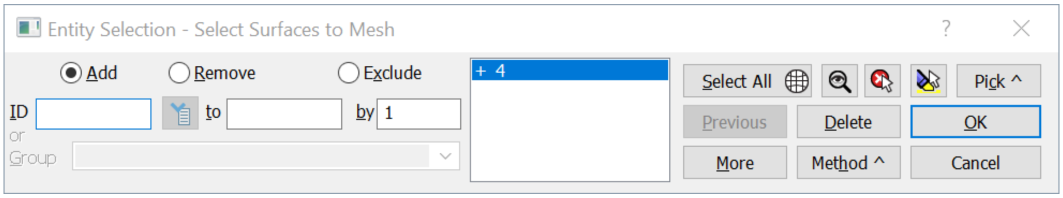
Open Model
Open Model
Opening Database...
Full OpenGL Hardware Acceleration (Double Buffered)
Your card supports OpenGL 4.6

Ready - Nodes: 10504, Elements: 5827

Prop: 0 Ld: 1 Con: 1 Grp: 0 Out: 0

Figure 1 — Femap User Interface

Within Femap, an entity selection dialog used for selection of a single type of entity at one time. Femap has a *verb-noun* interaction paradigm meaning that a user selects an operation from a menu or a toolbar (verb) to raise the entity selection dialog (noun). An example of the dialog is shown in Figure 2, where the command *Mesh->Geometry->Surface*



● Add ○ Remove ○ Exclude

by 1

+ 4

Figure 2 — Femap Entity Selection Dialog

This dialog appears as a modeless dialog at the bottom of the primary user interface and allows users to select entities in various ways, including on-screen selection, manual entry, and a variety of advanced methods (such as within a bounding box) found under the *Pick* and *Method* submenus. Selected entities are visually represented on the screen with a carat (Figure 3) and also listed in the selection dialog (Figure 2).

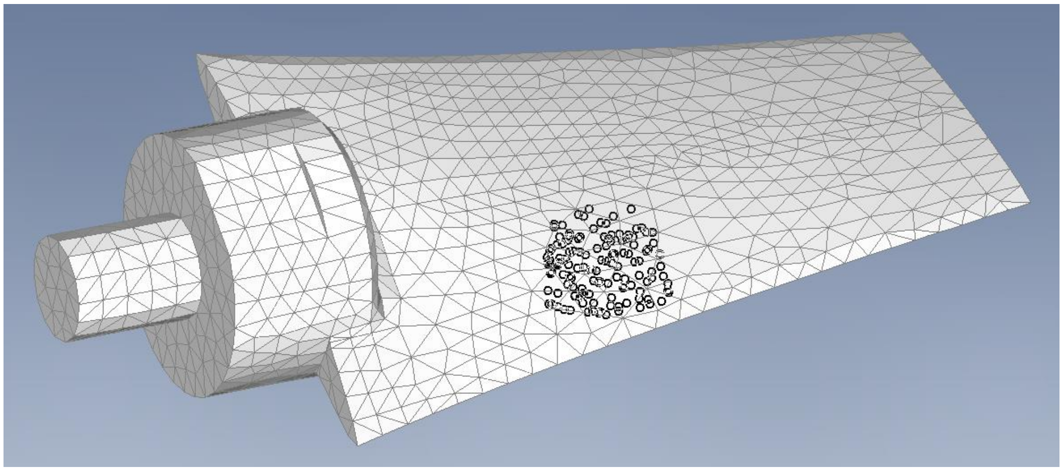


***Figure* 3** — Example of Selected Entity Carats

1.3 Try it out

If the reader of this paper would like to try Femap, a student version is available for download free-of-charge at the following URL:

https://www.plm.automation.siemens.com/plmapp/education/femap/en_us/free-software/student

2 INITIAL NEEDFINDING

2.1 Needfinding Plan

A variety of needfinding activities were considered for this project, however most were deemed unsuitable for the scope of this project. For example, there's generally not an open repository of user reviews for engineering software, so review surveys would not yield usable results. From the M exercises, my experience with user surveys was that the free response questions generally didn't yield much qualitative data, which I felt was important for this redesign.

Therefore, user interviews are the chosen needfinding method for this project. All survey participants are acquaintances of mine and were aware that I used to be a developer for Femap, however all were also made aware that I am no longer informed that I am no longer affiliated with the product; users were encouraged to be honest in their assessments and this likely mitigated any social desirability bias.

Interview participants were first read an opening script describing the purpose of the activity, then were asked as series of open-ended questions, shown below. Prior to conducting the interviews, the interview script was reviewed by a third party to try to avoid any bias or leading questions.

Opening script: Thanks for participating in this interview. The purpose of this is to gather some feedback regarding entity selection in Femap for an academic exercise in redesigning the interface for a more user-centric experience. As you may or may not know, I recently left Femap, so I'm no longer affiliated with the product – I encourage you to be as honest as possible; you're not going to hurt my feelings.

Also, know that because I'm no longer with the product, the results of this project may not get implemented. However, with your permission, I'd like to share the results of this with the development team so that they get some useful feedback and you also have the benefit of having your voice heard (we'll cover that at the end).

- How long have you been an engineer?
- What other finite element pre/post-processors do you have experience with?
- How long have you been a Femap user?
- How often do you use Femap?
- Could you describe your general impressions of entity selection in Femap?
- Do you remember how long it took you to get accustomed to entity selection in Femap / did you find there to be a difficult learning curve?
- Do you have any suggestions on how entity selection could be improved in order to better facilitate your workflow?
- Do you have any additional suggestions?

- As I originally mentioned, I'm no longer a developer on the Femap team, so there's no guarantee the results of this project are going to be implemented in the program (sorry to get your hopes up if you liked the idea). Having said that, I'd like to share this project with them. Could I include your interview feedback in anything I provide? It would be a consolidated version of all the transcripts and no identifiable information will be included.
- Thank you for your time.

Ideally, this activity would have been best performed as a focus group rather than individual interviews to facilitate building off of ideas, however due to the current health crisis, that was not an option. Interviews were conducted one-on-one via in-person, over-the-phone, and Microsoft Teams. The interview was conducted with seven participants. A larger participant base would have been ideal but despite the small size, useful and relevant result were still obtained.

2.2 Needfinding Results

The following is a summary of the needs uncovered during participant interviews. See Appendix A for aggregate interview results.

The main takeaway regarding the current selection interface was the flexibility and number of options available. Experienced users appeared to appreciate its consistency across multiple versions, however newer users also reported being intimidated by the interface initially and found it to have a more difficult learning curve compared to other software packages they had used. Once users had become proficient, they felt it was an efficient interface.

When participants who reported a difficult learning curve were asked specifically what aspects of the current selection interface were difficult, it was mentioned that Femap's selection interface does not function like most other finite element programs. One user who spends time in several different finite element programs reported that Femap was the hardest to regain proficiency because it is different.

Additionally, multiple users also specified they expected to just be able to select entities on screen and interact with them. One user reported that the multitude of available options was confusing, and they don't ever use the majority of them.

They also felt this was the reason it felt intimidating – in other programs, you "just pick."

Some miscellaneous critiques of the selection interface and Femap in general included:

- The user interface requires too many clicks to get something done
- The carats that are used to mark selected entities is hard to see once you select more than a few entities and some functions can make them disappear making it very difficult to know what was selected before going into the command
- With so many available commands, it can be difficult to remember where each one is
- Don't like how you can't save selection lists, don't want to always create groups and have to select from those

The interviews also indicated that user experience level and engineering experience level ranged the gamut from brand new to decades of experience, indicating that a wide range of users should be catered to. Some users use Femap exclusively, while others jump around between several different programs. Most users were under constant deadline pressure, and all indicated they needed an efficient interface, be it a redesigned one or the existing one. The task was never indicated to be entity selection itself but performing some operation on the model.

3 HEURISTIC EVALUATION

3.1 Strengths of the Current Interface

Consistency – One of the main strengths of the entity selection dialog is *consistency*. The same dialog is used for all entity types; once a user is familiar with the basic concept of entity election, all the basic and advanced selection principles apply to all entities for nearly all commands in the program. This *consistency* principle applies not only across different commands, but across time as well. To avoid confusion or alienation of the existing user base, the dialog remains relatively unchanged for nearly the entire life of the program.

Constraints – Because entity selection is performed after the command (i.e. task) is selected, this allows the selection dialog to be *constrained* to only select an appropriate entity type. By limiting the entity type to only the type relevant to the command, this reduces the chances of user error.

Flexibility – Similar with consistency and constraints, the entity selection dialog provides a lot of *flexibility* – while there may be a more advanced learning curve to understanding the more nuanced and advanced selection features, users are, for the most part, able to use it how they want to interact with the program, either through basic picking or more advanced methods such as Boolean rules. When new functionality has been added in the past, it augmented existing functionality. For example, one interview participant brought up how obsolete functionality they first learned how to use nearly twenty years ago is still supported to this day, negating the requirement for them to learn the new method.

Documentation – Having a consistent user interface for an extended period can also lead to mature documentation; there is an abundance of help documentation in printed, on-screen and, more recently, video form for all aspects of the entity selection dialog. This may also be a potential area for improvement, however, as the volume of documentation available can also hinder the ability for the user to find documentation specific to their exact need.

Gulfs of execution / evaluation – There are a number of ways that the current entity selection dialog bridges the gulfs of execution and evaluation. There are a number of keyboard shortcuts that reduce the number of mouse clicks required, bridging the gulf of execution. The gulf of evaluation is partially bridged by using on-screen markers to indicate to the user entities that have already been selected (this is also an area for improvement).

3.2 Areas for Improvement Within the Current Interface

Discoverability – One of the most common criticisms of Femap heard through needfinding was the inability of new users and users that have to switch between multiple software packages to inherently know where the desired command lies in the menu structure as well as buried functionality within the selection menus. While the functionality does exist, the current interface doesn't lend itself to being easily *discoverable*.

Direct manipulation – The verb/noun usage paradigm within Femap (selecting the command before selecting objects on which to operate) versus a noun/usage paradigm is a very subtle, but profound difference. It effectively makes the user feel like they're not directly manipulating an object / selection of entities because the object of the task has not been defined prior to the task. This only serves to widen the gulf of execution. It is also inconsistent in usage with many other CAD/CAE programs, violating *consistency* within a suite of similar programs.

Feedback – A criticism raised by a user in interviews was that when a selection is made on-screen its selection is marked by a carat (see Figure 3). For small number of selected entities, this may be an acceptable level of feedback, however when large numbers are selected, the display can become difficult to interpret the results of their selection.

Tolerance – The selection dialog can be somewhat intolerant of errors. In order to select an entity on screen, dialog focus must be active within one of the circled controls in the dialog:

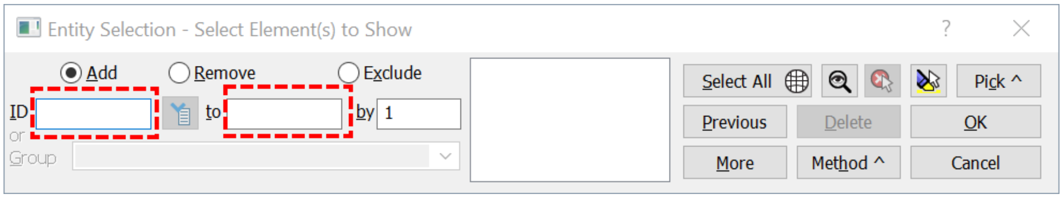
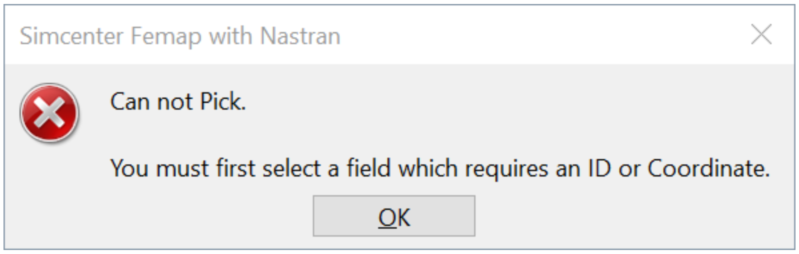


Figure 4 — Dialog fields that must have focus in order to perform on-screen selection

If the user attempts to make a selection on-screen without a proper control having focus, they are presented with the following somewhat cryptic and generic error message:



Can not Pick.

You must first select a field which requires an ID or Coordinate.

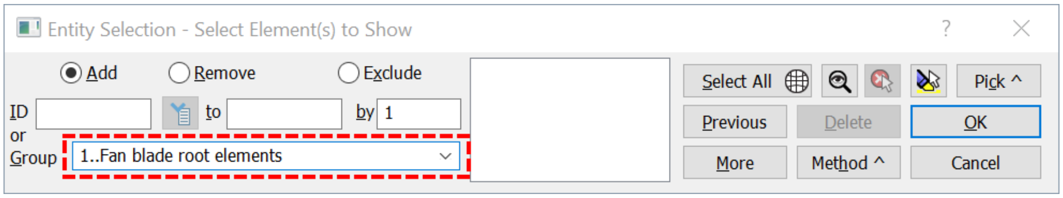
Figure 5 — On-screen selection error message

From my personal experience in providing instructional training for new users, this particular error is particularly frustrating to novice users and is one of the most common errors seen by all users (this is supported by telemetry data). Additionally, the error dialog is raised as a modal dialog, meaning the program can't be interacted with until the dialog is dismissed. This is disproportionate to the level of slip that occurred and discourages users from experimenting within the program.

Learning curve – The difficult learning curve for Femap was something brought up during interviews. The amount of functionality available and the lack of discoverability are major contributors to this.

Gulfs of execution / evaluation – Similar to how the entity selection interface has a number of aspects that bridge the gulfs of execution and evaluation, there are also aspects that widen the gulf. The example given for how the interface bridges the gulf of execution is also a way in which the gulf exists – keyboard shortcuts / accelerators can reduce the number of steps to complete the task, but their existence is not made obvious in the UI itself – users that are unfamiliar with them actually have a wider gulf of execution because they don't inherently know how to accomplish the task.

An example of a wide gulf of evaluation is the ability to select an existing group of elements using the Group combo box as shown in the following figure:



Entity Selection - Select Element(s) to Show

● Add ○ Remove ○ Exclude

ID ____ to ____ by 1

or

Group 1..Fan blade root elements

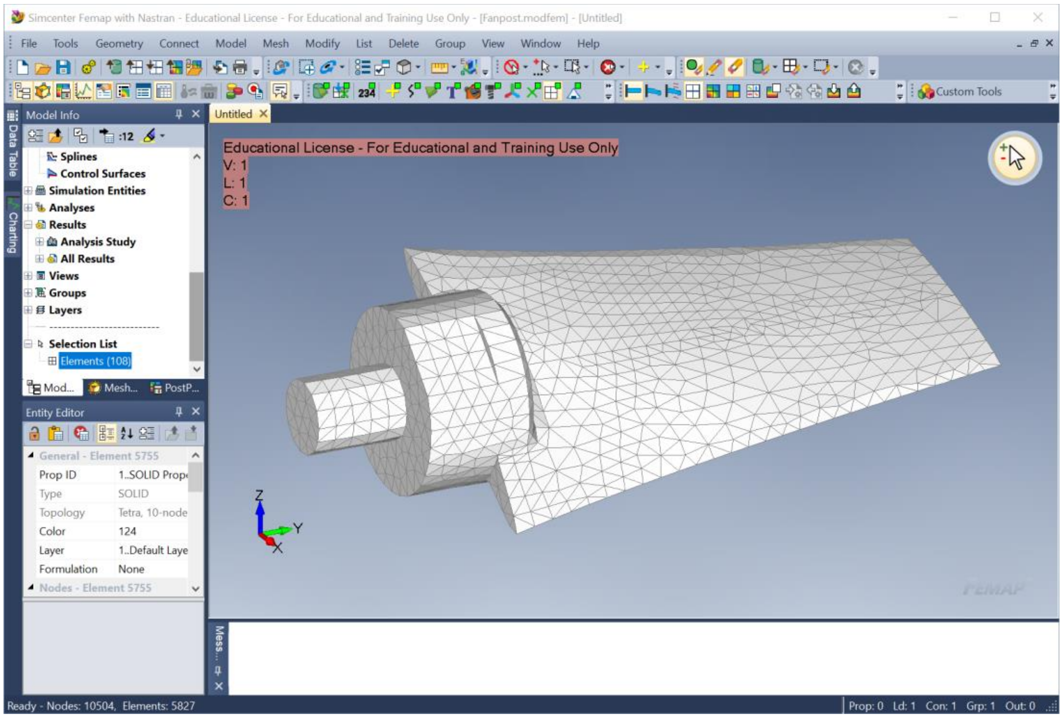
Figure 6 — Group selection option

The wide gulf of evaluation comes from the fact that this is insufficient to actually select the elements – to do so, the user must click *More* prior to clicking *OK*. There is no indication to the user that what they think was selected didn't actually get selected.

Expert blind spot – this particular disadvantage wasn't uncovered through need-finding or heuristic evaluation but is a known deficiency from my own experience. The development staff for Femap is comprised mostly of former engineers that have used the product in a production environment and can bring real-world experience to the table while implementing new features or redesigning existing ones. This is a double-edged sword in that at one point most developers were also a user so they have a unique understanding of industry needs, however, I can say from personal experience that that advantage diminishes over time. More importantly, however, this violates a cardinal rule of proper UI design – "I am not the user." Features and functionality could be implemented based on what I thought the market need was, however, this wasn't always accurate and was rarely supported with feedback from current users.

4 INTERFACE REDESIGN

A wireframe prototype was created in PowerPoint to illustrate the interface redesign. The redesign centers around a new "Selection Widget" that persists in the top right corner of the viewport as seen in the following figure:


Simcenter Femap with Nastran - Educational License - For Educational and Training Use Only - [Fanpost.modfem] - [Untitled]
File Tools Geometry Connect Model Mesh Modify List Delete Group View Window Help
Custom Tools
Data Table
Charting
Model Info
:12
Splines
Control Surfaces
Simulation Entities
Analyses
Results
Analysis Study
All Results
Views
Groups
Layers
Selection List
Elements (108)
Mod...
Mesh...
PostP...
Entity Editor
Untitled
Educational License - For Educational and Training Use Only
V: 1
L: 1
C: 1
Z
Y
X
FEMAP
Mess...

General - Element 5755

Prop ID	1..SOLID Prop
Type	SOLID
Topology	Tetra, 10-node
Color	124
Layer	1..Default Laye
Formulation	None

Nodes - Element 5755

Ready - Nodes: 10504, Elements: 5827

Prop: 0 Ld: 1 Con: 1 Grp: 1 Out: 0

Figure 7—Selector widget shown in the top left corner of the viewport

The default out-of-box behavior for the selection widget is to be "active" for toggle selection (clicking an unselected entity will add it to the selection list, while clicking a selected entity will remove it from the selection list). Like the existing entity selection dialog, the user can just pick on entities in the viewport by clicking them, however unlike the standard selection dialog, the default behavior is to select all entity types (this can be configured, discussed later). If the user clicks on the selection widget, it disables active selection.

As entities are selected / deselected, a live count of entities is shown near the widget, and the selection list (an existing feature of the *Model Info Tree*, seen on the right-hand side of Figure 7) is also updated.

A close-up view of the selection widget in active and inactive states are shown in the following figure, as well as the widget with a count of selected entities.

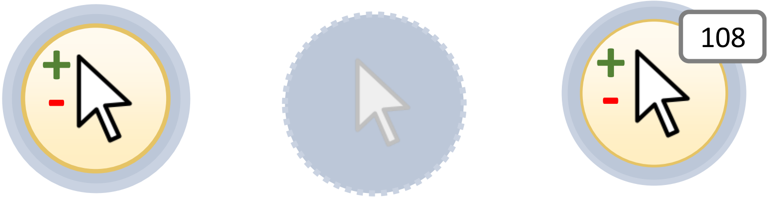


Figure 8—Close-up view of active selection widget (left) and inactive selection widget (center) and selection widget with 108 entities selected (right)

In addition to updating the selection count in the widget and tree, the viewport is updated by highlighting selected entities, as opposed to using a carat to indicate selection (ref: Figure 3). As an individual entity is hovered over, it will be shown in blue to indicate which entity will be selected / deselected.

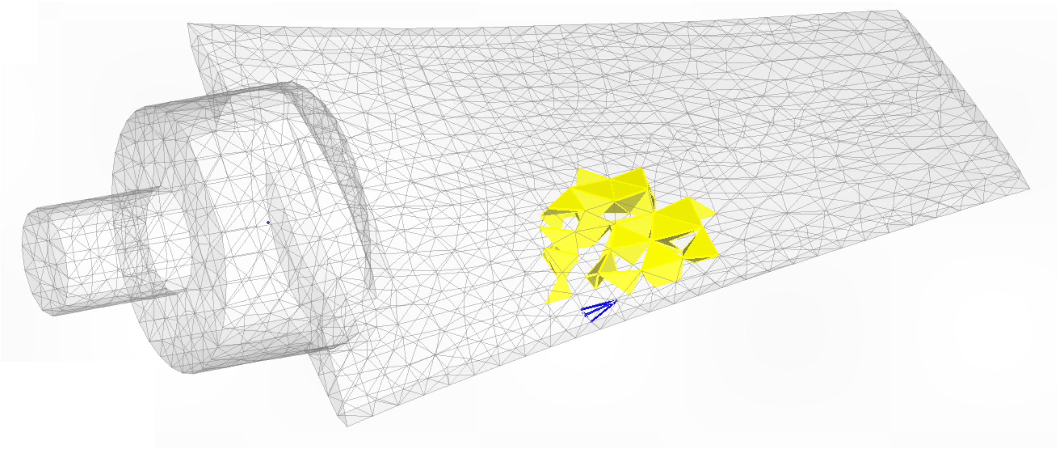


Figure 9—Selected entities shown with highlighting instead of carats, hovered entity shown in blue

As the user hovers over the selector widget, a flyout menu appears providing buttons for additional options.

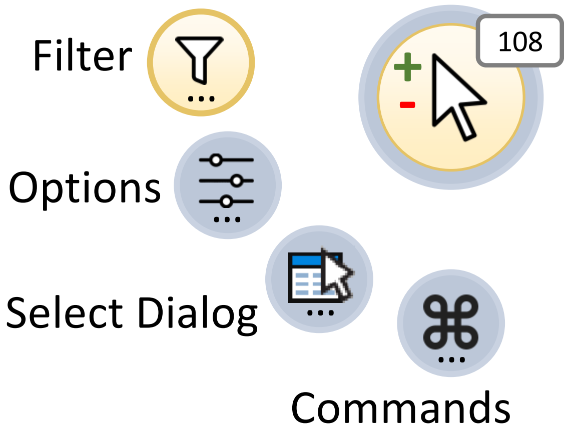
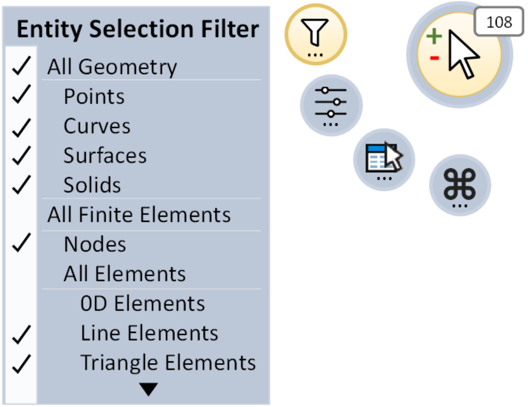


Figure 10—Selection widget flyout menu

The text labels in the above figure are also only shown for illustrative purposes in this paper. Each button has an ellipsis at the bottom to indicate additional functionality is available. When each button is hovered over, a menu will display showing additional commands and the button will highlight, as seen with the filter button.

Filter – the filter menu is intended for the user to select specifically which type of entity they would like to select. As previously mentioned, the default behavior is to select all entity types, however the user can pare down this list by manually specifying which entity types they would like to be selectable. An example of the *Filter* menu is shown below


Entity Selection Filter

✓ All Geometry
✓ Points
✓ Curves
✓ Surfaces
✓ Solids
All Finite Elements
✓ Nodes
All Elements
0D Elements
✓ Line Elements
✓ Triangle Elements

Figure 11—Filter menu example

Options – The options menu provides additional selection related options, such as the selection mode (add/remove, add only / remove only), tools such as box picking, circle picking freehand picking and the ability to use advanced Boolean selection options via an additional dialog (such as selecting elements by topology or color).

Additionally, there is an option to save the current selection list, load on from the database or clear the current selection.

Finally, there is an option to hide and disable the selection widget entirely for users that don't want to use it.

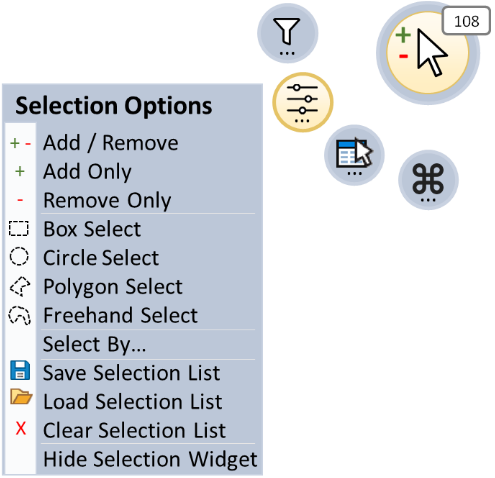


Figure 12 — Selection widget options menu

Dialog – the selection widget is not meant to replace all the advanced functionality from the existing selection dialog – its purpose is to provide quick and intuitive access to common selection tools. Therefore, the existing selection dialog can be used to select specific entity types.

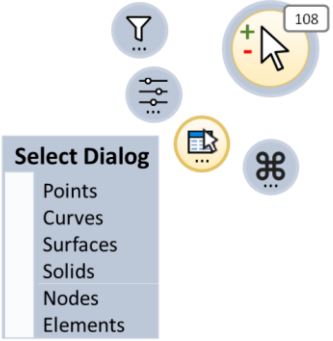


Figure 13 — Selection widget Select Dialog menu

Commands – The final menu is a commands menu that will show commonly used commands for each entity type selected (recall that commands are only applicable to unique entity types). Serialized command history is recorded via telemetry, so a Markov Chain model could be used for simple command prediction given the current state of selected entities and the previous few commands that have been executed by the user. This predictive model could be based on crowd-sourced user behavior but also trained locally on the user's machine to learn their habits and improve prediction over time.

Because it is not possible to display all related commands, an additional option would exist for all entity types that would display a list of all compatible commands.

Finally, a list of recent commands is also available to the user should they be doing numerous repetitive operations.

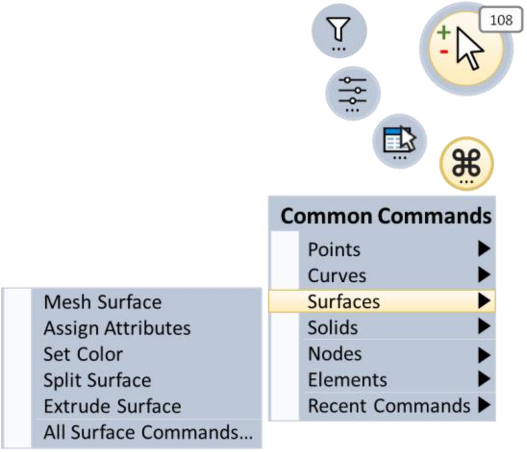


Figure 14—Selection widget Common Command menu, showing Surfaces submenu

5 INTERFACE JUSTIFICATION

Discoverability – One of the main weaknesses of the current interface the redesign is meant address is discoverability. By placing the selection widget in a visible location on the primary screen and enabling it by default, it encourages the user to interact with it.

Furthermore, the addition of displaying commands related to the current selection can aid users in finding the command they need or even learning about the existence of a new command without having to go through hundreds of commands via the menu or toolbar structure.

Direct Interaction – By allowing users to select entities first and providing a set of available commands, this gives the user the impression that they're directly manipulating something that they've selected. Also, implementing a noun/verb interaction paradigm is intended to give users the feel that they're directly operating on something.

Consistency – Many users indicated that they were already used to the Femap selection interface. For users that don't want to learn something new, consistency is maintained by allowing the user to completely disable the new interface – their workflow can continue as it always has.

Additionally, the selection widget is designed to be as consistent with existing functionality as possible. For instance, the capability to select a single entity type in advance of running a command already exists – this selection widget just builds on that capability by making it always on and providing the ability to select multiple entity types. If a command is selected with a compatible active selection, the selection dialog will simply be short-circuited and use the current selection. It is also consistent with existing functionality by integrating into the *Model Info* tree.

Even if a user does want to use the new selection widget, by providing an option to augment its capability with the existing entity selection dialogs, this further enhances consistency.

The iconography chosen was meant to be consistent with that used elsewhere in Femap and for cases where an equivalent icon didn't already exist, one analogous to other programs or interfaces (such as the option icon and the Mac command icon) were chosen.

Finally, the "always-on, select anything" nature of the proposed selection widget is meant to introduce a more consistent usage paradigm with other programs, especially CAD programs.

Flexibility – Flexiblity is addressed in a number of ways, such as not constraining the types of entities a user can select, but also providing filters so they can refine the operation to how it fits their workflow. Also, as previously mentioned the ability to disable the selection widget entirely or augment its functionality with the existing selection dialogs provides the user with additional flexibility.

Feedback – User feedback regarding what is selected is improved by switching from the carats to indicate selection and highlighting it on-screen instead (Figure 9). Additionally by updating the selected entity count directly and immediately on the widget, the user is given immediate feedback as to the results of their actions.

Tolerance – The new interface addresses the intolerant error messages (Figure 5) by eliminating them altogether. Not having an active dialog means that there is no field that would require focus and no need to raise the error.

Constraints – The old selection interface addressed constraints in a pretty effective manner by constraining the types of entities that could be selected to the command selected. The redesigned interface implements the same constraints in a different way – relevant commands are shown to the user based on the selected entities but does so in a way that makes the commands more discoverable.

Learning curve – The selection widget is intended to reduce the learning curve by implementing a usage a paradigm that is more consistent with other programs a user may be more familiar with. The discoverable nature of the widget itself, encourages the user to experiment with it and removing intolerant errors is intended to prevent the user from getting discouraged and spending more time trying to learn the interface.

Also, as previously mentioned, the increased discoverability of commands by displaying ones related to the selection reduces the need for a user to have to wade through the menus and toolbars to even learn that functionality they need exists. The predictive commands feature is meant to infer from the user what they may want to do next, negating the need for them to learn the exact location of a particular command.

Expert blind spot – this interface doesn't directly address this issue, however it is indirectly addressed by performing needfinding exercises as well as gathering feedback from actual users as part of the design lifecycle (the evaluation plan), before implementing a finalized feature in production.

6 EVALUATION PLAN

While this project focused on an improvement on the existing entity selection interface, there are a number of reasons that empirical evaluation methods may not be appropriate in this stage of the design lifecycle. Because the design is somewhat of a large change to the existing interface and no users have been exposed to it, all users, new or experienced, will have an initial learning curve. This can potentially introduce a lot of variability into the empirical results, leading to misleading results.

What is more important in this phase is determining simply if the new design makes sense to the user, is easy to understand, and if they're encouraged to use it. Qualitative evaluation to determine how a user thinks this interface will work for them is going to provide more useful feedback. If the feedback is generally positive, an improved design in subsequent iterations or A/B testing on alternatives of the same selection method (i.e., button placement, icon colors) are better candidates for empirical testing.

To perform the qualitative evaluation, each participant will be shown a series of prototypes and asked follow-up questions in accordance with the following script.

1. [User is shown wireframe prototype from Figure 7]
2. Q: "We've added new functionality to the user interface. Looking at this prototype, can you tell where it is and what its purpose is?" [collect user feedback]
3. A: [If correct] "That's right, we've added new functionality intended to make entity selection easier for all users, new and experienced." Q: [If incorrect] "The widget in the upper right corner is intended to make entity selection easier for new and experienced users. Can you describe what about the design made you think it was for a different purpose?"

   [All] "The idea is for the selection widget to be an 'always-on' selector that is enabled by default. You can just pick on-screen entities first without having to go into a command. Any entity type may be selected; you're not constrained to a single type."
4. Q: "Does this address a need or shortcoming you feel you have within Femap?" [collect answer]
5. Q: "How would you expect to interact with the concept of an 'always-on selection interface' as shown to this point?" [collect feedback]
6. [User is shown prototype from Figure 10, without the text labels]

7. Q: “When you hover over the widget, you get a flyout menu of buttons presenting different options. Just looking at the icons, what do you think they do?” [collect feedback]
8. A: “Counter-clockwise from the top left, the icons are for filter rules, selection options, access to the existing selection dialogs and common commands related to selected entities.”
9. Q: “Keeping in mind that this was not intended to replicate all options of the current selection dialog, do you feel this grouping of functionality is sufficient? Do you think it’s important to replicate every last feature within the selection widget?” [collect feedback]
10. [Show users Figures 11-13; hover menus for each icon, except commands]
11. Q: “These menus will appear when you hover over each icon. Do you think the functionality proposed in these menus sufficiently captures the options you would want to see in a live selector? If not, what else would you want to see?” [collect feedback]
12. [Show users Figure 14, command menu]
13. Q: “The command menu is a little different. We plan on using AI as well as individual personalization to come up with a list of commonly used commands based on the current selection. Do you think you would use this concept of accessing commands or would you prefer to use the existing toolbar/menu structure (which is fully compatible with the selection widget)” [collect feedback]
14. Q: “Do you find the concept of being able to select any entity type and being shown only relevant commands to be intuitive?” [collect feedback]
15. Q: “Do you find the fact that the selection widget exists in the viewport to be distracting?” [collect feedback]
16. Q: “Overall, what do you think about this proposed selection method? Do you feel it is an improvement, about the same or worse than the current section method?” [collect feedback]
17. Q: “Do you feel you would use this entity selection method or give it an opportunity to augment your existing workflow, or would you simply disable it, given the option?” [collect feedback]
18. Q: “Do you have any additional feedback you would like to provide?” [collect feedback]
19. “Thank you for your time”

One major difference between the participant evaluation phase and initial needfinding is the participant pool must be significantly larger. This new interface is a fundamental redesign of one of the most basic levels of functionality within the program. It is important that feedback be gathered from a much more diverse user population to minimize the likelihood of type I or type II errors, assuming the null hypothesis is that the proposed interface is an improvement over the existing interface.

Participants will be recruited from a pool of existing users with varying levels of experience with Femap. Users with no experience with Femap will also be recruited, however they will only be presented with steps 1-7 of the interview script simply to gauge if this method provides an inherently intuitive way of interacting with the program.

If possible, to minimize bias the evaluation script would be reviewed / modified by a neutral third party as well as administered (first person pronouns such as "we" would change to third person, such as "the developers").

7 APPENDIX A – AGGREGATE INTERVIEW RESULTS

1. How long have you been an engineer?
   1, 14, 20, 35, 5, 3, 17

- Min/Max: 1 / 35 years
- Mean: 13.5 years
- Median: 14 years

2. What other finite element pre/post-processors do you have experience with?

- MSC Patran (5/7)
- ABAQUS (4/7)
- Altair Hypermesh (3/7)
- Siemens SimCenter (3/7)
- ANSYS Workbench (2/7)
- LS-Dyna (1/7)
- ANSYS Prep7 (1/7)
- MSC Apex (1/7)
- Other CAD embedded CAE (5/7)

3. How long have you been a Femap user?
   1, 10, 20, 15, 5, 3, 10

- Min/Max: 1/20
- Mean: 9.2 years
- Median: 10 years

4. How often do you use Femap?

- More than 3 times/week: 4/7
- 1-3 times a week: 1/7
- 1-3 times a month: 1/7
- Less than once a month: 1/7

5. Could you describe your general impressions of entity selection in Femap?

- I'm brand new with Femap, so I felt it was hard for me to learn. I'm glad I had [others] around me that could explain it to me, because I really wasn't sure what to do at first
- I'm most experienced with Patran and honestly Femap's selection is intimidating. Once I figure it out, it's okay, but there's just too many options for me
- I've been using Femap for a really long time and I like that the interface has pretty much stayed the same the entire time. I like that I can do things the way I originally learned. For example, I can still remember the entity code for rigid elements is 29. I've gotten used to typing that in and it's good that I can still do that rather than having to pick from a list
- It's pretty powerful
- I found it kind of weird that you have to select a command first, then you can pick elements. For example in the CAD program I use, I can just highlight stuff and right click to get a list of available commands. I feel like with Femap, you have to know where stuff is first. Sometimes I want to just pick stuff and see what I can do but I guess there's not really a way to do that. If I have to switch back and forth between Femap and say SimCenter, Femap is the one that I have more trouble remembering how to use
- I like it. It was pretty easy for me to figure out
- I don't have any complaints really. I guess if I had to say anything it would be that I don't really get the point of having a long list of elements shown, especially when you're picking hundreds or thousands

6. Do you remember how long it took you to get accustomed to entity selection in Femap / did you find there to be a difficult learning curve?

- A few hours (1/7)
- A few days (4/7)
- A few weeks (1/7)
- More than a month (1/7)
- 5/7 users reported a difficult learning curve

7. Do you have any suggestions on how entity selection could be improved in order to better facilitate your workflow?

- It would be nice to be able to have multiple saved selection lists rather than just the previous one, I don't like using groups for that
- Previous should have more than just the last single selection. The vector and plane manager lets you do at least 5, can we get that for all selection?
- I really don't like how when you select elements it just marks them on screen. I usually go and highlight them manually but after you do that and refresh the screen, you lose everything, so I have to do it all over again to remember where I was
- No

8. Do you have any additional suggestions?

- Would be nice to have a more modern UI
- Too many things require too many clicks to accomplish something – I feel it can easily disrupt my workflow if I miss something
- I like that there's usually a bunch of different ways to get something done
- More YouTube tutorials
- I really wish I could interrupt a long running operation, like meshing. Sometimes I realize I did something dumb and I have to wait for the thing to complete first before I can undo
- I don't like when you guys add stuff, it's just buried under list of menu items there's no good way to know a new feature was added
- I feel like the new features that are added sometimes don't really work the same was as others, especially older ones in the program
- No

9. As I originally mentioned, I'm no longer a developer on the Femap team, so there's no guarantee the results of this project are going to be implemented in the program (sorry to get your hopes up if you liked the idea). Having said that, I'd like to share this project with them. Could I include your interview feedback in anything I provide? It would be a

> consolidated version of all the transcripts and no identifiable information will be included.

All participants provided consent.