Proposal for a Summary Feature to be added to PROPEL

Mentee: Valerie Gartland
Western Carolina University, Cullowhee, North Carolina
inhisimage@gmail.com
Mentors: Lori A. Clarke, George Avrunin
Laboratory for Advanced Software Engineering Research
University of Massachusetts at Amherst

ABSTRACT

This is a proposal for a new feature to be added to the PROPEL ("PROPerty ELucidation") program. This new feature will give the user a summary of the project. The summary feature should make the user able to understand the big picture without loosing detail contained in their project. This should help to minimize mistakes that may arise from misunderstandings about a project as a whole.

1. INTRODUCTION

PROPEL is a program designed to help software developers to clarify the properties of the system they are creating. With PROPEL, the user can define a property with clarity and mathematical precision, without sacrificing understandability. PROPEL is detail-oriented, and uses a both finite-state automata and disciplined natural language with the goal of representing properties exactly as the user intended.

With all of the information that PROPEL represents for each property, it can be difficult to understand the "big picture" from a set of multiple properties. PROPEL organizes properties with a hierarchy that can have varying levels. To understand the big picture, it is important for the user to understand this hierarchy.

Because of these complications, we found PROPEL needed a feature that would give the user a big picture. This feature should show the hierarchy without neglecting the detail.

2. PROPEL HIERARCHY

To understand the summary feature, it is necessary to understand the hierarchy used in PROPEL.

At the top level is the main *project.* This contains all of the properties and the rest of the levels in the hierarchy.

Within the main project, there can be an unlimited number of *subprojects* in nested layers.

Within the project and inner subprojects, *alphabets* hold a set of events that relate to each other.

In each alphabet, *properties* define how the events relate to each other. Multiple properties within one alphabet can use the same event.

Events are contained in alphabets and defined by properties.

3. NEED FOR THE SUMMARY FEATURE

Problems could arise for various reasons when using PROPEL.

- Since a project could have hundreds of events and properties, the user could lose track of events as they are created. A particular event could be used in many different properties within one alphabet. In that case, the event should be treated the same way in every property. However, as more events are created, inconsistencies may arise if the original meaning of the event is forgotten. The user must be aware of every property in which any particular event appears in order to understand the full meaning of the event.
- Another possible problem is that events can be created within an alphabet and never become applied in any property. This is fine if it is done on purpose, but if an event is accidentally "un-parameterized," it can affect the meaning of any properties that refer to all the events in the alphabet.
- When presenting or explaining a PROPEL project to someone else, it is hard to give an overview without a summary feature. To see the detail within properties, each property must be opened. Each property takes up a good deal of the screen, so it gets very crowded to have many open at once.

4. DESIRED ATTRIBUTES

The summary feature should prevent problems like these. There are two main focuses behind the design of this summary feature. We want to provide an overview of the structure of the project. At the same time, we wanted to make the details of each property available.

These are the categories that will be available for a summary:

- Project Name
- Project Comments
- Subproject Names
- Subproject Comments
- Alphabet Names
- Alphabet Comments
- Property Names
- Property Comments
- Event Names
- Event Comments
- Behavior Names
- Behavior Parameters
- Scope Names
- Scope Parameters
- "Is Disjunction Of"

As for the user interface, it should be as easily accessible as PROPEL itself, but at the same time it should give the user advanced capabilities. This required us to consider two different viewpoints at once. We could not allow the advanced features to confuse the user, but we could not limit the advanced users in favor of inexperienced users.

A flexible mode of display was also desirable. So many details would be made available with this feature that we wanted the user to choose which information to display.

5. PROPOSED SOLUTION

PROPEL summaries are displayed in a tabular format. To provide a flexible display that is still clear and still makes sense, the summaries are organized by a perspective. The perspective of a summary defines which information is most important in a table. There are four different perspectives, and each one refers to a different level of the PROPEL hierarchy. Choosing a different perspective will give the summary a different focus. These perspectives are:

Project – This focuses on the overall hierarchy of the project. The project name is the largest cell. See Table 1.
Alphabet – This is basically a rearrangement of the project perspective, with the alphabets as the largest cells instead. See Table 2.
Property – This has the most inner detail. Here, the behavior type, behavior parameters, and scope of each property is shown, as well as an indication of each property's location in the PROPEL hierarchy. See Table 3.
Event – This shows you the project from the event's point of view. One column contains the hierarchy of each property that uses the event. Another column indicates any events that are a disjunction of a particular event. See Table 4.

There are two different methods for creating a summary in this design. For the users who want something simple and quick, there are predefined summaries. These summaries only require the user to select the perspective, and then a summary will be automatically built. Tables 1-4 are examples of predefined summaries.

The other method for creating a summary is the summary builder. This is more advanced and more flexible. After choosing a perspective, the user can select the information that will be displayed in each column, step by step.

After creating a summary, the user can alter the display in almost any way. The information can be sorted, filtered, and rearranged. This provides the desired flexible display. The user may also hide whole columns to keep the table from becoming too crowded.

When the user alters the display, the information itself does not change. The information shown in the summaries is intrinsic to the project itself. This information is not generated for the summaries, rather the summary provides a way to organize and display the preexisting information.

There is an option in the PROPEL summary feature to show all of the information that is available a particular summary. This is very large and complicated, but if the user wants to display all of the information at once, they have this ability. This option includes every category that was listed earlier in the report. The table this generates is too large to include in this report, but it is included in the presentation on this feature.

The PROPEL hierarchy was illustrated in two ways. In Table 1, the columns are grouped to create a nested look. In Tables 2-4 the names of the levels are combined to show the location of a particular alphabet, property, or event.

Some columns in the summaries are colored. This option is provided to help the user see the hierarchy more clearly – blue for projects, purple for subprojects, red for alphabets, and green for properties. This can help the user to spot un-parameterized events, too. For instance, in Table 4, push_ok, steer, and turn are not applied in any property, so there is no green in their context. This can be seen at a glance.

6. CONCLUSION

This feature is in the process of being added to PROPEL, so there may be some changes to this initial proposal. Still, this design should meet the stated needs. The user interface is designed to be easily accessible, but to also provide advanced options. Formatting summaries according to perspective makes the information more organized. This feature should also make PROPEL projects easier to present.

7. ACKNOWLEDGEMENTS

Thanks to Lori A. Clarke, George Avrunin, and Rachel L. Smith for their guidance, support, and patience.

7. REFERENCES

[1] Rachel L. Smith, George S. Avrunin, Lori A. Clarke, Leon J. Osterweil, "PROPEL: An Approach Supporting Property Elucidation," Department of Computer Science, University of Massachusetts, Amherst, MA 01003.

APPENDIX

Table 1

Project Summary						
Project	**Subprojects**		**Alphabets**	**Events**	**Properties**	**Comments**
Main			Subway	arrive@station open_doors	station_arrival	Comments
	Hotel	Lobby	Elevator	arrive@floor called@floor elevator_stop open_doors second_arrival	floor_arrival open_on_floor	
		Service	Elevator	arrive@floor called@floor elevator_stop open_doors second_arrival	floor_arrival open_on_floor	
			Golf Cart	steer_left turn_left	turning left	
	Virtual		Errors	push_ok error_display ok_button	error_display	
			Queues	dequeue empty enqueue	empty	

Table 2

Alphabet Summary			
Project.Subprojects.Alphabet	**Events**	**Properties**	**Comments**
Main.Hotel.Lobby.Elevator	arrive@floor called@floor elevator_stop open_doors second_arrival	floor_arrival open_on_floor	Comments
Main.Hotel.Service.Elevator	arrive@floor called@floor elevator_stop open_doors second_arrival	floor_arrival open_on_floor	Comments
Main.Hotel.Service.Golf Cart	steer steer_left steer_right turn turn_left turn_right	turning left	Comments
Main.Virtual.Errors	error_display push_ok ok_button	error_display	Comments
Main.Virtual.Queue	dequeue empty enqueue	empty	Comments
Main.Subway	arrive@station open_doors	station_arrival	Comments

Table 3

Property Summary				
Project.Subprojects.Alphabet.Property	**Behavior**	**Behavior Parameters**	**Scope**	**Comments**
Main.Subway.station_arrival	RESPONSE	action: arrive@station response: open_doors	GLOBAL	Comments
Main.Hotel.Lobby.Elevator.floor_arrival	RESPONSE	action: second_arrival response: elevator_stop	BETWEEN	Comments
Main.Hotel.Lobby.Elevator.open_on_floor	RESPONSE	action: arrive@floor response: open_doors	AFTER	Comments
Main.Hotel.Service.Elevator.floor_arrival	RESPONSE	action: second_arrival response: elevator_stop	BETWEEN	Comments
Main.Hotel.Service.Elevator.open_on_floor	RESPONSE	action: arrive@floor response: open_doors	AFTER	Comments
Main.Hotel.Service.Golf Cart.turning left	RESPONSE	action: steer_left response: turn_left	GLOBAL	Comments
Main.Virtual.Errors.error_display	RESPONSE	action: error_display response: ok_button	GLOBAL	Comments
Main.Virtual.Queue.empty	EXISTENCE	action: dequeue	BETWEEN	Comments

Table 4

Event Summary			
Event	**Project.Subprojects.Alphabet.Property**	**Is Disj Of**	**Comments**
arrive@floor	Main.Hotel.Lobby.Elevator.floor_arrival Main.Hotel.Lobby.Elevator.open_on_floor Main.Hotel.Service.Elevator.floor_arrival Main.Hotel.Service.Elevator.open_on_floor		Comments
arrive@station	Main.Subway.station_arrival		Comments
called@floor	Main.Hotel.Lobby.Elevator.floor_arrival Main.Hotel.Lobby.Elevator.open_on_floor		Comments
dequeue	Main.Virtual.Queue.empty		Comments
elevator_stop	Main.Hotel.Lobby.Elevator.floor_arrival Main.Hotel.Lobby.Elevator.open_on_floor Main.Hotel.Service.Elevator.floor_arrival Main.Hotel.Service.Elevator.open_on_floor		Comments
empty	Main.Virtual.Queue.empty		Comments
enqueue	Main.Virtual.Queue.empty		Comments
error display	Main.Virtual.Errors.error_display		Comments
ok_button	Main.Virtual.Errors.error_display		Comments
open_doors	Main.Hotel.Lobby.Elevator.open_on_floor Main.Hotel.Service.Elevator.open_on_floor Main.Subway.station_arrival		Comments
push_ok	Main.Virtual.Errors		Comments
second_arrival	Main.Hotel.Lobby.Elevator.floor_arrival Main.Hotel.Service.Elevator.floor_arrival		Comments
steer	Main.Hotel.Service.Golf Cart	steer_left steer_right	Comments
steer_left	Main.Hotel.Service.Golf Cart.turning left		Comments
turn	Main.Hotel.Service.Golf Cart	turn_left turn_right	Comments
turn_left	Main.Hotel.Service.Golf Cart.turning left		Comments