Property Models: Algorithms for User Interfaces

Travis Kosarek
Texas A&M Undergraduate Student
travis.kosarek@gmail.com

Dr. Jaakko Järvi
Texas A&M Faculty
jarvi@cse.tamu.edu

John Freeman
Texas A&M Ph.D. Student
jfreeman@cs.tamu.edu

Abstract
The purpose of user interfaces in modern applications is to assist users in specifying parameters to execute a set of commands. It is common to find interdependencies among these parameters. As a result, user interfaces in general are often complex, incomplete, or difficult to implement. Property models attempt to alleviate this by increasing reuse in user interface programming. Property models employ a constraint system and reusable generic algorithms to do away with ad hoc “event handlers.” Until now, property models have been applied only within desktop applications. This paper describes how the property model approach can be extended to web applications.

The primary goal of my research is to determine the best architecture for running property model-based user interfaces over the web. To facilitate my research, I created a Rapid Application Development (RAD) framework to allow convenient development and testing of such interfaces. This framework constructs an interface from the following: (1) a declarative specification of a property model, and (2) a layout description. Any change in the property model or the layout description is immediately reflected in an interactive user interface generated by the framework.

As a result of this project, it is now possible to conduct experiments on property model-based user interfaces in web applications. I hope to show that adapting property models to the development of web applications will result in increased reuse and reduced defect rates, similar to what has been observed with desktop applications. Larger scale experiments to validate these conjectures remain as future work.

Motivation
The purpose of an interface is to assist the user in specifying some parameters in order to execute a command.

It is common to find interdependencies between parameters. As a result, it is difficult and costly to express the behaviors of an interface correctly and unique ad hoc solutions are used.

Property Models
A reusable model for the composition of components
Reuse has been applied in other areas of computer science to improve upon ad hoc solutions in the form of algorithms and libraries.

We use a constraint system as a reusable model for the composition of components in user interfaces.

A constraint system consists of two parts:
1. A set of variables tied together by constraints
2. Methods that enforce each constraint

Benefits of Our Approach
• Reusable user interface behaviors
• Manage network of relationships
• Enable/disable widgets
• Record and playback of scripts
• Clear separation between data model and presentation
• Allows for multiple controls to easily be attached to the same data
• Declarative specification of the interface
• Greatly reduces code written in large applications

References

Web Based Applications
Until now, property models have only been applied within desktop applications. I extended the use of property models to web applications.

I created a Rapid Application Development (RAD) framework to allow convenient development and testing of property model-based web interfaces.

This framework constructs an interface from the following: (1) a declarative specification of a property model, and (2) a layout description. Some of the features of this environment include instant updating, error checking, and an interactive live result.

Future Work
It is now possible to conduct experiments on property model-based user interfaces in web applications. I hope to show that adapting property models to the development of web applications will result in increased reuse and reduced defect rates, similar to what has been observed with desktop applications. Larger scale experiments to validate these conjectures remain as future work.

Acknowledgements
I would like to thank my faculty advisor Dr. Jaakko Järvi for all of his help with fine tuning my writing this summer. I would also like to thank John Freeman for helping guide my research throughout the entire summer.