Integrating Event-Driven and Capsule-Oriented Programming

Introduction

Problem
Panini is a capsule-oriented programming model that helps developers take advantage of concurrency [3]. However, it is difficult for developers to use Panini when developing applications with common event-centric components such as user interfaces. There is no way for a capsule to ask another to notify it when something happens, such as a button click.

Solution
An integration of Panini and events would allow developers to more easily design concurrent systems that have event-centric components.

Methods
- Implemented event-driven capabilities into Panini, an implementation of the Panini model.
- Code is available at: https://github.com/jlmaddox/panini/tree/event
- Examined current literature to see how others have handled the challenges faced during this project.

Objectives
This project combines the Panini and event-driven programming models into a single, integrated model respecting Panini’s properties that make it easy and safe to use.

Results
The result is a version of Panini with event-driven capabilities.

Capsule Events
- Capsules can contain events, such as a button click
- Capsules can specify some code that handles events occurring

The Model
This is how the integrated model works:

Step 1: At startup, capsules subscribe to the events it is interested in. Multiple capsules can subscribe to a single event.

Step 2: The event occurs

Step 3a: Broadcast notification: E is announced and handlers H1 and H2 are notified. Send the same message to both simultaneously.

Step 3b: Chain notification: E is announced and H1 or H2 may modify the message. Send the message to one and have it pass it on.

Challenge
This honors research project has led to the following findings:
- It is possible to integrate event-driven and capsule-oriented models into a single model to take advantage of both
- The missed announcements problem can be addressed reasonably by for example simply assuming that some might be missed or by requiring all capsules to start up before executing any code or handling messages
- The integrated model allows developers to create concurrent, thread-safe software that contain event-centric components in a clean manner

Conclusion
This project combines Panini and event-driven capabilities into a single, integrated model respecting Panini’s properties that make it easy and safe to use.

Reference

This work was supported in part by US National Science Foundation (NSF) under grants CCF-08-46059, CCF-11-17937 and CCF-14-23370. The opinions are those of the authors and do not reflect the views of the NSF.