

Christian Queinnec - In Influence of Browsers on Evaluators or, Continuations to Program Web Servers

Comments on:

Queinnec, Christian (2000). The Influence of Browsers on Evaluators or, Continuations to Program Web Servers. *ICFP 2000 - International Conference on Functional Programming* (September, 200, Montreal, Quebec, CA), p. 23-33.

It seems like the problems to the suggested solution under section 3.3 (Orphan) can be addressed by using frames or iframes in the same window to display two different results. Hence, if we have prior knowledge that a request to a computation produces two results as indicated by the code

```
(fork (show page1)
      (begin (sleep 10) (show page2)))
```

we can open a window with two frames and display the results in one frame each.

In this case, if a user, unaware of a pending second result, closes the window the second result will not be shown to any "non-vetted users". Also, this makes it easier for a user to expect a second result without having to implement any "smart flashy icon" with JavaScript.

You are correct that frames seem like an appropriate alternative solution, unless you really want separate windows. I had thought the same.

Queinnec said, "For instance, game where two players exchange their positions may be implemented by a simple exchange of continuations." The idea of continuations in games seems particularly useful. Do you know if continuations are a common technique used in games? It seems particularly applicable to online gaming or saving the state of a game.

It is my understanding that few systems use continuations. As you've seen from the initial readings, they can seem a bit confusing and sound inefficient. They also assume that the system keeps running. (You can't, for example, write a continuation to disk.)

This reading made me more appreciative of the use of continuations. Reading about critical sections and the use of threads, it seemed to me that these are some of the synchronization issues that also come up when creating operating systems. This may be silly, but are continuations used in developing operating systems?

You were hitting some aspects of the paper that most reflect the Web server aspect, which means multiprocessing. I expect that continuations are not used frequently in OS.

Also, how come Java does not support continuations? Is it possible/common for people to simulate continuations in Java by using Java threads?

Continuation use is uncommon in general, outside of the Scheme/Lisp (and, perhaps, Smalltalk) communities. The main problem, as I understand it, is that implementing continuations well means that you should not rely on the standard "One stack + One heap" architecture that most of us were raised to use. You are right that each thread seems to have its own stack, so threads might provide an opportunity.

Why is it uncommon to invoke a continuation multiple times in a sequential context, and more common to invoke a continuation multiple times in a concurrent context?

It's not clear that there is a difference. Quiennec is simply suggesting one context in which it makes a lot of sense to reuse the same continuation multiple times. The concurrency is simply another aspect of the same environment, but does not really relate to the reuse.

Is there any possibility of two requests from one user/browser arrive at a server concurrently? I would imagine some delay due to human limitations (It is not possible to submit a request on two different browsers (one of which is a clone of the other) at the same time. Even if you write a script for this particular purpose, I would imagine that there would be a microsecond delay.

It's actually not something that you have to worry about in this level of the communication stack. Lower levels handle the potential concurrency, and, in essence, send each request in sequence.

After reading this paper, I now see the value of using continuations. Continuations do have some disadvantages though. Take a situation where a user is filling in a multi-page questionnaire using a browser. With continuations, when a user hits the back button in the browser, he/she may not want to get rid of the contents of the page that they are in, but may just want to change the responses to the previous page. Is there any simple remedy for this?

That depends, in part, on the browser. Many browsers will simply fill in the fields with their previous values. But no, the thing you're suggesting is more a "Update my Entries" type of forward motion.