

Cooperative Task Management without Manual Stack Management

or, Event-driven Programming is Not the Opposite of Threaded
Programming

Atul Adya, Jon Howell, Marvin Theimer, William J. Bolosky,
John R. Douceur
Microsoft Research

Presented By

Amit Bhat

CS533 – Spring 2009

Concepts

Task Management

- Serial : Task finishes and then yields control
- Preemptive : Control can be preempted from a particular task
- Cooperative : A task yields control at well-defined points during execution

Concepts

Stack Management

Automatic : Using program stack provided by the procedural Language

Manual : Programming language procedures ripped into different Event Handlers

Event Handlers

E1

.

.

initiate I/O

register a Continuation

(data, E2)

Terminate

Event Handlers

E2

Code after I/O completion

.

.

Terminate

Manual Stack Management Drawbacks

- The program becomes less readable
- Difficult to debug (Stack Unrecoverable)
- Two or more language functions represent a single conceptual function
- More problems with I/O in functions having loops
- Software Evolution

Ideal Approach

- Task Management

Cooperative

- The middle ground between Serial and Preemptive styles

- Stack Management

Automatic

- For its readability and lack burden on the programmer

What people at Microsoft Think?

- Two groups of developers
 - One supporting Manual Stack Management
 - The other strongly advocating the Automatic Stack Management
- Pieces of Code exist written using both the styles of Stack Management
- Need to find a glue!!

Solution

- Cooperative Tasks - Fibers and Threads
- Manual Calling Automatic and Automatic Calling Manual
- Use Adaptors to reconcile

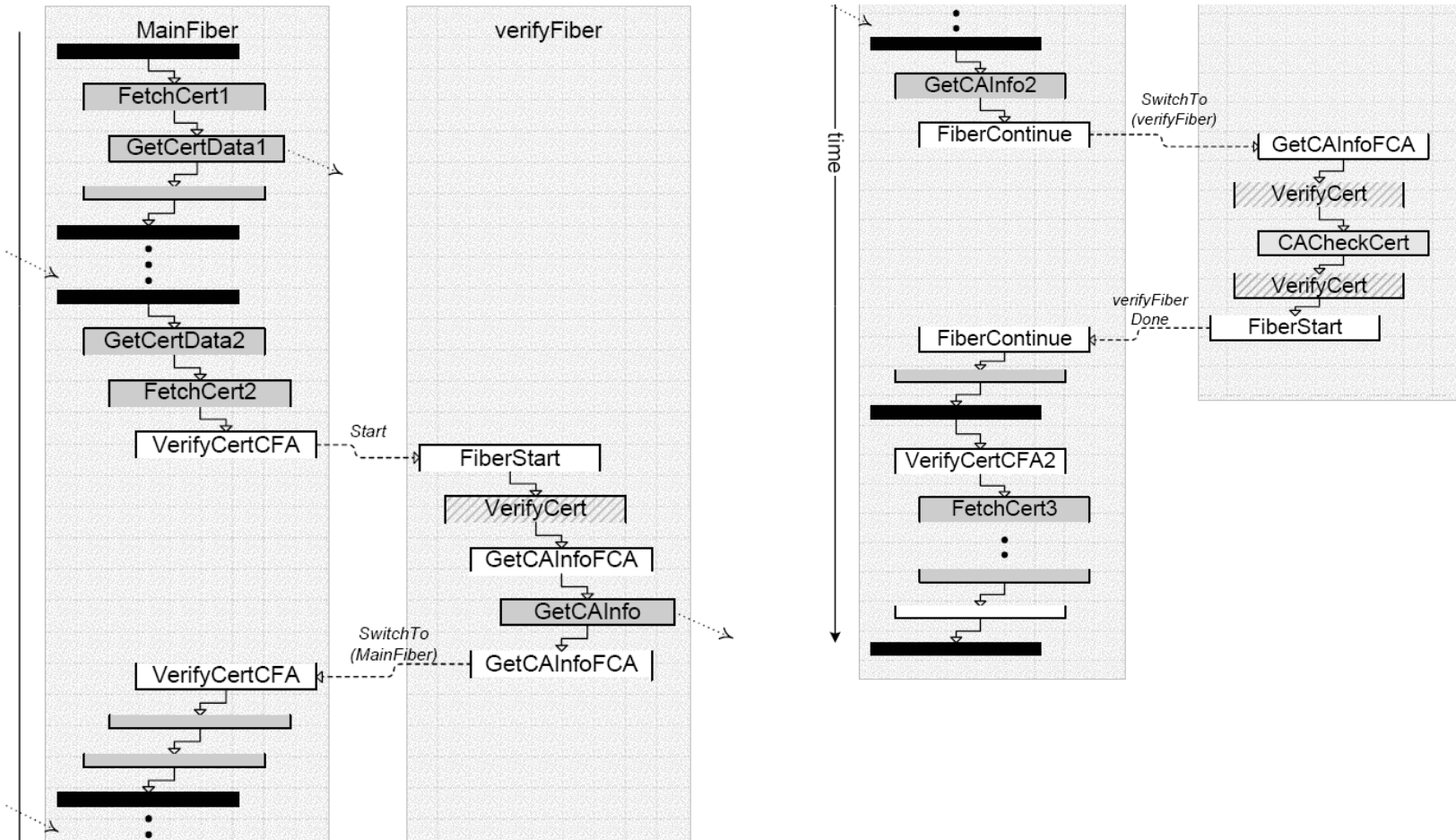
Solution

Manual

Automatic

Manual

Automatic



Implementation

- Farsite
 - A serverless file system
 - Runs as a daemon on Windows NT OS
- Ucom
 - A wireless phone application
 - Designed for Windows CE

Conclusion

- Issues of Task Management and Stack Management can be considered independently of each other
- Code pieces written in Manual and Automatic Stack management styles can coexist in a program
- This is achieved at the cost of embedding adaptors in the original code; not always desirable!