

CAIR Seminar

1pm Wednesday 11th March 2009

Programming without Programs!

Lyndon Evans

Computers are undoubtedly an essential part of modern society, but without the software which runs them, they would be of little use to us. A user starting up a PC is soon presented with an operating system and applications designed to increase creativity, productivity or just provide entertainment. This software is the result of hundreds or thousands of hours of development work by teams of computer programmers.

In theory, programming sounds rather simple; use a programming language's features to construct a set of ordered instructions (a program) which will do the required tasks. In practice however, getting the program to do something useful in the first place, and figuring out the problem when it doesn't perform as expected, soon gets rather complicated.

Fortunately for most users, they need not concern themselves with this complexity – just use ready-built applications through their user interfaces, as they have been designed to do. Useful though these applications can be, it soon becomes tedious for the user if tasks performed within or between them become repetitive. Many applications are designed with the capability of being automated, but how is this achieved? What if the user wants to automate more than one application in order to achieve the desired result?

In both cases, this can be done using a set of ordered instructions designed to run these applications automatically. These instructions need to be written in a programming language, following its rules and syntax – in other words a program. Back to square one!

This talk discusses the issues involved in equipping a user with a way to design, build and control an automation process, while attempting to shield the user from the complexities of programming. It has many challenges – and some interesting implications!