The promise of scripting languages for business process automation

Spurred on by all the dire security predictions recently on the use of Internet Explorer, I thought I would try some alternatives and impart what useful experience I could before continuing with a more optimistic theme for this month.

There seem to be two main problems currently with the average home Windows machine. First of all using Internet Explorer is like being trapped in a Jack in a Box factory during an earthquake. The second concerns interlopers gaining clandestine access to your machine. XP SP2 may or may not alleviate these so in the meantime, I downloaded a beta copy (0.92) of Mozilla Firefox, www.mozilla.org and a personal copy of ZoneAlarm, www.zonelabs.com both without charge. Why oh why did I not do this sooner ? Firefox is already gratifyingly robust on even very Windows centric sites and controls pop-ups with ease. Zone Alarm is equally impressive and together they improved my Windows browsing 'experience' immeasurably.

Now to business. So much of what I read these days is negative with security worries, scams, spams and the rest so I thought I would talk about something positive, the use of scripting languages in automating business processes. For years now, proponents of graphical user interfaces and desktops with Apple and Microsoft in the vanguard, have been trying to lead users away from scripting languages, ("Those command line programs are so clumsy and nerdy, just look at this glossy button-riddled interface ..."). I was reminded of this when working on test process improvement in a big European company recently. They are trying to glue together a spectacular mix of Word files, Excel files, Project files and miscellaneous others of various generations. The documents are in almost all cases very simple but they are forced to use the various graphical user interfaces to generate them. The whole thing is time-consuming, error-prone and highly resistant to either automation or data mining. Of course the central point about the really successful modern scripting languages such as Tcl/Tk, Perl, Python and PHP is that they are spectacularly good at prototyping and automating processes and they are completely portable across Windows and Linux environments. A suitable choice of a portable text-based document format, HTML or one of its more sophisticated siblings, maybe SQL database access, a few hundred lines of scripting language to glue it all together and it is simply astonishing the levels of custom-built automation which can be achieved reliably and quickly. Some of these languages also include the ability to build a graphical user interface portably, such as Tcl/Tk or Perl/Tk, so that the user can have a simple purpose-built graphical interface sitting on top.

Let me give you some of many examples: a sophisticated multi-currency dual-entry accounting system in 5000 lines of Tcl/Tk; a complete live multi-user athletics track and field results management system with automatic report generation, customised e-mailing and web-updating in 1500 lines of PHP with MySQL; automation of many of the requirements of the widely used Carnegie-Mellon Capability Maturity Model for software development in 2500 lines of Tcl/Tk and 1500 lines of shell-script. These are projects requiring only a few person weeks including the graphical interface.

Custom tool building fuelled the industrial revolution; we may see its like again.

lesh@oakcomp.co.uk