Add-Fun:
PowerShell, a Soldering Iron, Some LEDs, a Synthesizer

Dave Bracken
Christchurch Polytechnic Institute of Technology
130 Madras Street
Christchurch, New Zealand
+64 3 940 8161
Dave.Bracken@cpit.ac.nz

ABSTRACT

This paper explores a potential non systems administration centric approach to teaching scripting using PowerShell.

Categories and Subject Descriptors

K.3.2 [**Computer and Information Science Education]**: Computer science education

General Terms

Languages.

Keywords

Scripting, programming, PowerShell, hardware, interfacing, libraries, objects, Arduino, microcontroller and embedded computing.

1. INTRODUCTION

PowerShell is Microsoft's systems administration and task automation scripting language [1] which has been available for both server and desktop operating systems since 2006. CPIT introduced their DICT540 "Introduction to Scripting" course in 2012 as a consequence of Microsoft placing greater emphasis on task automation using PowerShell. The Microsoft courses that expected PowerShell skills available at the time illustrated how tasks that traditionally would have been repetitively done using GUI tools could be automated through the use of command line scripts but fell short of explaining the commands used or their syntax. In a departure from the traditional Management console, Microsoft GUI tools such as Exchange 2010's Exchange Management Console [2] became a PowerShell script generator invoking the script rather than calling the underlying API calls directly. After examining Microsoft's PowerShell course [3] in 2011, CPIT staff recognized that an in house course needed to be developed as the Microsoft course prerequisites required the student to have systems administration experience. The current DICT540 course does not assume prior systems administration knowledge but through the use of practical examples uses common systems administration tasks as a vehicle for delivering scripting concepts.

When the DICT540 content was created at the start of 2012, the author viewed underpinning knowledge of the environment an important factor as the Microsoft centric courses that followed expected the student to be competent in the use of unfamiliar commands. The content of DICT540 subsequently directed the students towards learning how to write code in a scripting language whose list of commands (technically cmdlets) can be extended by hundreds at a time through use of modules (libraries). Initial emphasis was placed on concepts (objects, properties and methods), syntax, parsing and parameter passing before moving onto the more typical concepts usually found in teaching programming languages such as iteration, conditional statements, functions, exception handling and debugging. The variation in approach was that the author used practical examples of managing infrastructure services throughout the course and due to the prerequisite course being DICT440 which is delivered using Scratch, it was necessary to "set the scene" and subsequently give high level overviews explaining the concepts of directory services, network name resolution, file systems, registry and the management of both client and server based operating systems. The scene setting approach was adopted in order to satisfy the practical requirements of the Microsoft centric courses that followed.

As the author wanted to focus on content delivery, this position led the author to consider changing the requirements in the course descriptor and/or a change in delivery approach. In an ideal scenario courses that laid the foundations for the systems administration focus of PowerShell would be available but to date, none are. While course feedback from the students was positive, the author thought more could be done to focus on teaching PowerShell and lessen the emphasis on systems administration topics. This led the author to consider an alternative approach.

This poster paper appeared at the 4th annual conference of Computing and Information Technology Research and Education New Zealand (CITRENZ2013) incorporating the *26h Annual Conference of the National Advisory Committee on Computing Qualifications*, Hamilton, New Zealand, October 6-9, 2013. Mike Lopez and Michael Verhaart, (Eds).

2. AN ALTERNATIVE APPROACH

As some students will not major in topics to which task automation is applicable i.e. networking, systems administration, infrastructure management or the provision of services within cloud computing, the author decided to test the water by pitching a scenario which he believed encompassed common programming concepts and added a command line based FunFactor parameter. The concept was presented to the students and described a model that would embody the objectives of the course and gave examples of an alternative method of delivery that would retain the systems administration aspect but with an added fun component. The example of reading the unique serial number from an RFID enabled student ID card and using that as a key to look up that student's Facebook ID (a specially created one not their personal account) in Microsoft's Active Directory then retrieving the details of their favorite track from Facebook and finally finding said track in a filing system and playing it through a MIDI enabled synthesizer, all using the command line version of PowerShell was well received. Other examples such as interfacing with 20x4 lcd displays led and sending SMS (txt) messages via GSM boards were equally enthusiastically received by the students. Using a prototype, the author demonstrated how a musical notes could be played on a keyboard from PowerShell.

The author explained that the interface devices would be treated as black boxes subsequently would be pre-programmed and via a module, cmdlets would be made available that facilitated interaction using PowerShell.

Raspberry Pi was examined as the bridging interface but Arduino Uno was chosen as it popular in the Academic environment, affordable to students and cheaper to interface to real world devices. The interface to the Arduino is via a USB hosted virtual serial port which PowerShell can connect to through .Net. As a result of the author showing the Arduino boards to the students, most students expressed an interest in the boards themselves. Subsequently, the author would like to explore the possibility of an advanced hardware course being adapted to present the teaching of computer systems incorporating programming and networking through the use of microcontrollers.

As PowerShell was primarily designed for systems administration tasks, the author is not aware of any environments teaching it through the use of microcontrollers and MIDI devices.

3. THE NEXT STAGE

A request for hardware has been placed and the author hopes it will be available in 2014. Using Microsoft Visual Studio, a module containing custom cmdlets will be written [4 , 5].

4. CONCLUSION

While an alternative approach may well help maintain the motivation for keeping the students actively interested in a subject, the issue of the suitability of the prerequisite course is still present and should be addressed. PowerShell provides a rich programming environment while aimed at systems administration is versatile enough to facilitate scripts to interact with the physical world. Students seemed genuinely enthusiastic about being able to have their PowerShell scripts interact with physical objects. The new direction will be cheap to implement and the hardware can be used by other courses. All existing course objectives would be met and no modifications to the course descriptor are envisaged. A bonus was that some students are interested in learning more about embedded computing, a topic which the author has a long personal interest in.

5. REFERENCES

1 Microsoft. TechNet Library. Scripting with Windows PowerShell retrieved Sept 6, 2013 from http://technet.microsoft.com/en-us/library/bb978526.aspx.

2 Microsoft. Exchange Management Shell in Exchange 2010 retrieved Sept 6, 2013 from http://technet.microsoft.com/en-us/library/dd795097(v=exchg.141).aspx

3 Microsoft. Course 10325A. Automating Administration with Windows PowerShell 2.0 retrieved Sept 6, 2013 from http://www.microsoft.com/learning/en-us/course.aspx?id=10325a

4 Microsoft. MSDN. retrieved Sept 6, 2013 from http://msdn.microsoft.com/en-us/library/windows/desktop/dd835506(v=vs.85).aspx

5 Microsoft. Visual Studio. http://www.microsoft.com/visualstudio/eng/products/visual-studio-overview