Proposal – U-M Dearborn Senior Design Project, Spring 2023

NEATset – CONFIGURE program

The Network Architect and Engineering Toolset (NEATset) project is a series of programs designed to make the job of network architecture and engineering easier by automating many of the manual teaks that network architect and engineers perform on a daily basis. These include creation of device configurations, auditing of existing devices, repairs and updates to configurations, and so on. The previous NEATset project focused around three (3) tools – AUDIT, REPAIR, and DASHBOARD. This new project seeks to completely re-write the CONFIGURE tools to expand on the programs capabilities, make it more efficient, and increase the potential data sources that the CONFIGURE program can use.

CONFIGURE works with incoming datafiles containing network control parameters and combines these with intelligent templates to create unique sets of command files, from complete configurations to changes to existing configurations. This process is much more than simple data replacement. The heart of the entire process are the control constructs that can be added to a network template to make it more than just static text. Network Intelligent Templates, or NIT files, contain many of the same concepts as a modern 4th generation programming language, such as IF-THEN logic, SELECT-CASE logic, or FOR-NEXT loops. These NIT files can also allow the end user to create one to one and one to many relationships with secondary data files. NIT files can also call other NIT files, allowing for boiler-plating of a template using known and proven sub-templates.

NIT files are a form of a programming language, one that allows for the creation of a device configuration from the combination of configuration code, variable replacements, and intelligent processing of the templates. The combination of features found in NIT files effectively makes CONFIGURE the compiler, or more properly, the interpreter of these files, capable of generating a device configuration for each device listed in the primary data file. Thus, CONFIGURE can produce not just one configuration per run, but hundreds or thousands of such configurations during a single session.

In addition to a complete re-write of the CONFIGURE tool itself, there are three more components to this project:

1) Dockerizing the CONFIGURE tools so that it can run in a self-contained shell on virtually any operating system.
2) Creating a pre-compiler of the NIT templates to speed up assimilation of these files during a CONFIGURE session as well as to perform syntax checking on the basic constructs
3) Creating a template conversion tool to convert older Smart Templates to the new NIT format. This component is optional and would serve as a stretch goal for the team.

Ideally since there are some substantial libraries that would be used to support CONFIGURE, it should be written in PERL. The same is true for the pre-compiler. The template converter could be written in PERL or Python, but should also run from the DOCKER environment.

More details on the proposal can be found in the included PowerPoint presentation.

Thanks,

Mark Elias, M&M e-Services, LLC

586-241-3939 (mobile)