Unified Communications Deployment Tool

Shadan Saniepour Esfahani
Cisco Systems
Advanced Services
Unified Communications Practice
ssaniepo@cisco.com

Talal Siddiqui
Cisco Systems
Advanced Services
Unified Communications Practice
tsiddiqu@cisco.com

Abstract- This article elaborates on the architecture of a web based tool for large scale Unified Communications Deployment. Unified Communications Deployment Tool (UCDT) is geared towards reducing implementation cost and time by automating and standardizing the process. The tool allows enforcing design and deployment best practices by standardizing and automating the process that prevents the human errors. UCDT provides one platform to provision all the components of Unified Communications (UC) such as call agent/soft switch, phones, voice mail, presence, and other collaboration platforms in the network, and hence eliminates the need for adopting multiple tools for the UC deployment. UCDT is accessible over the network that allows performing the deployment activities from remote locations and save travel costs.

I. INTRODUCTION

Unified Communications (UC) is gaining more popularity by small and large businesses, and more enterprises are adopting UC as their main mean of collaboration. While adopting UC has enormous advantages [4][5][6], deploying UC involves some challenges, namely, implementation cost and time. Unified Communication networks are becoming complex as they include soft switches, gateways, intelligent switches to facilitate media transport with QoS settings, network voicemail, integration with multiple vendor products, and other advanced features such as presence, web sharing. It is not simply enabling a dial tone to a desk phone. In a large enterprise there will be different line of business, each with its own set of requirements. This increases the complexity of planning and designing. This also leads to ensuring that a complex architecture with multiple devices and complex feature set gets deployed accurately. Unified Communications Deployment Tool (UCDT) is geared towards addressing these issues and reducing cost and time by automating and standardizing the deployment process.

UCDT is a web based application that allows users to access it remotely over IP network with proper credentials. Web applications are widely being adopted throughout the workplace due to their flexible nature [1][2][3]. The accessibility of UCDT to the target devices eliminates the need for users to be at the location where the deployment is taking place.

UCDT provides a single platform for provisioning multiple UC products. Network administrators are often concerned with the total cost of ownership of communication network and a unified tool to manage a wide range of devices and applications is highly desirable.

UCDT allows writing re-usable templates for provisioning requests, which can enforce implementation best practices. Once the templates are created and verified for one site, they can be used to provision different sites and no further verification is required. This eliminates verification time for every site, and minimizes the deployment time by utilizing re-useable templates. Furthermore, since the templates embed the intelligence, once they are finalized and put under version control, the deployment tasks can be performed by junior staff with limited technical knowledge.

Some of main advantages of the UCDT can be listed as:
1) Increases productivity by utilizing intelligent, re-useable templates.
2) Reduces implementation costs and time by means of automating the process
3) Avoids configuration mistakes by means of automating the process.
4) Reduces configuration verification time, by utilizing re-useable templates
5) Brings consistency to the survey and provisioning process, by embedding intelligence in the re-useable templates
6) Reduces admin knowledge requirement by embedding intelligence in the templates.
7) Provides a centralized database for surveyed data
8) Enforces best practices rules via standardized intelligent templates
9) Provides a single platform to provision different UC products
10) Provides a standardized deployment. A standardized deployment may address all the entitlement concerns in a large enterprise or for a service provider managing multiple clients.
11) Is a web based application and therefore eliminates the requirement of being at the location where deployment is taking place.

In this article we introduce the architecture of the Unified Communications Audit Tool (UCDT), designed to facilitate the deployment of enterprise’s Unified Communication applications including Cisco Unified Communications Manager, Cisco Unity Connection, Unity, Cisco Presence Server, Voice gateways, and the
underlying infrastructure. We elaborate on the internals and operations, and study this architecture in practice.

II. PROBLEM STATEMENT

There are two main challenges for deploying UC in large enterprises:

1) Configuration should be done for every site according to the network design
2) Multiple device types (such as gateways, network voicemails, presence etc), possibly from different vendors, should be configured

Automation is very desirable to facilitate the first point. Consider a financial enterprise with thousands of branches around the world. All the devices (gateways, switches, servers) should be configured based on the same design, and only few variables vary from site to site (such as usernames, IP Phone MAC IDs, directory numbers, etc). Therefore automating this task would save productivity. Customized tools can always be provided to automate the provisioning of a single device, but that means to satisfy the second point mentioned above users have to deploy multiple tools, just to perform the task of deploying UC. Therefore, the problem is how to have a single platform that can provision any device. UCDT provides a unique, one of the kind architecture that allows automation of the provisioning task for multiple device types, even from different vendors. It is a single platform that can configure any device. This reduces the cost of ownership significantly. The flexible architecture of this tool is introduced in this paper to demonstrate how it overcomes the challenge of automating deployment of multiple device types by a single tool.

III. SYSTEM ARCHITECTURE

The main components of the system are defined as: Administrator agent, Template Builder agent, Site Survey agent, Provisioning agent, Data Import agent, Remediation agent. Fig. 1 illustrates this architecture.

The Administration agent controls the access of the users to the tool based on their roles and permissions. It also allows defining parameters that can be used enterprise wide (only for a specific customer) or system wide (for all customers). System maintenance such as back up, applying software upgrades and finally uploading license is also done by this agent.

The “Template Builder” agent, allows users create templates, which are the framework of the provisioning requests. Templates are in fact provisioning request that have variables instead of the site specific data. Once user feeds the site specific data to the tool, these variables would be replaced by the data and a provisioning request would be formed. In other words, templates plus site data would provide provisioning requests. UCDT uses “Velocity” engine for the template writing language. Users can utilize all the features available by “Velocity” such as string manipulation or any kind or mathematical functions to write their templates. This would allow writing complex templates and embedding all the intelligence inside the template. This
would save users from the need to massage the data prior to uploading it to the tool. The templates are usually written by senior engineers and contain the fundamentals of the UC network design.

Once templates are created, tool would look into the templates and would find the variables used inside the templates and would generate datasheets which contain columns as variable names that need to be provided to the system. These variables are typically the name and details of the enterprise users, or MAC ID of the phones, or any other information that is required to be fed to the UC components for provisioning. The datasheets are filled out by the user and fed to the system through the “Import” agent. Since all the intelligence has gone to the templates, the task of populating datasheets (which is ongoing tasks for all sites) can be left to junior engineers or people with limited knowledge about the design of the network.

The “Survey” agent allows reviewing the uploaded site data. It provides a centralized database that can store all the site data. This is a good mean for storing the site data, for all UC products in a centralized database prior to doing provisioning.

Once the site data is provided to the tool it is used to replace the variables in the templates and form provisioning requests. Provisioning agent allows user review completed requests before sending them off to the UC server for provisioning.

UCDT currently supports provisioning of Cisco Unified Communications Manager, Cisco Presence Server, Unity, Cisco Unity Connection, Voice Gateways and Switches. However the tool provides ability to write plugins to provision any third party device as well.

The flexible template builder agent design, which is based on Velocity and allows generating datasheets according to the templates, plus the ability to provide plugins to provision new devices are two essential elements of UCDT that have made possible to do deployment of any device from a single platform.

IV. SYSTEM OPERATIONS

As Fig. 2 illustrates, there are two main phases for using UCDT, namely: the initial phase, when the templates are created, and the ongoing provisioning phase.

During the initial phase (Phase 1), users go through the UC network Low Level Design (LLD) and create the templates. The templates contain the skeleton of the provisioning requests. Any value that is consistent among sites/branches should be hardcoded and any value that might differ from site to site should be defined as a variable. The values for the variables would be provided.
to the system during site provisioning, when the datasheets are populated. Populating datasheets is a task prone to error and time consuming, so as the best practice, users are encouraged to hardcode as much information as possible inside the template. Also users are recommended to enforce best practices and avoid providing too much flexibility in the design. For example, it is recommended to assign product type based on job roles, for example, Cisco IP Phone 7961 to all regular employees and Cisco IP Phone 7975 to all site managers. This way we can define a rule in the template to specify the phone model based on job title, as opposed to having any kind of phone for any employee. This not only would simplify the provisioning and deployment process but also would greatly simplify the maintenance of the network.

Since templates are based on the LLD, if anything in the LLD should change, it would require template update, to reflect the same change. Therefore, it is recommended creating templates once the LLD is locked done. Experience has shown that changes in LLD are inevitable, but with this policy at least changes to the templates can be minimized.

Once the templates are created, they should be tested on a lab cluster. So datasheets are generated and then populated for the lab cluster. The provisioning task is then scheduled and launched, and the results are verified. Templates are revisited to fix any potential error or to optimize. Once templates are perfected, we can go to the next phase which is the ongoing provisioning phase.

In the beginning of the second phase (Phase 2), training is conducted to educate the people who would be responsible for populating datasheets and scheduling provisioning tasks on how to use the tool and the templates.

Upon the completion of the training, staff would be able to independently utilize the tool. They should gather the site survey data and populate the datasheets, then import to the tool and schedule and launch provisioning task.

Populating the datasheets is one of the most time consuming tasks in the whole provisioning process, as it is manual and needs human interference to make sure correct data is provided to the tool for provisioning. This is one of the factors that add delay to the provisioning process. In addition, there is a throttle speed that would be adjusted on the UCDT side, to control the speed of sending provisioning requests. This is to prevent the UC components from becoming slow under high volume of requests per second.

In order to increase the speed of the process with regard to these limitations it is suggested to populate datasheets for next sites/branches, while running provisioning tasks on previous sites/branches. Experience has shown that provisioning up to 50 sites/branches per week is practical.

System should be monitored during phase 2, but since the templates have already been verified, thorough verification is not required in this phase.

V. UNIFIED COMMUNICATION DEPLOYMENT TOOL INTERNALS

In this section we would visit three topics: security of the tool, currently supported platforms and the provisioning mechanism for each of them, and finally the template builder engine.
A. **UCDT Security**

UCDT is a web-based application and hence eliminates the need to be locally present at the site to conduct the provisioning. To ensure the security of the tool, it utilizes https protocol and therefore all the communication between browser and the tool is encrypted. UCDT also allows defining granular access rights to each agent of the tool for different users. The administrator of the system can grant access or deny each user from accessing specific modules. Typically all users have access to “Import” and “Provisioning” modules, whereas only a few users have access to “Template Builder” module. The reason is, templates are usually written by the senior engineers who have designed the network and other users are usually not supposed to modify or delete templates once they have been verified.

Furthermore, users can be denied from accessing/launching specific templates. The reason is some templates such as delete templates would have catastrophic impact if ran unintentionally. So the access to this type of templates would be typically only granted to administrators or senior users.

All passwords and critical information are also encrypted inside the tool. Therefore users do not have to worry about providing the credentials of their UC servers to the tool.

---

B. **UCDT Template Builder**

Template Builder is one of the major modules of UCDT. It provides a template editor and allows users write flexible templates and utilize programming language to conduct complex operations such as string manipulation or complex mathematical calculation. This would greatly simplify the provisioning process as the raw data can be fed to the tool as-is and no data manipulation has to be done over the data prior to feeding to the tool.

UCDT utilizes “Apache Velocity” engine in the template builder module. Therefore users can utilize all the commands available by Velocity to write templates. Once the templates are parsed and variables are replaced with data provided by datasheets, the provisioning requests are formed. Provisioning agent has a feature that allows users view completed provisioning requests, before sending them off to the UC server. This allows detecting parsing errors and other misses in the templates and greatly helps with the troubleshooting.

---

C. **UCDT Datasheets**

As it was mentioned, one of the most time consuming tasks for the whole provisioning process is populating datasheets, since it is a manual process and prone to errors. To facilitate the datasheet population UCDT provides concept of tags pick-lists. Tool allows users to define enterprise-wide, system-wide or even branch-wide pick-lists. Once the values are defined for a pick-list it appears as a drop down menu in the datasheet. This allows users to select a value from the menu instead of typing the value. UCDT tags are in fact variables that can be populated only once and be used for all the
templates. This would also simplify populating the datasheets.

The concepts of tags and picklists have greatly helped with facilitating the datasheet population task and have saved users from making mistakes and typos as well as saving time for typing the values.

VI. CASE STUDY

Figure 5 shows the effort hours needed to provision sites without using UCDT. Every site takes about 4 hours to be configured. So the total duration of deployment is 4 hours multiplied by number of sites.

Figure 4 shows the effort hours when utilizing UCDT. Initially around 3 weeks time is needed to create and fine tune the templates. Once templates are created, then provisioning sites would take approximately 15 minutes. So the total duration is number of sites multiplied by 15 minutes plus 3 weeks.

It can be concluded the initial time for writing templates is the most time consuming part. Once templates are created, the provisioning takes very little time. This fact makes UCDT a good candidate for deploying in large enterprises with so many sites/branches. If there are only a few branches for provisioning, UCDT might not have a great impact with respect to the time/cost saving.

Figure 3 shows the impact of utilizing UCDT on a deployment with large number of sites/branches. This is an actual case study from one of the customers.

VI. SUMMARY

This article explains the architecture, internals and process for using Unified Communications Deployment tool. The advantages of the tool are explained and the cost saving figures are presented.

References


5) Elizabeth Herrell, Lindsey Coit, Robert Whiteley, Forrester Research, Time To Revisit Your Unified Communications Priorities, Phase In Key UC Applications To Accelerate Business, August 2010