第 XXI 部

次世代 IT 運用管理システムの開発

第 21 部
次世代 IT 運用管理システムの開発

第 1 章　はじめに

本報告は、現在進行中である「次世代ネットワーク情報基盤運用管理システム」の開発に関し、その必要性や開発ポリシ、開発の現状について報告する。

第 2 章　次世代ネットワーク情報基盤運用管理システムの必要性

近年の情報基盤技術の普及、ネットワーク機器やコンピューティング環境の高度化・低価格化・多様化により、導入された多数の情報機器に対し、システム管理者の負担は増すばかりである。さらに、社会活動のネットワークや情報システムへの依存も高まりつつあるため、障害発生から対応までの迅速化が求められている。しかしながら、システム管理の人員や予算は十分でないばかりか、むしろ削減・圧縮が求められているため、システム管理・運用・監視を効率化するためには、自動化システムの導入が不可欠である。また、小規模な情報基盤であっても、運用管理の重要性は高まっており、誰もが手軽に利用できるシステム運用管理手法が望まれている。さらに、小規模環境で得られたスキルが大規模環境でも活用できることが望ましい。

運用管理の自動化や効率化・支援を目的として、すでに、様々な有償ソフトウェアシステムやオープンソースソフトウェアシステムが存在している。しかしながら、既存のソフトウェアシステムの多くは特定の環境や機器を対象とした閉じたシステムが多く、小規模な環境で利用するためは、不要な機能が多すぎる場合があり、異なるメーカー機器が混在している大規模なネットワークを管理するためには、単一のソフトウェアでは不十分で、複数のソフトウェアの利用が必要になる場合が多い。さらに、複数のソフトウェアを利用している場合、一台の機器の設定を変更すると、複数の管理ソフトウェアの設定を変更する必要がある、といった問題が生じている。また、特定のツールを利用するためには、様々なノウハウを学ぶ必要があるが、複数のシステムを利用するためのノウハウを学ぶことはコストが高く、運用管理の標準化のためにも問題が多い。さらに、情報システム管理者のスキルが、所属している組織が利用しているシステムに依存してしまい、汎用的なスキルとしての利用が難しくなる。結果として、特に大学等で構築されるようなキャンパスネットワークの運用や、WIDE のテストベッド、JGN2plus のイベントなどのネットワーク設置・運用の実情に対応し、統合的に、かつ安価に利用できるような適切な運用管理システムは存在していない。そこで、我々はそのアーキテクチャを検討するとともに実際の運用管理に適用可能なシステムの開発を行うこととした。

第 3 章　システム開発会議の発足と運営

我々は、上記のような問題意識に基づき、多様な立場からの問題意識や運用管理システムへの要求仕様をまとめるために、オープンな会議として「第 1 回 次世代ネットワーク情報基盤運用管理システム開発会議」を平成 21 年 8 月 21 日に開催した。参加メンバーは、ルータベンダー、システムインテグレータ、大学、などであった。また、会議の議事録や資料は Web サイト（`http://ngms.info/bbs`）上に公開している。開発会議は平成 21 年末までに WG を含め 14 回開催され、運用管理システムの仕様や方向性に関する議論を進めている。なお、開発するシステムの名称を「NGMS: Next Generation Management System」とした。また、開発の成果はすべてオープンソースとし、当初から SourceForge（`http://sourceforge.jp/projects/ngms/`）上での開発を進めることとした。また、開発会議や NGMS に興味をもっている方の情

報共有手段として GoogleGroups 上に ML を作成した（http://groups.google.co.jp/group/ngms）。

第 4 章　NGMS への要求仕様

開発会議の結果、NGMS が対象とする範囲を策定した。既存のソフトウェア資産を生かすことを前提とし、かつ多様な環境で動作することを期待すると、可能な限り広範な機能を持つ必要がある。当初 NGMS に対して検討した要求仕様を以下に挙げる。なお、これらの仕様策定をする際には、既存の運用管理ツールに関する調査も行った。

機能：

- モジュール化により、以下の機能を個別に選択利用可能に（依存関係管理）
- 運用監視（ping/ssh/ftp/http/https/pop/smtp/snmp など）
  既存ツールの活用・連携（Nagios/Hinemos/Hobbit/ZABBIX/OpenNMS など）
- 状況の可視化（稼働状況やフロー状況）（Cacti/MRTG 等）
- 資産管理（機器管理、ケーブル管理、IP アドレス管理）
- config 管理（各ベンダー telnet/SSH 対応、NetCONF 対応、バージョン・差分管理）
  既存ツールとの連携（rancid 等）
- VLAN 管理、可視化（各社機器への自動投入、end-end の指示だけで、中間自動策定）
- ACL 管理、自動投入（各社機器への自動投入、理由、履歴管理）
- 段階的なエスカレーション可能なトラブル等通知機能
- トラブルチケット管理
- トラブルシュート支援（ワークフロー、ナレッジベース、トラブル推定など）
- 認証・権限管理（Single Sign on との連携/RADIUS/LDAP 連携など）
- 他システムとの連携
  - WebAPI 等の公開による外部アプリ開発支援機能（JavaScript ライブラリなど）

インタフェース：

- Web ブラウザ経由で利用　・API 経由でのクライアントも可能
- 管理・設定用インタフェース
- 一般ユーザ用インタフェース
- ウィザードによる簡単導入機能

稼働環境：

- マルチプラットフォーム対応
- 分散/HA 稼働も可能とする

第 5 章　開発言語の選定（Scala）

NGMS では、マルチプラットフォームでの稼働を目指すため、開発言語や開発環境に関する検討を行った。当初は Java や Ruby の採用に関する検討を進めたが、Java は開発コストが高くなる傾向があり、Ruby では、言語としての仕様の安定性の問題（Version 1.8 系 or 1.9 系）や、動的言語ならではの実行時エラーの問題等が指摘され、大規模・長期間・分散開発に向いていない、という議論があった。結果として、近年注目されている新しい言語である Scala を採用することとした。Scala は JVM 上で動作するオブジェクト指向言語の関数型拡張であり、静的型チェックを有し、強力な抽象化機構を持つため、NGMS の開発に向いている、と判断した。なお、開発言語の選定のために、サンプルコードを作成し、評価を行った。これらの結果もすべて SourceForge 上に公開されている。

第 6 章　開発の現状（平成 21 年度の開発目標）

本プロジェクトが目標とする次世代の運用管理システムは、すべての機器やシステムを統合的管理するための機能を有するべき、と考えているが、開発期間や工数の関係上、今年度の開発では、NGMS の最小構成としての実装を進めることとし、平成 21 年の開発目標を以下のように定めた。

A）資産管理ファイル構造整理（DeviceTree）

B) 基盤データ構造管理系
C) ネットワーク機器情報の自動取得 (SNMP 経由)
D) ネットワーク監視用 config 作成
　1. Nagios、Cacti 等との連携
　2. 通知用プラグイン・通知フィルタ
E) インシデント管理
　1. インシデントのエスカレーション
F) インストーラ・アップデートシステム
以下、個別に解説する。

A）資産管理ファイル構造整理（DeviceTree）

NGMS では、すべての機器管理の基本に資産管理を置くこととした。すなわち、どのような機器がどこに設置され、どのような設定であるのか、という情報は、どのような管理システムであっても必要である。そこで、こういった広範な情報を記載するためのデータストアが必要と判断した。最近の運用管理システムでは、バックエンドにデータベースを持ち、機器の情報の管理をしているものもあるが、新しい情報を追加する度にデータベースのスキーマを変更する必要があったり、他のシステムとの連携が困難であったりする。NGMS では、これを DeviceTree と呼ぶ簡単な木構造で表現し、当初の実装はファイルシステム上で行うこととした。ファイルシステム上に、機器毎のディレクトリを作成し、ファイルに機器情報を記載する。ディレクトリ構造は、場所を基本として階層化することとした。また、ファイルシステムの履歴管理を行うために、バックエンドに Subversion を採用した。これにより、従来のデータストアでは管理が困難であったシステム改修の履歴管理も可能とする。また、これらのデータストアを簡便に扱うためのフロントエンドの開発を行うこととした。

例：DeviceTree のサンプル
root - Area - Location - Building - Floor - Room - DeviceName - log
- 名古屋 - 東山地区 - 豊田講堂 - メインホール - EPS - Cat2950G - logs
- 工学部 - IB 電子情報館 - 1F 東 EPS
– Alaxala-2430S-24T

例：Desc.txt のサンプル（機器情報ファイル）

```
DeviceName:swcc-37a03
DeviceType: L2SW
ModelName : Catalyst3750G-24TS-S
IPAddress :10.5.4.4
DatePurchased : 2010/01/04
DateAssetIDIssued: 2010/01/04
AssetID:  1001ICT0001
BudgetName: 情報連携統括本部 NICE4 導入
LocationName: エネルギーセンター　ノード室
SerialNo:
FirmwareNo:
```

B）基盤データ構造管理系

資産管理を基礎とした階層型データストア（現時点では、単なるファイルシステム）を効率的に扱うためのツール群の開発を行っている。Scala パッケージ info.ngms.nmtree は、データストアを扱うための基本的な API を提供し、info.ngms.nmshell は、各種ルータの CLI（Command Line Interface）と同様に、データストア上の各種操作を簡易化するためのシェルを提供する。また、データストアの履歴管理のための Subversion に対し、自動的な更新や同期を行う機能を提供する。

C）ネットワーク機器情報の自動取得（SNMP 経由）

機器の基礎情報（IP アドレス）等を指定するだけで、機器に関する情報を自動的に取得し、データストア上に記録する機構を導入する。初期段階では、SNMP を利用したデータ取得を行うが、将来的には、LLDP（Link Layer Discovery Protocol）を利用した、L2 レベルでの接続関係の解析や VLAN 情報の解析なども視野に入れている。

D）ネットワーク監視用 config 作成

データストア上の機器情報を利用して、ネットワーク機器を監視するためのツール（Nagios や Cacti）の config を自動生成する。これにより、機器情報を変更するだけで、様々なツールへの変更が不要になる。また、監視ソフトウェアから、インシデント管理系への通知を行うためのプラグインや、無駄な通知を行わないためのフィルタの実装も進める予定である。

E）インシデント管理

管理システムで把握した機器の異常や様々なトラブル等をインシデントとして、統一的に管理を行う

機能を開発する。インシデントに関しても、NMTree を用いた階層型データストアによって管理を行う。各インシデントの緊急度によって、対応を自動的にエスカレーションする機能を持つ。具体的には、管理者には、連絡先の登録とともに、階層的な優先度を割り当てる。機器等のインシデントが発生後、日付や時間帯等に応じ、担当管理者に通知を行う。一定期間内にインシデントの対応ステータスが変更されない場合、自動的に上位の階層の担当者に連絡を行う仕組みを「エスカレーション」と呼ぶ。これによって、重要なインシデントが担当者不在によって放置されるような事態をなくすことが期待できる。また、エスカレーション時には、連絡手段を電子メールの宛先を変えながら行うような仕組みを導入することを検討する。

F）インストーラ・アップデートシステム

どのように素晴らしいシステムでも、導入がむずかしければ、ユーザには受け入れられにくい。可能な限り簡単なインストール手法を実現することを目指す。また、システムが継続的にアップデートされた場合でも、システムを継続的に更新可能な仕組みを導入する。また、可能な限りマルチプラットフォームでの対応を目指すが、当初は Linux での管理を可能とする。

第 7 章　まとめ

NGMS は、現在開発中の次世代運用管理基盤ソフトウェアであり、長期的な視点で継続的に開発を進めることが求められる。また、多様な派生ツールの基盤となる仕組みの実現を目指しており、多くの開発者の参画を期待している。来年度以降は、コンソーシアムの設立を視野に入れた開発・普及活動を行いたいと考えている。

以上