

NDLによるネットワーク設計システムのためのエージェント記述 Agent Description format for Network Design System using NDL

奥山 雄紀[†] 武田 利浩[†] 平中 幸雄[†]
Yuki Okuyama Toshihiro Taketa Yukio Hiranaka

1. はじめに

近年ネットワークインフラの普及率は非常に高くなっており、ネットワーク規模が増大している。またネットワーク技術の多様化、それに加えネットワークのマルチベンダー化が進みネットワークの複雑さが問題になっている。それによりネットワークの設計、管理などが困難になっている。その困難さはネットワーク管理の効率を著しく低下させる原因になる。

我々はネットワーク記述言語 (Network Description Language、以下NDLと記す)によってネットワークの設計、管理などを統一的に扱うことを提案している[1]。これまでにネットワーク設計システムの開発してきているが、エージェント記述には非対応であった。本稿ではエージェントの記述形式の検討とシステムの実装について報告する。

2. NDLの概要

ネットワークの設計や設定、管理を行うためのXML形式で記述されたデジタルデータである。

2.1 記述ファイル

(1) ネットワーク設計の記述ファイル

ネットワークの設計はネットワーク設計者によって記述されるネットワークの要求仕様、いわゆる設計図である。ネットワークの構成、使用機器の情報などが記述される。

(2) ノードの定義の記述ファイル

ノード (ネットワーク機器) の基本的な設定項目をあらかじめ定義し、記述したもの。ノードのタイプごとに記述されている。ネットワーク設計ツールを起動するとき読み込まれる。

(3) ネットワーク機器の設定の記述ファイル

ネットワーク機器の設定を記述したもの。ただしネットワーク機器の設定はネットワークの設計を何らかの方法で機器に適用可能な形に変換する必要がある。

(4) ネットワークの状態 (スナップショット) の記述ファイル

ネットワーク管理時に、ネットワークの状態をNDLで記述し、ネットワークの設計と比較できるようにしたもの。

上記の4つのうちネットワークの設計とノードの定義が設計、ネットワーク機器の設定が設定、ネットワークの状態が管理にあたる。それぞれ同一のファイルではないので注意が必要である。

2.2 記述の対象

NDLにおけるネットワーク記述の対象には4つがあ

る。記述するのは次の通りである。

(1) ノード

ルーターやPCのような通信を行う物理的なネットワーク機器

(2) リンク

ノード間にある物理的な接続。LANケーブルなどがこれにあたる。

(3) エージェント

ノード上で動く通信を伴うソフトウェア

(4) フロー

エージェント間での接続、通信

2.3 NDLの構造

ネットワーク記述言語の基本的な構造を以下に示す。ネットワーク記述言語はXMLで書かれているのでまずタグの説明を図1で行う。

図示のnetworkタグはツリー構造の最上位の要素である。nodesタグはノードの情報を格納する。ネットワーク全体に一つしか存在しない。nodeタグは個々のノードの情報を格納する。属性としてidを持つ。linksタグはリンクの情報を格納する。このタグの子要素であるlinkタグには個々のリンク情報を格納する。linkタグはidを使いリンクされているノードを示す。agentsタグはエージェントについての情報を格納する。このタグは各nodeタグにそれぞれ一つしか存在しない。agentタグはエージェント個別の情報を格納する。flowsタグはフローの情報を格納する。Flowタグは個々のフロー情報を格納する。node、agentタグの子要素には具体的な情報を格納するタグが多数用意される。

本研究では、具体的な情報にそのネットワーク監視に必要なタグをいくつか追加し、ネットワーク管理と設計設定の連携を図った。

```
<network>
  <nodes>
    <node id="*****">
      <Agents>
        <Agent>
        </Agent>
      </Agents>
    </node>
  </nodes>
  <links/>
  <flows>
    <flow>
    </flow>
  </flows>
</network>
```

図1 NDLの基本的な構成を行うタグ一覧

[†] 山形大学 Yamagata University

```

<Agents>
  <Agent>
    <name>
    </name>
    <work>
    </work>
    <time>
    </time>
    <trouble_value>
    </trouble_value>
    </Agent>
  </Agents>

```

図2 エージェントタグの一例

2.4 エージェント記述

本研究で追加したエージェントの記述について述べる。本研究では議論されていたエージェントの記述を元に管理側の情報を加えて実装した。

実装したエージェントのタグの一例を図2に示す。

図示の `name` タグではエージェントの名前を記述する。`Work` タグはエージェントに関する情報を格納したタグを保持している。`time` タグはエージェントの稼動時間帯を記述している。`trouble_value` タグはエージェントの重要度を記述する。このタグの内容は `must/should/may` の三段階で表す。`flow` タグは図1のタグと同様の内容である。`option` タグは機器やエージェント固有の追加情報を記述する。`hreshold` タグはPCなどネットワーク機器のCPU使用率、ディスク使用率などの閾値を記す。`work・threshold` タグは管理側に必要な情報である。

3. システムの概要, 実装

3.1 概要

NDLは前述のとおりXML形式である。つまりテキストエディタで書くことができる。しかし様々なタグがあり拡大する現在のネットワークを全て書いていくのは現実的ではない。そのためそれを効率化する必要がある。

先行研究によって設計ツールの開発はなされているがどちらもエージェントの記述までは実装にいたっていない。よって今回の研究ではエージェントの記述に対応した設計ツールを作成する。

エージェント記述対応に関しては以下のやり方で検討する。

(1) GUI上で他のノードを配置するようにエージェントを配置できるようにし、それぞれのノードに対応付けできるようにする。

(2) それとは別にエージェント用のフレームを作り、効率的に設定できるようにする。

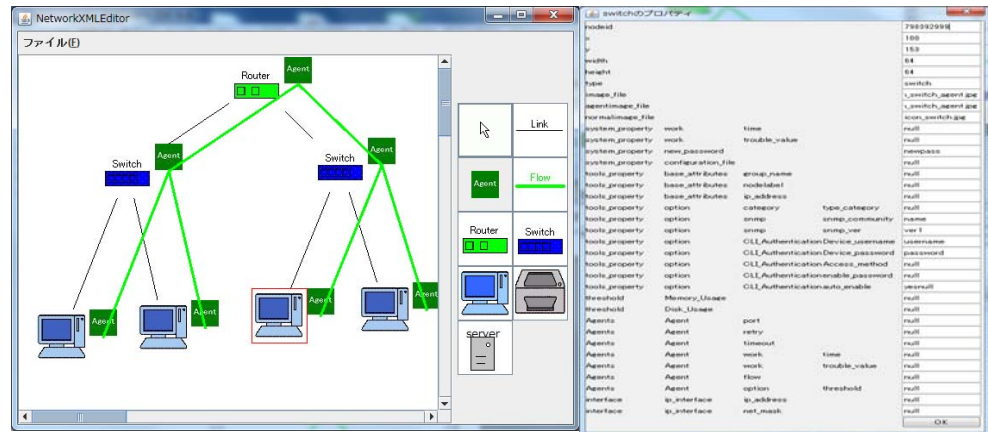


図3 設計システム画面

図4 エージェント情報などの記述画面

3.2 実装

本プログラムでは開発言語をJava[2]とし開発環境としてNetBeans IDE6.8を用いた。以下の機能を実装した。

(1) エージェントのネットワーク設計エリア上への記述対応。

(2) エージェントの記述対応に伴い、NDLの再定義・管理に必要な情報を導入した。

図3, 4に設計システムを用いての設計画面を示す。図3は設計システムの画面、図4はエージェントやノードの方法を記述する画面である。

設計図をネットワーク設計エリアに記述し、「XMLファイルのエクスポート」を選択すると設計ファイルが出力される。

4. 考察

エージェントを記述できるようにするという目標は達成した。しかし設計エリアからの情報の入力方法やNDLで記述すべき情報など検討すべき問題がある。またNDLの定義ファイルにいくつかのネットワーク監視に必要なタグを追加したのでそれを考慮したネットワーク設計システムの開発が必要である。

5. おわりに

本研究ではNDLにエージェント記述を追加し、ネットワーク設計システムをエージェントの記述に対応させた。また、NDLの設定ファイルの定義に管理側の情報を追加しネットワークとの連携をとりやすくした。

本研究の課題としてはグループ操作機能の実装があげられる。ネットワーク規模が増大している近年ではノード数が非常に多くなるのも珍しくない。そうになると設計システムを使用しても設定の入力は非常に手間となる。そのためNDLでグループ機能の定義が必要であり、それを設計システムに実装する必要がある。

参考文献

- [1] 金子佳正, 武田利浩, 平中幸雄, "ネットワーク記述言語を用いたネットワーク設計支援に関する一考察", FIT2010(第9回情報科学技術フォーラム), 第4分冊167-168, L002 (2010)
- [2] jsvsTM Platform Standard Edition 5.0API仕様
<http://java.sun.com/j2se/1.5.0/ja/docs/ja/api/index.html>