

RDBを用いた通信ノードパラメータの整合性検証

4Q-4

萬木優子 金枝上敦史 清水桂一* 小山啓一**

三菱電機(株)情報技術総合研究所 *同 通信システム開発センター **同 通信システム統括事業部

1. はじめに

ATMやフレームリレーなどのネットワーク構築では、ネットワーク上に配置される通信ノード(交換機等)に対し個々にパラメータ設定を行う必要がある。しかしノード数が増えるに従い、このネットワーク構築作業は、作業量が膨大になり、手作業でノード間の整合性を確保することは困難である。そこで今回この作業を支援するツールを開発した。

本ツールは、PC上でネットワーク内の全てのノードについてパラメータデータを作成・管理し、整合性を検証した上で、入力形式を自動生成する。本ツールはRDB上に構築されており、RDBのデータモデルを用いて整合性ルールを記述している。本稿では、ツール開発に際しRDBを用いて効果的に整合性検証機能を実装する方式について検討したので報告する。

2. パラメータ設定における課題

ネットワーク構築時に設定するパラメータは、そのままではバラバラな複数の通信ノードに対し隣接ノードやネットワークを認識させ、ノード内部の交換機能を定義することにより、各ノードをネットワーク内の1ノードとして役割付けるものである。この作業で設定すべきパラメータ数は1ノード当たり、少ない場合でも数十以上あり、規定値をあまり用いない場合には千以上に上ることもある。ネットワーク構築時には、各ノードに設定するパラメータをテキストで作成し、それを通信ノードに入力しエラーチェック後、再びコンピュータ上で修正するということを繰り返す。また各ノードに設定したパラメータが形式的に正しくても、複数のノード間に設定したパラメータ間に矛盾があればネットワークとしては機能しない。従って、パラメータの形式的、意味的な整合性確保が課題となっている^[1]。

3. ツール概説

今回ネットワーク構築を支援するため以下の機能を持つツールを開発した。

- データベース機能：作成するパラメータデータを格納する
- 通信ノード入力形式生成機能：データベース内のデータから通信ノードへの入力形式データを生成する。
- 検証機能：データベース内のデータが整合性を満たしているかどうかを検証する。
- 入力機能：データベースにデータを入力するためのGUI機能。入力機能の構造により、基本的な整合性を確保することが可能である。

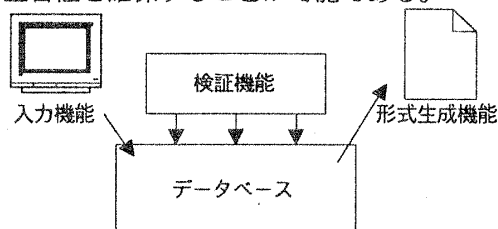


図1: 支援ツールの機能構成

4. 整合性検証方式

本ツールによる支援の主目的はパラメータデータに対する整合性の検証である。検証ルールは、今回開発したツールで百余りとなった。ここでは検証ルールの種類と実装方式について述べる。

本ツールは、RDBがベースとなっており、RDBのデータモデル上に検証ルールを構築したので、まずこのデータモデルについて述べる。本ツールのデータモデル(データベース構造の定義)は通信ノードパラメータをテーブルとして持つ。例えば、物理的なカードやポート、仮想バスや仮想チャネルをそれぞれテーブルとして定義する。そして、テーブル間に従属関係がある場合、それらを関連として定義する。またネットワーク内の全ノードに共通なパラメータを管理するためにネットワークを表すテーブルを設ける。従って、本ツールのデータモデルは図2のような構造を持つ。

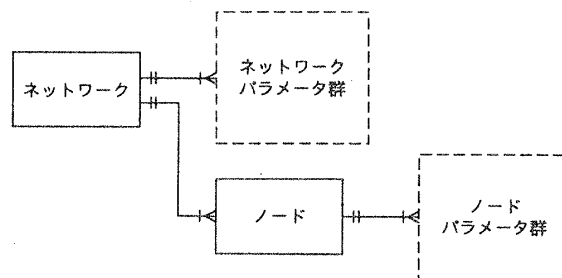


図2: データモデル構造

Verifying Parameters of ATM Switch by Relational Database
 Yuko YURUGI, Atsushi KANAEGAMI, Keiichi SHIMIZU*,
 Keiichi KOYAMA**,
 INFORMATION TECHNOLOGY R & D CENTER,
 *COMMUNICATION SYSTEMS R & D CENTER,
 **COMMUNICATION SYSTEMS BUSINESS DIV.
 MITSUBISHI ELECTRIC CORPORATION

検証ルールはこのデータモデルに対して定義する。検証ルールには意味的に、以下のようなレベルが存在する。

- ノード内

1つのノードに設定するパラメータ同士の整合性ルールである。ノード内に存在する物理的構造やサービスに関するルールである。

- 隣接ノード間

隣接ノード間の接続関係に関するルールである。隣接するノードは互いにペアとなる物理ポートや仮想パスなどを保有しなければならない。

- ネットワーク

ネットワークが設計通りに機能するためのルールである。ある2つの通信拠点間に要求された回線を確保するには、それを経由する通信ノード群に物理的・論理的パスを定義し、必要な容量を満たすようなパラメータ^②を設定しなければならない。

しかし、RDB上での実装の観点で見た場合、ネットワークレベルの検証ルールも、同じテーブル内のインスタンス同士の関係として定義することができる。今回、検証ルールを検証ルールの対象パラメータのデータ構造により分類し、実装した。以下、この分類に従って実装方式を説明する。

(1) 単独パラメータのルール

1つのパラメータに値の許容値や既定値がある場合のルールである。このようなルールはデータモデルの属性定義として定義でき、定義された場合RDBMSにより、登録時チェックが行われる。

(2) 単独インスタンスのルール

単独インスタンス即ち同時に設定される属性グループ内のルールである。この種のルールはGUIにおいて、入力時に自動的に値を修正したり、インスタンス登録時に検証を実行することができる。

(3) テーブル間の構造に関するルール

テーブル間に従属関係がある場合の整合性ルールである。物理的なカードとポートの関係や、仮想パスと仮想チャンネルのように、ノードの構成に関するルールが相当する。このような従属関係に対しては、テーブル間に参照整合性制約を持つ関連を定義する。参照整合性制約とは、従となるテーブルのインスタンスが、他の主となるテーブルのインスタンスに依存する時、主側のテーブルのインスタンス削除時に従側のテーブルのインスタンスを削除したり、主側の削除自体を禁じたりすることにより整合性を保つ機能で、RDBが一般に持つ機能である。従って、データモデルでテーブル間に参照整合性制約を定義することにより、テーブル間の構造に関する整合性が確保される。

(4) 意味上の関係に関するルール

(3)の従属関係のように、固定的な関連がないテーブル間、インスタンス間でも、ある概念によってそれらを結び付ける関係が存在することがある。例

えば、物理的コンポーネントとサービスの関係や、隣接ノード間の関係など、ノードの役割を決定するようなパラメータに多い。しかし従属関係のないテーブル同士の関係をRDBの関連によって定義すると、インスタンス登録時にこの関係が完全な形であることが要求される。従って、利用上プログラミングによる弱い連結が向いているが、このような関係に対しルールが複数存在する場合も多いので、この関係を表す何らかの定義が欲しいところである。そこで、この関係をRDBのビュー(仮想テーブル)として定義し、このビューに対しルール定義を行なうこととした。この方法は検証ルール作成の効率上と利用上の双方で都合が良い。

(5) インスタンスの集合に関するルール

ある条件を満たすインスタンス群に対し集計的な計算を行い、この計算結果に対して適用されるルールである。例えばある物理ポートに定義できる仮想チャンネルの容量合計やネットワーク上のトラフィックに関する計算など、設計ノウハウに関するルールがこれに相当する。この種のルールも(4)と同様データモデル上にビューとして定義可能である。しかし設計ノウハウには複数の要素が含まれる場合も多く、ビューの定義が複雑になりやすいため、このような場合にはプログラミング言語により記述する方が柔軟で、間違いが起きにくかった。

5. まとめ

検証ルールを対象データの構造により分類することで、検証機能を効果的に実装することができた。

通信ノードの構成を定義するようなパラメータの検証には、データモデル定義による入力値チェックや参照整合性制約の機能が便利である。

また、関係が複雑で最初から全てのルールに適合するようにデータを作成することが困難なパラメータが存在し、これらのパラメータに関しては、入力時にはある程度矛盾を許し、検証を最後に行った方が良い場合があることがわかった。このようなパラメータの関係はデータモデルに定義する代わりにビューとして定義することにより、効率良く実装できた。

しかし、本ツールの支援対象は1つの機種に限られており、機種毎にツールを別々に開発する必要がある。検証ルールの中には実装に工数がかかり、どの機種でも通じる普遍的なルールが存在する。今後、このようなルールを再利用できるよう、データモデルを抽象化するための構造を検討していく必要がある。

参考文献

- [1] 松田博文, 岩崎英一他, ネットワーク構築支援システム, MSY技報 Vol.2, 1999
- [2] マーク・ミラー, 広帯域ネットワークシステム, プレンティスホール, 1997