拡張ネットモデルによるNFSサーバプロセスの設計と動作の検証

The design and veri¯cation for the Network ¯le system based on an Extended Petri Net

C-22

山口　真之介＊　　和崎　克己＊　　師玉　康成＊
Shin'nosuke Yamaguchi　Katsumi Wasaki　Yasunari Shidama

1　はじめに

本研究は，拡張ペトリネットによるネットワークファイルシステムのモデル化を行い，その記述能力について評価を行った．並列システムを設計するとき，システムの安全性や動作を解析するために，ペトリネットによるモデル化を行うことが多い．これはペトリネットのモデル記述能力が並列システムの記述に適していることと，ペトリネットの解析が容易であることによる[1][2]．しかし，従来のプレース・トランジションペトリネットの場合，システムの動作条件が複雑になるにつれて，極端にプレース，トランジションの数が拡大し，モデル化が現実的でなくなる．このため，従来のペトリネットを拡張したLCペトリネットを既に提案している[3]

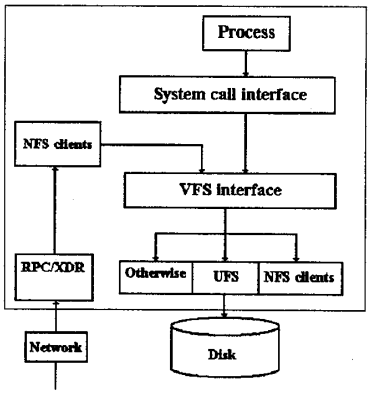

図 1: NFSサーバの構成図

2　NFSプロシージャの設計例

NFSはクライアント・サーバ型のネットワークファイルシステムである[4]．図1にサーバの構成を示す．ユーザーはNFSクライアントプログラムによって，サーバ上のファイルをマウントし，利用することが出来る．NFSではサーバ上のディレクトリやファイルを，ファイルハンドルと言う64bitの整数で識別している．NFSクライアントプログラムは，21のプロシージャによって，サーバ上のファイル操作を行う．本論文では，そのプロシージャ毎にLCPNによるモデル化を行い，その動作について再検証を行った[5]．例としてREADプロシージャの設計について取り上げる．

プロセスのモデル化には従来のペトリネットではトークンの持つ情報，トランジションの発火条件などの記述能力が要求される．しかし従来のペトリネットではその記述能力が低いので，トークンに情報を付加し，発火条件を論理式で表し，更にプレースの出力も演算結果として表現する色つき論理ペトリネットを採用した[3] まず本研究で定義した，仕様に基づいたトークンのデータ構造を図2に示す．このマーク構造には，READプロセスに必要なファイルハンドル (¯le)．読み出しのポインタ (o®set)，読み出すデータ量 (count) を定義し，サーバからのマークには読み出し後のファイル情報 (¯le_attributes)，実際に読み出したデータ量 (count)，EOF フラグ (eof)，読み出したデータ (data) を定義している．エラーの場合は読み出したデータに関する情報は存在しない．

次にREADプロシージャのネットモデルを図3に示す．p_1 がユーザからの入力であり，p_{33} が，サーバのレスポンスになる．READプロシージャの実行中に他のクライアントから別のREAD要求を受け，新しいトークンが入力された場合，先に入力された要求が上書きによって損失する恐れがある．そこで今回の設計においては，排他処理用のリソースとしてプレース Pr_1 と Pr_2 を設ける必要があった．また，プロセスのモデル化において，エラーハンドリング高速化を試みるために並列モデルの設計を試みている．それぞれのエラーに対応したトランジションが与えられた情報をそれぞれが並列にチェックを行うことで，逐次にチェックするよりも高速に処理される．図では，t_2〜t_4，また，t_9，t_{10} が，それに

＊信州大学工学部　情報工学科

当たる．この際，NFSの仕様では，クライアントに返すエラーはただ一つであり，並列にエラーチェックを行う場合，異なるエラーが複数同時に現れると言う問題が生じた．そこで，それぞれのエラーに優先度を与え，エラーチェック後に最も優先度の高いエラーのみをクライアントに返すことで解決を図った．

No.	File-handle	Offset	Count1

クライアントから送られるマーク情報

No.	Status	File-attributes	Eof	Count2	Data

サーバが返すマーク情報(NFS_OK)

No.	Status	File-attributes

サーバが返すマーク情報(NFS_ERROR)

図 2: READ プロシージャにおけるマーク情報の定義

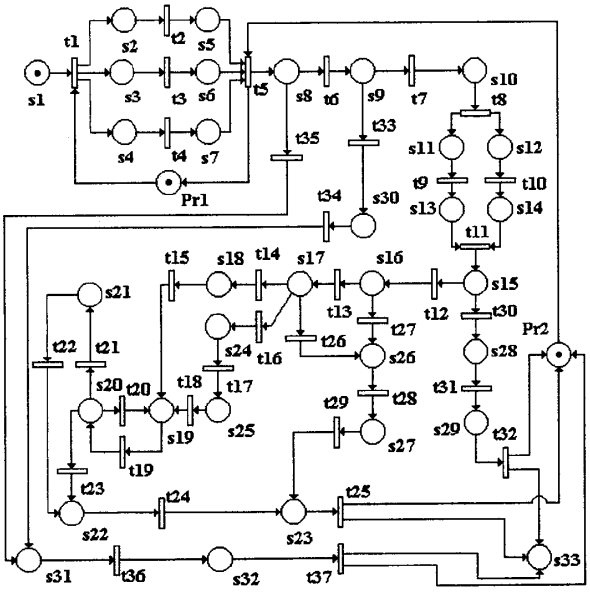

図 3: READ プロシージャのネット構造

3 ネットモデルの動作と検証

本論文では Renew(The Reference Net Workshop)[6] を利用してこのネットモデルを設計，その動作について検証した．Renew は Java 上で動作するペトリネット設計ツールの一つであり，ネット内で Java のクラスの参照や，設計されたネットモデルを XML ファイルとして出力することが可能である．Renew 上でマークの動作のシミュレーション行った結果，READ プロシージャのネットモデルは，どのルートにおいても，正常にマークが流れ，デッドロック等の構造上の問題点は無いことが確認された．

さらに Renew の機能である Java のクラス呼び出しを利用して，READ プロシージャの動作の再現を行った．ここでは，マークに図 2 で定義した情報を実際に与え，それぞれのトランジションで，動作に応じたクラスを呼び出し，設計したプロシージャの動作を検証する．ただし，Java 上でのファイル情報には，本来の NFS サーバのファイル情報に含まれないものが存在しており，それについては，エラーや読み出し処理に必要なファイル情報のみをクラスによって取得，処理を行っている．このシミュレーションでは，ファイルの存在，属性に基づくエラーチェック，要求された容量に応じたファイルの読み出しが確認された．なお，今回のネットモデルには，51 個のトランジション，58 個のプレースを必要とした．設計したネットモデルは XML ファイルに変換し，そのデータ構造を取り込み，構造に従って動作する Java によるエンジンプログラムによって実装を行っていくが，この規模であれば XML ファイルを利用した，HDL 等の他の言語への変換も可能であると考える．XML ファイルの入出力には Relaxer を採用し，Java プログラムによる実装への効率化を図る．

4 まとめ

拡張ペトリネットを用いて，ネットワークファイルシステムの READ プロセスに関するモデル化を行い，動作とエラーハンドリングについての検証を行った．今後の課題はネット構造の XML ファイルの定義と，そのファイルを読み出して動作する，エンジンプログラムの作成が必要となる．

参考文献

[1] T.Murata : \Petri Nets: Properties, Analysis and Applications ", Proc. IEEE, Vol.77, No.4, pp.541{580, 1989.

[2] J.L.Peterson : \Petri Net theory and the Modelling of Systems" , Prentice-Hall, Inc., 1981.

[3] K.Wasaki, Y.Fuwa, M.Eguchi and Y. Nakamura : \Logical Coloured Petri Net Expanded to be Suitable Making the Control System Model.",Fourth International Conference on Control, Automation, Robotics and Vision (ICARCV'96) Proc. TA-2-2,pp.708-713,December,1996.

[4] RFC1813: \NFS Version 3 Protocol Speci¯cation",June 1995

[5] 山口 真之介，和崎 克己，師玉 康成: \拡張ペトリネットを用いた NFS プロセスのモデル化と実装" 信学技報,CST2002-2,May,2002.

[6] Renew (The Reference Net Workshop) , http://www.renew.de/