ショートノート

分散処理システムを例とした不変集合解析による システムの系列的な故障診断

勝間田 仁† 菅澤喜男††

近年，ネットワーク技術の飛躍的な発達で，非同期・並列的な処理をする分散処理システムの構築と利用が様々な分野でなされてきている．分散処理システムを代表的な例とした非同期・並列的な処理をするシステムの故障要因を系列的に把握することは，一般的に複雑であると共に，時間と経費をも要する作業となる．本論文では，非同期・並列的な処理をするシステムの代表的な例として基本的な分散処理形態を有するシステムを取り上げペトリネットを用いてモデル化する．ペトリネットでモデル化されたシステムの不変集合を求めることで，システムの故障診断を系列的に行う方法を示し，効率の良い故障診断あるいは保守のあり方について提案する．

Serial Failure Diagnosis of the System by Invariant Analysis as an Example of Distributed Processing System

Masashi Katsumata† and Yoshio Sugasawa††

Recently, network technologies have rapidly developed and distributed processing systems which are of asynchronous and parallel characteristics are very useful in various fields. However, it is very difficult to figure out the factors of failure of such systems. In this paper, we make a Petri net model of a simple distributed processing system consisting of a main system and a subsystem, and show how to diagnose serial failure of the system by analyzing invariant sets of the model.

1. はじめに

分散処理システム[1]が社会に広く普及し重要な使命を持ちつつある現状では，その障害が社会に与える影響は極めて大きい．また，分散処理システムを構築した場合，その規模が非常に大きくなるのが一般的であり，システム全体が常に無故障であると仮定することはほとんど不可能である．そこで，システムの一部分に故障が発生した場合，故障修理等の対策を効果的に実行する必要がある．そのためには，システム全体の系統的な保守性（Serviceability）がシステムの信頼性を高く維持するために重要な問題となる．

従来より，システムの系統的あるいは体系化された故障要因を示す方法としては，故障木（Fault Tree）[2]が用いられてきた．故障木の目的は，システム（あるいはサブシステム）故障となる条件をモデル化することである．これは，現在考えているシステムが故障状態（動作不能状態）となる故障事象と正常事象の列挙を必要とする．従って，故障木の構成方法は，システム解析者の持っているシステムの知識と理解度に深く関係し，解析者の経験に頼る面が多々ある．また，故障木は比較的簡単なシステムの故障要因を系統的に示すことができるが，分散処理システムなどを対象にした場合，システムを構成する要素あるいはユニットが相互に関連して処理あるいは動作するために，構成要素の故障に相互依存性が生じ，システムの構成要素面からだけでは故障要因を把握することは困難である．

そこで，本論文では，非同期・並列的な処理あるいは動作をする分散処理システムを取り上げて，ペトリネット（Petri Net：以後 PN と記す）[3],[4]を用いてモ

† 北海道大学工学部情報工学科
Faculty of Engineering, Hokkaido University
†† 日本大学生産工学部数理工学科
Faculty of Industrial Technology, Nihon University

デル化する．分散処理システムを PN でモデル化することで，システム構成要素における故障の相互依存性を記述することが可能になる．次に，モデル化されたシステムの不変集合を求め，システムの故障要因をシステムの構成要素面からだけではなく，システムの働き，つまり，機能面をも考慮した故障要因を系統的に把握する方法を示す．

2. 分散処理システムの PN モデル

ここでは，PN の簡単な概略を述べ，分散処理システムの PN モデルを示す．

本論文で取り上げた非同期・並列的な処理をする分散処理システムなどをモデル化するのに優れたモデル化技法として知られている PN は，システムにおける条件と対応づけられる場所 (place) の集合，システム中の事象に対応づけられる遷移 (transition) の集合およびそれらの条件と事象の関係を表す有向線分 (directed arc) とにより構成され，場所に刻印つまりマーキング (marking) を与えることにより，そのシステムの状態を表す．

PN における N は，次の四つの組で定義される二部有向グラフであり

$$N=\langle P, T, A, M_0\rangle \qquad (1)$$

で表される．ただし，

$$P=\{p_i \mid 1\leqq i\leqq |E|\} \qquad (2)$$

$$T=\{t_j \mid 1\leqq j\leqq |T|\} \qquad (3)$$

であり，P は有限個の場所 p_i の集合で○印で表す．T は有限個の遷移 t_j の集合で｜印で表される．A は有限個の有向線分の集合で，場所 p_i から遷移 t_j への有向線分の部分集合と遷移 t_j から場所 p_i への有向線分で構成される．M_0 は初期刻印でシステムの初期状態の設定により決定される．集合 A に属する有向線分で，場所 p_i から遷移 t_j へ向かう有向線分があるとき，場所 p_i を遷移 t_j の入力場所と呼び $I(t_j)$ で表す．次に，集合 A に属する有向線分で遷移 t_j から場所 p_i へ向かう有向線分があるとき場所 p_i を遷移 t_j の出力場所と呼び $O(t_j)$ で表す．従って，有向線分を→で示し，入力場所 $I(t_j)$ にある●印で示される標号 (token) を遷移の発火 (fire) により出力場所 $O(t_j)$ に置く．また，遷移の発火により刻印がなされてシステムの状態を捕らえる．

本論文では，オンライン処理とローカル処理を中心とするメインシステムとサブシステムの代替が可能な基本的な分散処理形態を有するシステムを取り上げる．メインシステムは主にオンライン処理をし，サブシステムは主にローカル処理をするが，ある一定の条件を満たすとメインシステムとサブシステムが交替しシステム全体としての機能を果たす．**図1**に分散処理システムの機能概略図を示す．次に，分散処理システムの故障発生時の処理について述べ，**図2**にその処理手順を示す．サブシステムは，ローカル処理中に通信障害等によりシステム故障となるが，メインシステムの機能によりその機能は回復する．しかし，メインシステムとサブシステムが交替中に，通信障害等が起こると双方の機能とも停止し，トラップ状態（あるいは

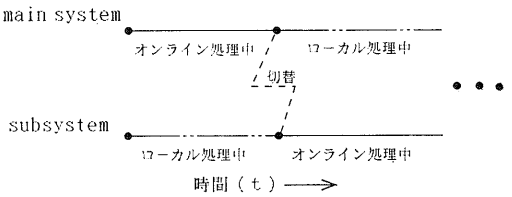


図 1 分散処理システムの機能概略図
Fig. 1 Configuration of a distributed processing system.

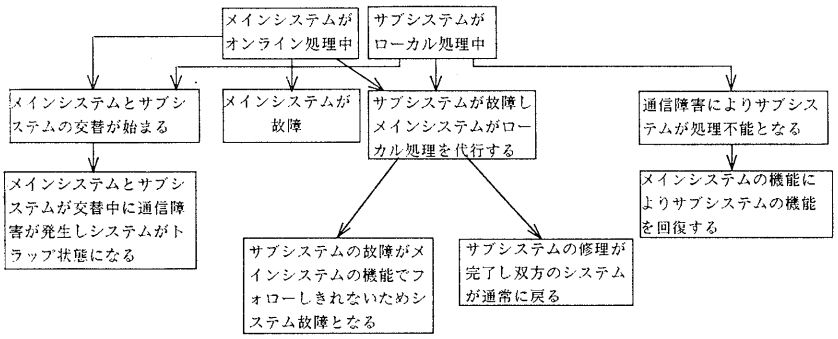


図 2 分散処理システムの障害処理手順
Fig. 2 The procedure of failure analysis of a distributed processing system.

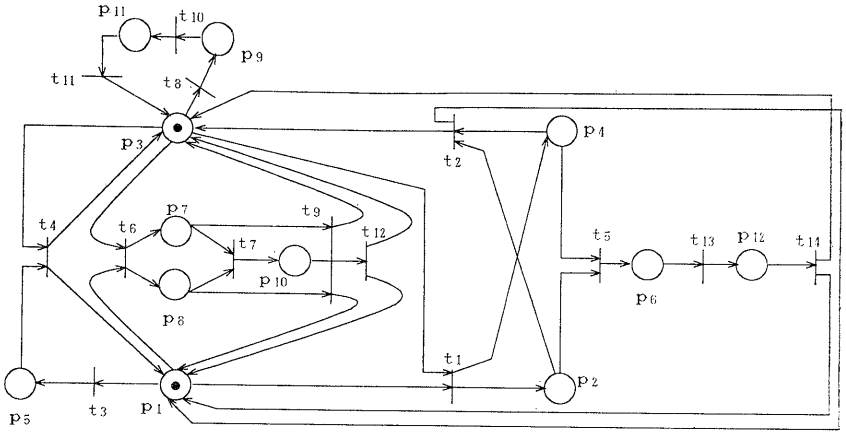


図 3 分散処理システムの PN モデル
Fig. 3 PN model of a distributed processing system.

システム故障）となる．また，サブシステムが独立で故障した場合は修理を受け再稼働する．また，システムが故障状態，トラップ状態になった場合には修復機能によりシステムは回復する．この様なシステム構成要素の故障に相互依存性がある挙動をする修復可能な分散処理システムの挙動を PN でモデル化したものが**図 3** である．図 3 における各場所 ($p_i=1,2,\cdots,12$) と各遷移 ($t_j=1,2,\cdots,14$) の意味は次のとおりである．

p_1 ：サブシステムがローカル処理中
p_2 ：サブシステムがメインシステムと交替中
p_3 ：メインシステムがオンライン処理中
p_4 ：メインシステムがサブシステムと交替中
p_5 ：サブシステムが故障しシステム故障状態
p_6 ：メインシステムとサブシステムがトラップ状態
p_7 ：サブシステム修理中
p_8 ：メインシステムがローカル処理中
p_9 ：メインシステムが故障しシステム故障状態
p_{10} ：メインシステムとサブシステムが故障状態
p_{11} ：メインシステム修理中
p_{12} ：メインシステムとサブシステムのトラップ修理中
t_1 ：メインシステムがオンライン処理からローカル処理へ，サブシステムがローカル処理からオンライン処理へそれぞれ切り替えられ，メインシステムとサブシステムの交替が始まる
t_2 ：メインシステムとサブシステムの交替が終了し，メインシステムがオンライン処理，サブシステムがローカル処理にそれぞれ復帰する
t_3 ：通信障害などでサブシステムがシステム故障となる
t_4 ：メインシステムによりサブシステムの機能を回復する
t_5 ：サブシステムとメインシステムとが交替中に通信障害などが発生し，システムがトラップとなる
t_6 ：サブシステムが故障となり，メインシステムがローカル処理を代行する
t_7 ：サブシステムの故障がメインシステムの機能でフォローしきれないためシステムが故障となる
t_8 ：メインシステムが故障しシステムがシステム故障となる
t_9 ：サブシステムの修理が完了し，双方のシステムが通常の処理に復帰する
t_{10} ：メインシステムの故障修理を開始する
t_{11} ：メインシステムの故障修理が終了する
t_{12} ：サブシステムの修理が完了し，システム再始動となる
t_{13} ：メインシステムとサブシステムのトラップ修理を開始する
t_{14} ：メインシステムとサブシステムのトラップ修理が終了する

3. PN モデルの不変集合解析

PN の不変集合は，PN の構造的な性質を知るための解析手法で，P-不変集合と T-不変集合がある．P-不変集合は，PN モデルの各マーキングにおいて標号の総数が一定である場所の部分集合で，モデルの安全性を確認するために用いられる．T-不変集合は，あるマーキングから出発して発火可能な遷移を何回か発火させると，元のマーキングに戻ってくる発火系列の部分集合で，モデルのあるマーキングの再生成（regen-

eration) を可能にすることを示す.

システムの故障診断を効果的に行うには，システムの構成要素面からだけでなく，システムの働き，つまり機能面をも考慮した故障要因を系統的に把握する必要がある．PN の P-不変集合は，システムにおける状態の安全性を確認できるが，システムの機能の流れを捕らえることは不十分である．そこで，PN モデルの T-不変集合は，システム中の事象を対応づけた遷移の集合であるので，本論文で，取り上げた修復可能な分散処理システムにおける故障修復系列と一致し，システムの働き，つまり機能面をも含めた故障要因を系統約に捕らえることができる．

図3で示した PN モデルの T-不変集合は，

$$N^{T}\cdot Y=0 \qquad (4)$$

を満足する Y である．ただし，T-不変集合において T-不変集合となる真部分集合を持たないものとする．

式(4)において，N は PN の接続行列であり，

$$N=(N(t_j, p_i)) \qquad (5)$$

として定義される．ここで，$t\in {}_jT$ で $p_i\in P$ であり，

$$N(t_j, p_i)=\begin{cases} 1 & p_i\in O(t_j),\ p_i\notin I(t_j) \text{ のとき} \\ -1 & p_i\in I(t_j),\ p_i\notin O(t_j) \text{ のとき} \\ 0 & \text{それ以外} \end{cases}$$

である．ここで，図3で示した PN モデルの接続行列 N を式(6)に示しておく．

$$N=\begin{pmatrix} -1 & 1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix} \qquad (6)$$

次に，図3で示した PN モデルにおける T-不変集合を求めると，

$$Y1=[000000011011000]^{T} \qquad (7)$$

$$Y2=[000000100100000]^{T} \qquad (8)$$

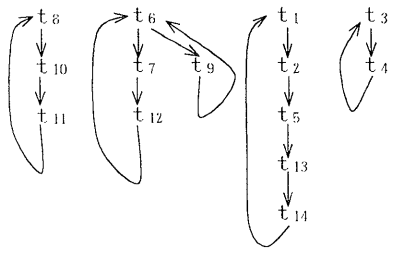


図 4 図3の故障系列
a)：メインシステムの故障系列
b)：サブシステムとメインシステムの故障系列
c)：システムのトラップ故障系列
d)：サブシステムの故障系列

Fig. 4 The failure sequence for Fig. 3.
a) failure sequence of main system
b) failure sequences of subsystem and main system
c) trap failure sequence of system
d) failure sequence of subsystem

$$Y3=[000001100000100]^{T} \qquad (9)$$

$$Y4=[110010000000011]^{T} \qquad (10)$$

$$Y5=[001100000000000]^{T} \qquad (11)$$

の5組が求まる．

4. システムの系列的な故障診断への応用

求められたT-不変集合をシステムの故障系列として捕らえると**図4**で示す4つの故障に関する系列が得られる．

1) 図4のa）は，メインシステムの故障系列を示す．式（7）で示したT-不変集合 $Y1$ の t_8, t_{10}, t_{11} の系列であり，メインシステムが故障しシステム故障となり，メインシステムの故障修理をするメインシステムの故障対策系列である．つまり，メインシステムの故障修理は，t_8, t_{10}, t_{11} の順で系列的に対応すれば良いことになる．

2) 図4のb）は，サブシステムとメインシステムの故障系列を示す．式（8）で示したT-不変集合$Y2$の t_6, t_9 と式（9）で示したT-不変集合 $Y3$ の t_6, t_7, t_{12} の遷移 t_6 が共通遷移であるので，T-不変集合 $Y2$ と T-不変集合 $Y3$ を遷移 t_6 を根元事象とする同一の系列と見なせる．つまり，二通りの故障に対応する系

列がある．第一に，サブシステムが故障となり，メインシステムがローカル処理を代行したあとにサブシステムの修理が完了し，双方のシステムが通常の処理に復帰する．次に，サブシステムの故障がメインシステムの機能でフォローしきれないためシステム故障となり，サブシステムの修理が完了し，システム再始動（system restart）となるサブシステムとメインシステムの故障系列である．従って，b)は，システムのハードウェア的な故障を系列的に把握するものではなく，システムの機能的な面をも含んだ対策が必要となる．

3) 図4のc)は，システムのトラップ故障系列を示す．式(10)で示したT-不変集合 $Y4$ の t_1, t_2, t_5, t_{13}, t_{14} の系列であり，メインシステムがオンライン処理からローカル処理へ，サブシステムがローカル処理からオンライン処理にそれぞれ切り替えられ，メインシステムとサブシステムの交替が開始され，交替中に通信障害が発生し，システムがトラップ状態になる．つまり，システムのトラップの修復をするためのトラップ対策に対応した系列である．

4) 図4のd)は，サブシステムの故障系列を示す．式(11)で示したT-不変集合 $Y5$ の t_3, t_4 の系列であり，通信障害などでサブシステムがシステム故障となり，メインシステムによりサブシステムの機能を回復するサブシステムの故障系列に対応した修理が必要である．

このように，PNにより，システム構成要素の故障に相互依存性がある修復可能なシステムをモデル化することによって，システムの故障要因をシステム構成要素面からだけでなく，システムの機能面をも考慮し，システムにおける全ての故障に関する相互関係をPNのT-不変集合を求めることで，系統的に捕らえることが可能となる．

5. おわりに

本論文では，基本的な分散処理形態を有するシステムを取り上げ，PNを用いてシステムのモデル化を行い，PNモデルのT-不変集合を求めることによりシステムの故障要因を系列的に捕らえ，システム全体の保全あるいは診断を効果的に行う方法について考察した．

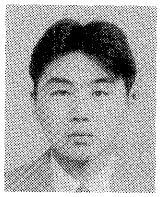
従来のシステムの故障要因を示す方法である故障木による解析における欠点は，故障木の作成において，故障要因を分析する者の経験に頼る面が多々あると共に，システムの構成要素面での解析が中心であることであった．しかし，分散処理形態を有するシステムでは，システムを構成する要素間の手順がやや複雑であり，システム全体の故障要因を系列的に把握することが困難である．そこで，本論文で提案した方法は，PNを用いることで，非同期・並列的な挙動をするシステムのモデル化を可能にし，PNモデルのT-不変集合を用いることによって，故障要因をシステムの構成要素面からだけではなく，システムの働き，つまり機能面をも考慮した故障要因を系統的に示すことによって，システムの保全あるいは診断を効率的に行うことを可能とした．ここで，本論文で取り上げた分散処理システムは，小規模かつ基本的なシステムであり，図3で示したPNモデルからシステム全体の故障に関わる要因を系列的に推測することも可能かと思われる．しかし，より現実的かつ実際的なシステムを対象として詳細なモデルを構築した場合，本論文で提案したシステムの故障要因を系統的に示す方法は，システムの機能とシステムの故障対策を考慮したPNモデルより，システム全体の故障要因を系統的に捕らえ，システムの保全あるいは診断の手順を視覚的に把握するためには，有効であると考えられる．よって，今後の課題としては，分散処理形態を有するシステムなど，処理手順が複雑なシステムである場合，あるいは，複数のユニットが相互関係を有しているシステムの故障を診断する場合などを取り上げて，より複雑なシステムへの応用を試みる必要がある．また，本論文で示した方法は，システムの要素間の機能的なつながりを系列的に捕らえるので，システムを構成する各要素（モジュール）が自分の行動を自律的に決定し，要素間の協調を図り，全体としての目的を達成する自律分散システムの協調動作の系列への応用としても検討すべきであろう．

参考文献

1) 金，菅沢，瀬谷：分散システムにおけるシンク発生の確率モデルと挙動解析，電子情報通信学会論文誌，Vol. J75-A, No. 3, pp. 658-660 (1992).
2) Johnson, A. M., Jr. and Malek, M.: Survey of Software Tools for Evaluating Reliability, Availability, and Serviceability, *ACM Comput. Surv.*, Vol. 20, No. 4, pp. 227-269 (1988).
3) Murata, T.: Petri Nets: Properties, Analysis and Applications, *Proc. IEEE*, Vol. 77, No. 4,

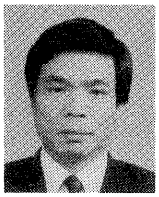
pp. 541-580 (1989).
4) Reisig, W.: *Petri Nets*, Springer-Verlag (1982).

(平成6年2月1日受付)
(平成6年6月20日採録)



勝間田 仁（学生会員）

昭和44年生．平成4年日本大学生産工学部数理工学科卒業．平成6年同大学大学院生産工学研究科数理工学専攻博士前期課程修了．現在，北海道大学大学院工学研究科情報工学専攻博士後期課程在学中．システムの性能評価，自律分散システム，ペトリネット理論の応用などの研究に従事．電子情報通信学会会員．



菅澤 喜男（正会員）

昭和43年日本大学理工学部経営工学科電気専攻卒業．昭和45年より米国ミシガン大学，ワイオミング大学等に留学．昭和49年ノースロップ工科大学大学院修士(情報理論)課程修了．以来，システム工学の研究に従事．近年，特に，非同期・並列システムの性能評価にペトリネットを応用した研究に興味を持っている．昭和57年北海道大学より工学博士を授与される．現在，日本大学生産工学部数理工学科教授．電子情報通信学会，日本OR学会，日本経営工学会等各会員．