

ペトリネット・モデルを利用した分散ソフトウェア要求の動的分析

4 Z C - 3

井上哲哉[†]創価大学大学院工学研究科^{††}

1. はじめに

近年、情報通信技術の著しい発展により情報システムの運用形態が多様化している。これにともない、情報システムで利用されるさまざまなソフトウェアは、集中型から分散型へ移行し、その構成要素は、ますます複雑化している。このことから、分散処理システムには研究すべき数多くの課題がある。その一つに分散化されたソフトウェアの信頼性が挙げられる。これは、それぞれのソフトウェアに依存するOS (Operating System) や通信ネットワークなどのシステムの信頼性に影響されることが多い。このことから、分散化されたソフトウェアの信頼性の尺度には、通信ネットワークの状態を含むシステムの性能評価を考慮する必要がある。

そこで筆者は、実際の運用において、分散化されたソフトウェアの信頼性の尺度を求めるソフトウェア要求モデルを考えた。本稿では、実際の運用の中から得られる多様なデータを統計的に分析し、その分析結果をソフトウェア要求として定義する方法について述べる。

2. 分散ソフトウェア要求の動的分析

2. 1. 従来の開発方法における限界

ソフトウェアの信頼性向上には、代表的なモデルとして、信頼度成長モデルがある。ソフトウェアの信頼度とは、ある一定の環境と期間内において、ソフトウェアのフォールト(障害)が発生しない確率のことである。信頼度成長モデルは、ソフトウェアの試験を繰り返し行い、ソフトウェアに潜んでいる欠陥や不具合によって影響を受けるフォールトを除去させながら、ソフトウェアの信頼度を改善させていくモデルである。

しかしながら、ソフトウェアの開発段階では、実際の運用の中で発生するすべての事象を網羅することは現実的に不可能である。特に、フォールトの発見は、情報システムの規模が大きくなればなるほど困難であり、ソフトウェアの開発費の大半が保守にあてられることから明らかである。このことは、分散処理システムのソフトウェア開発においても例外ではなく、その開発方法は、いまだ確立されているとは言えない。

このような背景から、より高度なソフトウェア開発技術が

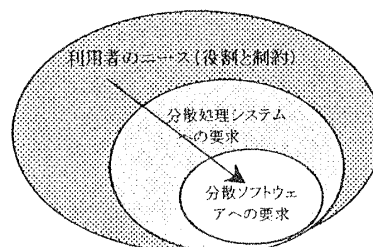


図1 分散ソフトウェア要求の包含関係

望まれる。そのためにはまず、分散処理システムの特性を明確にし、分散化されたソフトウェアの信頼性に対する尺度を求める必要がある。

2. 2. 分散ソフトウェア要求へのアプローチ¹⁾

本稿では、分散化されたソフトウェアを分散ソフトウェアと呼び、その要求を分散ソフトウェア要求と呼ぶことにする。本稿で取り扱うソフトウェアは、企業の情報系システムなどで扱われる応用ソフトウェア(application software)を指している。

さて、前述したように、分散ソフトウェアの信頼性は、そのソフトウェアを取り巻く環境に大きく関係している。図1に示すように、分散ソフトウェアへの要求は、そのひとつ外にある分散処理システムを分析することによって確定される。さらに、分散処理システムへの要求は、その外にある利用者のニーズを分析しなければ確定されない。利用者のニーズは、役割と制約に分けられ、システム、およびソフトウェアに対して次のようなものを指している。

- (1) 分散処理システム、および分散ソフトウェアの果たすべき役割の明確化
- (2) 評価基準および制約の具体化
 - システム性能評価
 - OSに関するもの: CPUの負荷, メモリやディスクなどの使用状況
 - 通信ネットワーク: スループット, 応答時間など
 - 信頼性
 - 分散ソフトウェアの信頼性を含むシステム全体の信頼度

(1)の利用者のニーズが分析される時、分散処理システムの役割では、システムの構成要素を主として、端末構成、通信制御装置、通信回線などハードウェアの側面からそれぞれの役割が明確化される。また、分散ソフトウェアの役割では、分散化されたソフトウェアを、例

Dynamic analysis of distributed software requirements using Petri nets model

[†] Tetsuya INOUE

^{††} Graduate School of Engineering, Soka University

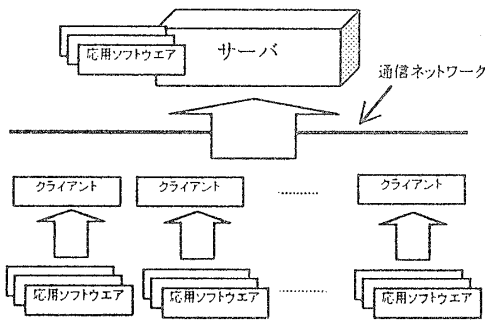


図2 クライアント/サーバ型

例えば、企業などの業務に応じた処理内容や扱うべきデータの詳細といったようなものが明確化される。(2)の評価基準および制約については、(1)で明確にされたそれぞれの要素に対して評価基準および制約が割り当てられる。

このように、分散ソフトウェアへの要求は、利用者のニーズから、それぞれの役割と制約を明確化し、具体化することにより求められる。

3. 分散ソフトウェア要求のモデル化²⁾

3.1. モデル化の定義

一般にペトリネット・モデルの構造を N とし

$$N = (P, T, A)$$

と表すことができる。

P は状態変数を表すプレースの有限集合、 T は状態遷移を表すトランザクションの有限集合、 A は情報や制御の流れを示すアークの有限集合とする。また、プレース p からトランザクション t へのアークを W と表し、トランザクション t からプレース p へのアークを W^+ と表す。そして、ペトリネット構造 M は

$$\begin{aligned} P &= \{p_1, p_2, \dots, p_n\} = \{p_i \mid 1 \leq i \leq n \mid |P| = n\} \\ T &= \{t_1, t_2, \dots, t_m\} = \{t_j \mid 1 \leq j \leq m \mid |T| = m\} \\ A &\subseteq (P \times T) \cup (T \times P) \quad (P \cap T = \phi, P \cup T \neq \phi) \end{aligned}$$

$$N = (P, T, W, W^+)$$

と表現できる。

3.2. モデル化の目的

一般に、図2に示すようなクライアント/サーバ型の分散処理システムでは、応用ソフトウェア、クライアント、サーバ、通信ネットワークの4つの構成要素に大きく分けられることができる。図2では、応用ソフトウェアからサーバへボトムアップにデータが流れていることを示している。

本稿では、4つの構成要素を使って、クライアント/サ

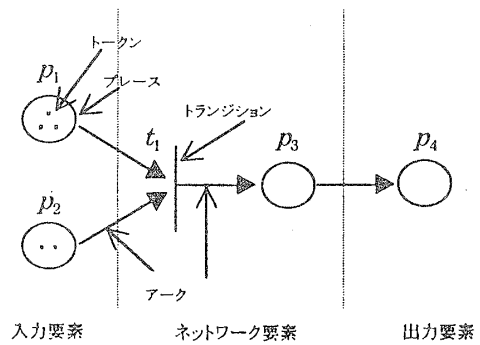


図3 分散ソフトウェア要求のモデル化

ーバ型の分散処理システムを形式的に表現するためにペトリネット・モデル (Petri Nets model) を適用する。このモデルを適用する理由には、次のことが上げられる。

- (1) 動的分析が可能なモデル
- (2) 大規模システムにも十分対応できるモデル
- (3) システムの性能評価に適するモデル

モデル化の目的としては、システムの性能評価に狙いを絞り、挙動の側面からモデル化を行う。

図3では、図2に基づいて、分散ソフトウェア要求をモデル化し、プレースを p_1, p_2, p_3, p_4 で表し、応用ソフトウェア、クライアント、サーバ(応用ソフトウェア)、分散ソフトウェア要求のそれぞれを示している。さらに、トークンは、応用ソフトウェア p_1 、および p_3 の場合は、それぞれのソフトウェアの信頼度を表し、プレースがクライアント p_2 の場合は、クライアントのシステム性能を表している。そして、トランジション t_1 は通信ネットワーク、およびサーバの状態を、アークはデータの流れをそれぞれ表している。

これらのことから、図3に示す入力要素とネットワーク要素を動的に分析し、その分析結果が出力要素となる。したがって、ここでの詳細な説明は割愛するが、この出力要素が、分散ソフトウェアの要求定義となる。

4. まとめ

以上、本稿では、ペトリネット・モデルを利用した分散ソフトウェア要求の動的分析について述べた。

今後の課題として、分散ソフトウェア要求モデルを適用した監視ツールを作成し、このモデルがどのくらいの規模のシステムまで適用できるのかを検討していきたい。また、このモデルを適用するにあたって、情報収集の仕組みを具体化していく。さらに、動的分析には、多変量解析を使って分析を行うことも考えている。

参考・引用文献

- 1) 松本吉弘: ソフトウェア工学, 電気書院, 1997
- 2) 奥川峻史: ペトリネットの基礎, 共立出版, 1995