

ネットワークアーキテクチャに関連する ソフトウェア工学の問題

服部光宏 勅使河原可海 島 直 東 基衛
(日本電気株式会社)

1. はじめに

プログラムの生産性を低くする要因としては、種々のものが考えられるが、それらのうち影響度の高いものから順に、以下のような要因を上げることができる(1)。

- (1) カスタマインタフェースの複雑さ
- (2) 要求仕様決定の際のユーザ参加度
- (3) 要員の全般的な習熟度・経験度など
- (4) 同程度あるいはより大規模・複雑なアプリケーションに対する経験
- (5) プログラミング言語に対する経験度

これらは一般にシステムが複雑になるほど問題となる場合が多い。特に、オンラインシステムやセンサベースシステムは複雑なシステムが多く、それらはネットワークシステムの形態をもつことが多い。

ネットワークアーキテクチャ開発は、ネットワークシステムを体系的に構築するための基礎的概念として、コンピュータ通信に関する統一的、標準的インタフェースあるいはプロトコルの設定を行うことと、その基礎的概念にもとづくシステム構築を支援する手法あるいはツールとからなると考えられる。

ソフトウェア開発・保守の観点からとらえると、インタフェースあるいは通信プロトコルの体系化、標準化は(1)に対応して、コスト低減に最も効果があると期待されており、近年最も活発な研究開発が行われている分野である。後者に関する研究は、元来は前者よりも早くから始められたものであり、ネットワークアーキテクチャの基礎概念が固まるに従い、その基本的な裏付けを行うツールとして、また基礎概念にもとづいたコンピュータ通信システムを構築する支援ツールの手法として、ますます重要性が認識される傾向にある。本資料では、これらの手法、ツールとソフトウェア工学との関連、それらの問題点について簡単に述べる。

2 ネットワークシステムとソフトウェア工学との関連

ネットワークシステムとソフトウェア工学との接点としては、ソフトウェア工学的アプローチによるネットワークシステム開発支援ツール/手法と、コンピュータ通信技術のソフトウェア開発(保守を含む)システムあるいは手法への応用との二つの側面が考えられる。図1にこれらの関係を示す。

2.1 ネットワークシステムの開発支援ツール・手法

ネットワークシステムの計画段階では、システムの需要予測、開発コスト、利用コストなどの予測ツールが必要である。ネットワークアーキテクチャの開発段階では、通信プロトコルの設計やその正当性の検証が必要であり、プロトコル記述手法あるいは記述言語、正当性の検証システム等が利用される。またシステム設計段階では、ネットワークトポロジーの設計、ネットワークシステムの性能/

コストの予測、個々のアプリケーションごとの性能/コスト予測、種々のネットワーク資源の配置の問題などに対するツールがあることが望まれる。

2.2 ネットワークシステム構築・利用技術のソフトウェア工学への寄与

ネットワークアーキテクチャの体系化・標準化はネットワークシステムのソフトウェア開発コストの大きな低減に寄与する（あるいはこれによりソフトウェアの開発・保守が実質的に可能になると云ってよい）。またネットワークOSの主な研究対象であるネットワーク資源（ホストコンピュータ、フェイル装置、回線、ラインプリンタ、……などの物理的資源とデータ、ユーザプログラムなどの論理的資源）管理の統一的手法は、信頼度の高い高品質のソフトウェア生産に有効である。さらに構築されたネットワーク上で、種々のネットワーク資源やソフトウェア開発ツールを利用できるようにすることは、ソフトウェアの迅速なかつ統一的な開発に有効である。さらにネットワークを用いてシステムを保守する方式は、今後の情報処理システムを保守していく上での一つの有効なアプローチとされている。

以下では、3、4節でネットワーク開発支援ツールについて、また、5、6節でネットワーク資源管理とネットワーク上でのソフトウェア開発について述べる。

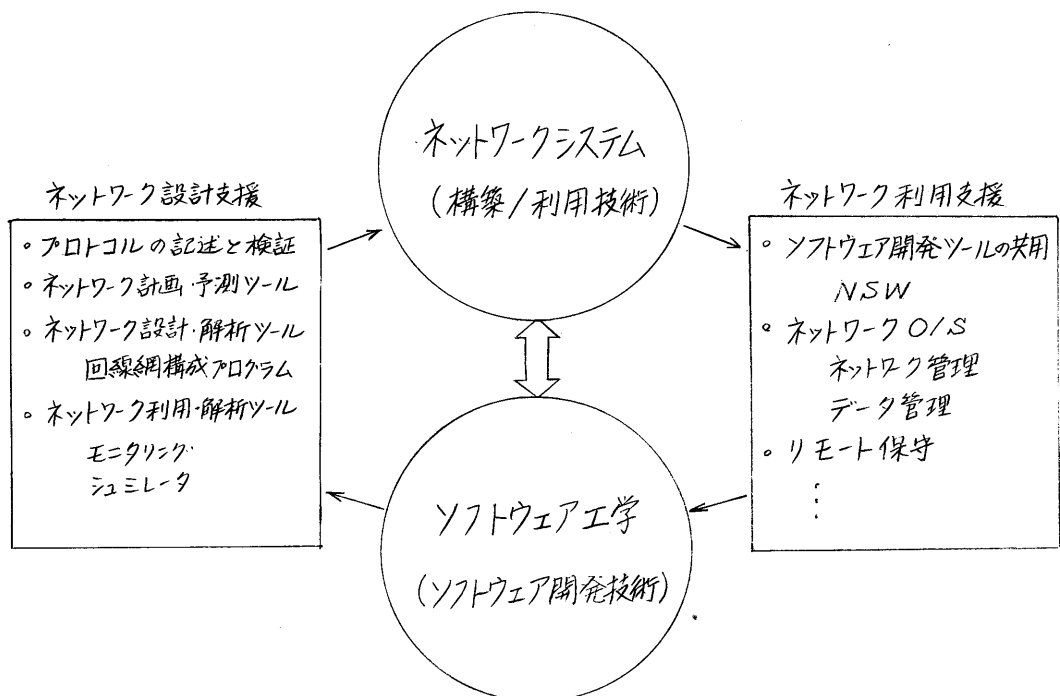


図1. ネットワークシステムとソフトウェア工学との関連

3. ネットワークアーキテクチャの設計

ネットワークアーキテクチャの設計において、コンピュータ通信プロトコルの設計とその正当性の検証は重要な位置を占める。コンピュータ通信プロトコルには、データ転送に関するものと、制御機能に関するものがあり、おのれの正当性に関する検証が必要である。

(1) データ転送

- a. データのロス、重複、転送メッセージの損傷などの回避
- b. メッセージの正しい順序での受け渡し。

(2) 制御機能

- a. 正しい初期化、終了
- b. 制御情報の同期
- c. デッドロック、ループの有無
- d. システムの故障時のプロトコルの誤り
- e. システムが永久終了状態か動作中の判定

これらの項目について、プロトコルの検証を行うためには、

- ・プロトコルの定義が正確で、一意的かつ簡潔であること、
- ・動作の正当性検証のための形式的手続きを与えられること、
- ・メッセージ・コーディング、効率などの諸問題をモデルに反映できること、
- ・モデルからインプリメンテーションが自動的に行えること、

などを満たすことが望まれる[2]。

プロトコルに従って情報の転送・制御を行う抽象的な機能単位はプロトコルマシンと呼ばれる。一般にコンピュータ通信プロトコルは階層化されているため、プロトコルマシンも階層化された層の持つ機能単位ごとに対応づけられ、プロトコルマシン間の通信とプロトコルマシンの実行によりプロトコル機能が実現される。プロトコルの正当性の検証については特に最近になり、多くの報告がなされている[3]。プロトコルマシンとしては、有限オートマトンやペトリネットをモデルにしたものが多く、主として2つの検証アプローチがとられている。一つはプロトコルマシンの状態記述を行い、種々状態への致達可能性の解析を行うもので、他の一つは、おのれのプロトコルマシン実現アルゴリズムを記述し、その正当性の証明を行おうとするものである。これらについての研究は、かならずしも(1)、(2)の全てを対象にしているわけではない。一部分を対象とした検証システムの中には、実用レベルに達しているものもあるが、対象範囲の拡大や、検証コストの低減など、今後の研究がまたれる。

4. ネットワークの解析と設計

本節では、ネットワークシステムを計画、設計する際に、必要と考えられる支援ツールについて考察する。ネットワークシステムの計画時には、データトラフィックの予測やネットワークコストの把握が必要であり、また設計にあたっては所要のパフォーマンスを満足して最も経済的なシステムを実現することと、また逆に設計されたシステムのパフォーマンスを正しく解析することが重要である。ネットワークシステムの設計に関しては、ネットワーク上に、ネットワークファシリティ（回線、集配信装置など） プロセッシングファシリティ（コンピュー

タ)。データベースファシリティを如何に最適配置するかが問題となる。これらについては、具体的な例と示して概説する。

4.1 ネットワーク計画・予測ツール

ネットワークシステムの構築にあたっては、まず最初に、ネットワークへのデータトラフィックとユーザの要求を正しく把握し、それに伴うネットワークコストの見積りが必要である〔4〕。しかもそれらは比較的長期に渡って検討せねばならない。そのためには、将来のシステムでのトラフィックやユーザの要求の予測と評価を行うためのツールが考えられる。そこで、将来における新しいデータ通信機器や通信コストの予測も含めて考えねばならない〔5〕。システムダイナミックス手法によるシミュレーションを用いてそうした予測、評価を行うツールの利用が有効であろう。例えばネットワークシステムがユーザをどの様に引きつけるかネットワーク利用の需要を予測する試みもこの一例と考えることができる〔6〕。

4.2 ネットワークトポロジの設計ツール

ネットワークファシリティの最適配置として最適ネットワーク構成を設計するプログラム／アルゴリズムの開発は、より経済的で効率的なネットワークを求めて積極的に進められている〔7, 8〕。設計すべきネットワークの形態としては、分散形とツリー形とが考えられるが、本稿では後者についてのネットワーク構成の設計ツールの一例として、回線網設計プログラムNETS (Network Economizing Techniques and Software) を以下に示す。

NETSは、ネットワーク設計を機能的に実現するために以下のサブプログラムより構成している。

(1) 集配信装置・マルチプレクサの最適配置サブプログラム

このサブプログラムは、各端末の位置、トラフィック、センタの位置などを与えて、リモートの集配信装置またはマルチプレクサの最適配置を決定するものである。低速度の端末回線と、集配信装置、時分割多重装置、周波数分割多重装置を用いて集線し高速回線に絞ることにより通信回線が高速になればそれだけ単位長あたりの伝送コストが安くなることを利用して回線コストの低減を計っている。

(2) 分岐による最適構成サブプログラム

これは、分岐方式により回線コストの低減を計るもので、回線使用の際の技術基準や回線速度の制約を満足させて最小コストの回線構成を設計するものである。分岐方式による回線網構成はポイント・ツー・ポイント方式に比べ、回線数、総回線距離の節約がなされ、さらに遠距離通減料金が適用され極めて有利である。

(3) 公衆回線の利用サブプログラム

ここでは、前記サブプログラム(1)、(2)が特定通信回線を扱っているのに加えて、公衆回線も回線網設計の対象として取扱うことにより、公衆回線の特徴と考慮しながら、更に経済的な回線構成を設計しようとするものである。

(4) 応答時間解析サブプログラム

分岐回線上の各端末に対して、メッセージの平均応答時間とパーセンタイルを計算するものである。また逆に、回線網構成時に、平均応答時間を指定された範囲内に収まるようチェックすることもできる。

その他に、グラフィックディスプレイを用いて回線網構成を逐次修正を行い、より最適な、現実的な回線網を設計するサブプログラムがある。

4.3 ネットワーク利用ツール

ネットワークシステムのシステム設計段階、運用段階では、システム解析・評価、性能予測を行う必要性がしばしば生じる。これらの目的のために使用されるツールに要請される機能あるいはデータとしては、システムの性能/コスト、システムの最適資源配置などがあり、計画段階と類似しているが、次の点において計画段階のそれと大きく異なる。

- a. ツールの利用頻度が多い
- b. 必要に応じて詳細なデータが得られること
- c. システム全体の性能/コストだけでなく、個々のアプリケーションに対するデータが要求される。
- d. ツールの利用者はかならずしもO/Sの設計者や、システム性能評価の専門家ではなく、むしろ一般的なシステムエンジニア、EDPS担当者、アプリケーション設計者が利用対象者となる。
- e. システムの動的特性が求められること。

これらの要求を満足するためには、解析的モデルによる性能評価よりは、モニタリングや、シミュレーションによる解析・評価の方が適していると云える。しかしながら、ネットワークシステムのモニタリングやシミュレーションには、データ収集時間がスタンドアロンの場合と比較して極端に増加するという欠点がある。従って、上に述べたようなツールの最も大きな問題としては、

f. 使い易いインターフェース

g. モニタリング/シミュレーションコストの低減
が上げられる〔9〕。

システム設計段階でのネットワークシステムの利用解析ツールとして分散情報システムシミュレータ〔10〕があり、シミュレーションコストを低減する方式について提案している。このシステムの評価項目を表1に示す。fに関しては、分散形情報システム記述方式〔11〕についてより深い検討が必要であらう。

表1 分散形情報システム・シミュレータの評価項目

		(評価)内容
実行環境		ネットワーク構成、ホスト/ノード/回線個別性能・特性 データベース・ファイル分散形態/管理方式、ネットワーク制御方式 ホストの制御方式、アプリケーション特性
性能	スループット	システムスループット、個別ホストスループット
	応答時間	端末応答時間、プロセス間通信応答時間
	資源使用率	CPU、周辺装置、回線、CCP、システムプログラム使用率
	詳細情報	アプリケーションプログラムのチェックポイント間通過時間、 メッセージ処理時間、メッセージ待行列の統計、 資源使用時の待時間、データベースマクロ処理時間 (最大値、最小値、平均、標準偏差、分布)
		トータルコスト、アプリケーションコスト

5. ネットワーク上の資源管理について

本節では、ネットワークシステムを有効に使うためのネットワーク管理内データ管理のネットワーク上での資源管理について記述する。

5.1 ネットワーク管理

ネットワーク上の資源を効率的に共有するためにネットワークオペレーティングシステムが考えられてきた〔12〕。ネットワークOSは、

- ① ネットワークジョブの実行、
- ② ネットワークデータのサポート

の機能を提供せねばならない。そのためには、

- ① ユーザに対して異機種間でも同一に見せるためのユーザーシステムインタフェース、
- ② システム間での転送に必要なインタフェースを提供するシステム間インタフェース、

の機能が必要であるが、

システム間インタフェースについては、

- ① システム間でのデータ転送機能としてのプロセス間通信（IPC: Interprocess Communication）
- ② 異機種間で意味のあるデータをアクセスしたり保護する機能としてのリモートレコードのアクセス、

の機能が必要となる。特に、プロセス間の基本的なデータの開始、プロセス間のデータの転送制御を提供するIPCが重要な機能となる。

IPCの問題点としては主として

- a. プロセス間の同期
- b. データの安全性などのデータ保護、
- c. プロセス間通信プロトコルの定式化、

などが挙げられるが〔13〕、1つのジョブをいくつかのジョブステップに分けて分散されたプロセッサで実行される様な場合では、プロセス間での同期のタイミングをうまく取ることが特に問題となる。

またネットワークOSには、ネットワーク資源を有機的に管理する機能が必要であり、この機能を加えて上に挙げた機能を実現するために、ソフトウェア工学からのアプローチとしては分散/集中されたネットワークOSのソフトウェア構造の検討が課題となる。

5.2 ネットワーク上のデータ資源管理

ネットワーク上のデータ資源管理に関する技術的な問題は分散データベースの問題に集約される。分散形データベースシステムは広義には「物理的に分散したデータベース間に論理的関係のあるシステム」で、最近になり特に活発な研究が行われるようになった分野である〔14, 15〕。分散形データベースのアプローチには、既存の集中形（スタンドアロン）データベースの分散化によるものと互いに独立なスタンドアロンシステムのネットワーク結合とによるものがあり、おのこの特長が異なるが、一般的には、以下に記すような特長をもつとされている。

- a. データ資源の有効利用、
- b. 情報処理システムの組織構造への適合性、

- c. 通信コストの低減
 - d. 高応答性
 - e. 障害の局所化
- 一方、分散データベース構築上の技術的に大きな課題としては、
- f. データベースプロトコルの標準化
 - g. データの並列アクセスの制御
 - h. システムの障害時のリカバリ対策

などがあり、このほかに、分散問合せ処理やデータ/ディレクトリの最適配置などの問題がある。以下に（gを中心に）f, g, hについて簡単に述べる。

(1) データベースプロトコルの標準化

データベースプロトコルは、ネットワークアーキテクチャにおけるプロトコルの中では高いレベルの階層に属し、より下位の層の標準化にともない、次第に標準化が進むものと思われる。そのためには、分散形データベースアーキテクチャにおいて、データ（モデル）の互換性の考慮は当然として、分散データベース管理構造が明確化される必要がある。また以下に述べるように、並列制御やリカバリに関する方式の統一も必要である。

(2) データの並列アクセス制御

分散形データベースの属性には、

- a. 一つのデータを物理的に異なるノードから共有する（共有データ）
- b. 同一データを物理的に異なるノードに配置する（重複データ）

とがある。aは多重プロセッサ環境にあるスタンドアロン・データベースシステムの延長として考えられるが、bは分散型データベースにおいて新たに生じる利用形態である。重複データを持つことにより、応答時間の短縮、負荷の分散などの利点が生じるが、一方、重複データの更新時におけるデータの完全性の維持が大きな問題となっている。

重複データの更新には即時的更新（同期的）と非即時的更新（非同期的）とがある。非即時的更新の統一的研究はほとんど皆無であるが、即時更新については多くの報告がある〔14, 16〕。即時更新方式としては、更新を行なうプログラムまたはホストを1つに限定する方式がある。この方式では複数のホストでの同時更新ができないため、特定のアプリケーション環境にしか適用できない欠点がある。複数プログラムによる同時更新が可能な方式としてスタンドアロン・データベースシステムからの拡張である集中管理方式がある。この方式は管理構造が簡単であり、実用性も高いが、特権ホストを置く必要がある、その他に特権ホストを必要としない方式として、物理同期方式、多数決方式、論理時刻方式などが提案されている。論理時刻方式は、重複データベース単位に論理的なクロックをもち、論理クロック間で同期をとりながら並列制御を行うもので、他の方式と比較して応答性、負荷の平均化などにおいて優れた特性をもっている〔16〕。

(3) リカバリ

集中形データベースにおいてもリカバリは最も大きな課題であり、分散形データベースの場合は問題により複雑にしている。分散形データベースのリカバリとして問題になる点は、

- a. 障害の検出・通知
- b. 障害発生時のリカバリ手順
- c. エラーの伝播

などがあり、またリカバリが容易なアーキテクチャの開発も重要な問題である。分散データベースに関するリカバリの研究はまだ目が見え、今後最も検討を要する対象である。

6. ネットワーク上でのソフトウェア開発

ネットワークを利用したソフトウェア開発支援としては、ネットワーク上のコンピュータシステムにある支援ツールを他のコンピュータからアクセスする場合と、ネットワークを利用した保守とがある。前者の例としてはNSW (Network Software Works) があり、後者は一般にリモート保守と呼ばれている。

(1) NSW

今まで、スタンドアロンシステム上では多くのソフトウェア開発ツール——コンパイラ、アセンブラ、エディタ、デバッガ、ドキュメントプロセッサ——が考えられ個々に利用されてきた。NSWはARPAをベースとして、その上のコンピュータシステム上に構築されたこれらのソフトウェア開発ツールを統合化し、他のコンピュータシステムから共有できるようにすること、すなわち、分散プログラミング環境を導入することを目指している。NSWはプロシジャコールにベースをおき、プロセス間通信に考慮が払われていない。また、本来、資源の共有は、データにベースをおくべきであり、処理にベースをおくべきではないなどの批判もないわけではないが、このようなシステムがソフトウェアの生産性向上に直接あるいは間接に寄与する可能性は大きい。

(2) リモート保守

リモート保守に関する機能としては

- a. ローカルシステムのリモートからの診断
- b. ローカルシステムの保守情報のリモートからの収集
- c. ローカルシステムに対する保守情報からのリモートに対する提供

などが考えられる。これらは、ローカルシステムの地理的条件が悪い場合、ローカルシステムのシステム管理能力が低い場合、ローカルシステムのシステム管理者が不足している場合などに有用であり、すでに一部で利用されている。ソフトウェアのライフサイクルにおける保守に占める割合は特に大きいため、リモート保守機能の発展により、ソフトウェア保守費用の低減をはかることが期待される。

7. おわりに

これまで、ネットワークシステムとソフトウェア工学との関連を述べてきたが、ネットワークシステムとソフトウェア工学との結びつきは、ネットワークシステムを設計、構築するためのソフトウェアエンジニアリングの活用と、完成されたネットワークシステムを一つのツールとしてソフトウェア工学に利用することが考えられる。特に後者については、前節で示したが、今後の重要な課題となろう。即ち、ソフトウェアの開発場所を固定したり、リモートジョブエントリシステムを利用するだけでのソフトウェア生産性が高まることが知られていることからし

1). ネットワークシステムを有効に利用することにより、ソフトウェア開発メンバの有機的な結合や資源共有によるコンピュータパワの最適利用などがもたらされソフトウェアの生産性の向上を計ることができよう。いずれにせよネットワークシステムはソフトウェア工学と更に結びつきを深めることによりお互いに進歩発展していくことが大いに期待される。

文献

- [1] C.E. Walston and C.P. Felix, "A Method of Programming Measurement and Estimation," IBM Systems Journal, Vol. 16, No.1, 1977, pp.54-73.
(日経エレクトロニクス 1978.9.18, pp.186-196).
- [2] 「分散システムに関する調査」電振協・国際動向専門委員会報告 53-C-352, 昭和53年3月, pp.14-32.
- [3] C.A. Sunshine, "Survey of Protocol Definition and Verification Techniques," Proc. Computer Network Protocol Symposium, Feb. 1978, pp.F1.1-F1.4 (ACM SIGCOMM, vol.8, No.3, 1978, pp.35-41).
- [4] W. Chou, "Problems in the Design of Data Communications Networks," ACM SIGCOMM, Vol.4, No.2, 1974, pp.1-6.
- [5] L. Hopewell, "Management Planning in the Data Communications Environment," Proc. NCC 74, 1974, pp.561-564.
- [6] H. Miyahara, Y. Takahashi and T. Hasegawa, "Estimation of Users' Behavior in a Computer System," IEEE Trans. on System, Man and Cybernetics, Vol. SMC-8, No.1, 1978, pp.37-41.
- [7] M. Gerla and L. Kleinrock, "On the Topological Design of Distributed Computer Networks," IEEE Trans. on Communications, Vol. COM-25, No.1, 1977, pp.48-60.
- [8] Y. Teshigawara, I. Wakayama, K. Wakabayashi and T. Yoshimura, "A Communication Network Design Tool-NETS," Proc. COMPCON 78 Fall, 1978, pp.158-165.
- [9] D.E. Morgan, W. Banks, D.P. Goodspeed and R. Kolanko, "A Computer Network Monitoring System," IEEE Trans. on Software Engineering, Vol. SE-1, No.3, 1975, pp.209-311.
- [10] 西原 真名垣 金子 服部, 「分散型情報システムシミュレータ」情報処理, コンピュータネットワーク研究会 CN20-3, 1977, pp.1-10.
- [11] Shi-Kuo Chang, "A Model for Distributed Computer System Design," IEEE Trans. Systems, Man and Cybernetics, Vol. SMC-5, No.6, 1976, pp.344-359.
- [12] S.R. Kimbleton, H.M. Wood and M.L. Fitzgerald, "Network Operating System - An Implementation Approach," Proc. NCC 78, 1978, pp.773-782.
- [13] Proc. ACM SIGCOMM/SIGOPS Interprocess Communications Workshop, 1975.
- [14] J.B. Rothnie and N. Goodman, "A Survey of Research and Development in Distributed Database Management," Proc. VLDB 1977, pp.48-62.
- [15] 情報処理学会第19回全国大会講演論文集, 昭和53年, pp.945-960
- [16] 西原 金子 鶴岡 服部, 「分散型データベースシステムにおける重複データ制御方式」信学会計算機研究会 EC 78-6, 1978, pp.11-22.

- [17] R.A. Robinson, "National Software Works: Overview & Status,"
Proc. COMPCON 77 FALL, 1977, pp.270-273.