

## 解説



## ネット指向パラダイムを求めて

## 6. リアルタイムシステムにおけるネット指向 開発技術の適用†

伊藤 潔†† 杵嶋 修三†††

## 1. はじめに

リアルタイムシステムは、多くの自律的に稼動する構成要素からなり、必要に応じてこれらの構成要素の間で同期信号やメッセージなどを送受し、外界に対してきわめて迅速に応答を行うシステムである。リアルタイムシステムを対象とした従来のソフトウェア生産技術の研究では、主に、**トランザクション処理**（たとえば、[Yourdon 72], [Gray 93]）を中心としたデータシステム（EDP, オンラインバンキングなど）を対象とした。ここでは、主に、コンピュータシステム内部でのタスクの**機能設計**の観点から、リアルタイムシステムのソフトウェア開発の問題をとらえた。

一方、プラントの計測・制御・監視を目的とした産業用アプリケーションも重要なリアルタイムシステムである。近年、この**計測制御リアルタイムシステム**は、**リアクティブシステム**〔Boussinot 91, [Harel 87, 88, 90], [Kishima 92]〕とも呼ばれ、この概念に基づいたアプローチが、特に欧米の研究で新しい流れを形成しつつある。

計測制御システムは、図-1に示すとおり、制御する側の**スーパーバイザ**と、制御される側の**プラント**からなる。これらは、互いに相手の**アクション**に対してリアルタイムな**リアクション**を行う。計

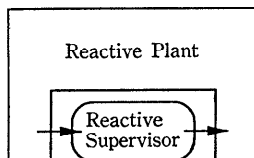


図-1 Instrument and Control  
Real-Time System's Organization

† Application of Net-Oriented Development Methods to Real-Time Systems by Kiyoshi ITOH (Sophia Univ.) and Shuzo KISHIMA (Yamatake Honeywell).

†† 上智大学理工学部

††† 山武ハネウエル(株)アドバンステクノロジーセンター

測制御システムでは、スーパーバイザと外界のプラント（プラントプロセス、プラント装置、プラントオペレータなど）が、センスとコントロールの関係で密接に結び付いて相互に作用する。この相互作用により、プラント全体の操業上の目標（物の生産・加工や環境の維持など）を実現する。この開発では、システム内部のタスクの設計に加えて、対象となるプラントプロセスを明確に認識する必要がある。関与するオペレータや各種装置の役割も対象となる。制御対象となるプラントの状態は、さまざまな物理的特性一空調制御システムでは、温度・湿度・塵埃・においなどに支配される。

一般のデータ処理のソフトウェアやトランザクション処理のリアルタイムシステムの要求分析や設計の仕様記述のために、さまざまな**ネット**（**ダイアグラム**, **チャート**）を用いた方法が提案されている（たとえば [DeMarco 79], [Jackson 75], [Martin 85], [Yourdon 78] など）。ネットを計測制御システムに適用した事例は、多くの文献（たとえば [Cooling 91], [Laplante 93], [Otsuki 90], [Shlaer 92], [Yamamoto 90]）に詳しく述べられている。Hatley は、計測制御システム向けのオリジナルなネットを考案した〔Hatley 87〕。

計測制御システムの実際開発では、スーパーバイザとプラントからなる全体の**アーキテクチャのひな型**を想定している。その上で、さまざまな方法論やツールで分析・仕様化を行い、実現すべきシステムの構造と機能を具体化している。

現実のリアルタイムシステムの分析・仕様化では、そのシステムのさまざまな側面や性質を記述しなければならない。もし、このためにネットを使用するとすれば、側面や性質に応じて多くの種類のネットを使い分けて記述せざるをえないのが現状である。解析能力をもつといわれているべ

トリネットモデルなどの守備範囲は、同期分析など、必ずしも広いものではない。リアルタイムシステムの分析・仕様化へ適用するネット指向技術を概観すると、解析能力の有無に関わらず、多くの種類のネットを使い分けて実際的な例に適用することが現状では有効である。オリジナルな事例に適用した結果、万能なネットはなく、次の5種類に既存のネットを類別して適材適所で相補的に使い分けることが有効である、との認識を得た。それらは(a)機能分析・仕様化向きネット、(b)挙動分析向きネット、(c)タイミング・同期分析向きネット、(d)プログラム構造設計向きネット、(e)性能分析向きネット、である。

2. では、事例として扱った実際的な計測制御システム(空調制御システム)のアーキテクチャのシステムチャート、3. ではネットの類別と個々を用いる際の分析・仕様化の観点、4. では類別した個々のネットの説明と適用例を述べる。5. では、これらの適用例を作成した際に得た印象を述べる。

2. 計測制御システムのアーキテクチャ事例

システムの開発を行う場合、一般に、システムの稼動するターゲットとなるアーキテクチャのひな型を想定していることが多い。たとえば、EDPなどの小規模の事務処理システムのひな型として、数台の端末、数種の帳票、数種のファイル、

単一コンピュータのアーキテクチャを想定する。トランザクション処理では、EDPよりも多数・多種の端末・帳票・ファイル・データベースと、複数コンピュータによる分散処理と通信を行うアーキテクチャを想定する。計測制御システムでは、負荷・役割・危険を考慮した分散処理構造と、対象となるプラントに装備された多種多様なセンサ群やアクチュエータ群からなるアーキテクチャを想定する。

図-2 は、計測制御システム(空調制御システム)のアーキテクチャのひな型のシステムチャートである。ビル内の各フロアまたはルームを制御対象のゾーンとみなす。監視制御装置(スーパーバイザ)として、(1)ビル全体の電力管理と火災報知を行う中央監視装置(Building Facility Total Management Computer)、(2)各ゾーンの環境を快適に維持するための空調制御と火災検出を行うゾーン監視装置(Air Conditioning Zone Controller)、(3)ゾーンへの供給空気の状態を調整/制御する空気生成制御装置(Air Generating Controller)、がある。

以上のスーパーバイザは、センサからの入力に基づいてプラントの状態を同定し、プラントの状態を一定の目標値に保つように制御出力を行う。また、火災センサからの入力に基づいて、プラントの異常状態を検出し、必要な制御と報知を行う。

制御対象プラントを詳しくみると、(1)空調

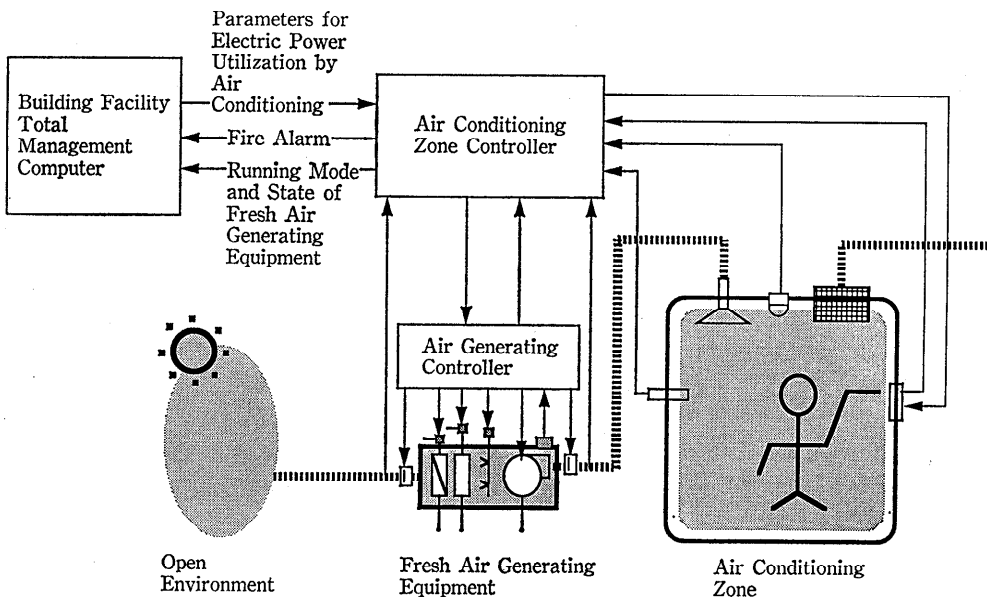


図-2 Overall System Chart—Air Conditioning Zone Control

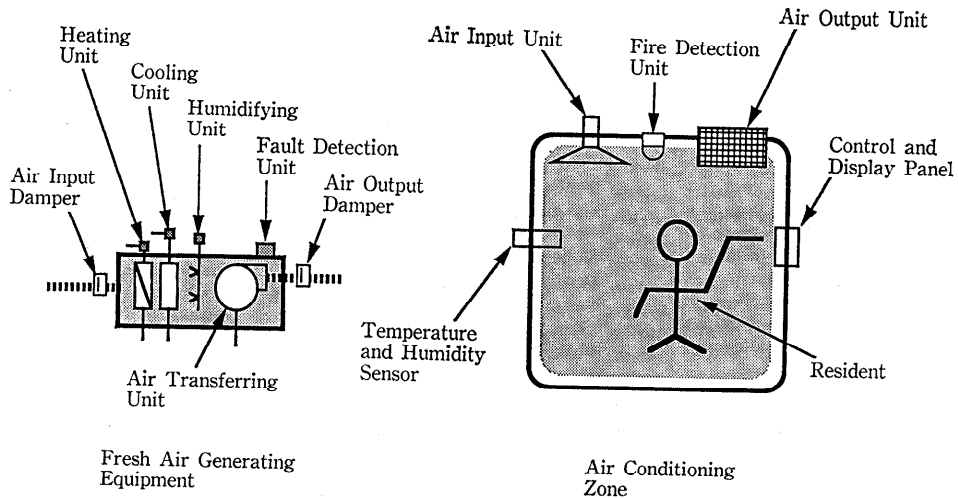


図-3 Explanation of Components in Overall System Chart

ゾーン (Air Conditioning Zone), (2)空調機 (Fresh Air Generating Equipment), (3)外界 (Open Environment) から構成される。空調機とゾーンの構成要素を図-3 に示す。空調機には冷房・暖房の運転モードがあり、この切り替えはスーパーバイザの指示で行われる。ゾーン内の居住者もプラントの中の能動的な主体である。個人的な感覚に基づいてゾーンが快適か否かを判別し、独自に冷暖房の切り替えを指示する。ゾーン内の状態は、発生する負荷 (人数, 人の活動量・着衣量, 照明など) と、外界の状態 (外気温度, 日射量, 方位など) に依存して常時変動する。

### 3. 分析・仕様化上の観点によるネットの種類

計測制御システムの開発では、2. で述べたアーキテクチャのひな型上で、どのように分析・仕様化して、実現すべきシステムの構造と機能を具体化すべきかについての手順や方法論が重要である。この際に既存のさまざまなネットを用いる場合、一種類のネットが万能ではなく、次の5種類に類別して適材適所で相補的に使い分けことが有効である。以下に、個々の種類のネットを用いる際の分析・仕様化上の観点を述べる。

#### 3.1 機能分析・仕様化向きネット

計測制御システムは、構成要素および入出力情報・信号が多種多様であり、きわめて複雑に密接に結び付いて相互に作用する。この相互作用により、プラント全体の操業上の目標 (物の生産・加

工や環境の維持など) を実現する。

機能分析・仕様化向きネットで、ひな型のアーキテクチャの上で、具体的に実現すべきシステム内の多種多様な構成要素 (スーパーバイザとプラント) と互いの複雑で密接な関係を認識・列挙して、具体化・実際化する。構成要素は、入出力情報と信号、センサとアクチュエータ、バッファ、各種インタフェース装置などの物理的構成要素と、それらを結び付けるアクション、スーパーバイザ内のタスク、リアルタイムな通信メッセージなどの論理的構成要素とに分かれる。

#### 3.2 挙動分析向きネット

計測制御システムは、さまざまな外界からのイベントに反応して多様に動作する。物理的な信号は連続系である場合が多く、信号の種類ごとにしきい値を決めてイベントを認識する。イベントが生起するとシステムはそれを処理するアクションを起こす。その結果、システムはある状態に移行し、次に生起すべきイベントを待つ。挙動分析向きネットでは、イベント・タスク・アクションを認識・列挙し、次に、タスクの実行にともなう変化を状態として列挙した後、実行順序関係を状態遷移として分析する。

#### 3.3 タイミング・同期分析向きネット

計測制御システムでは、複数の入力信号や情報が揃わなければ動けないという同期・待合せを行わなければならない状況や、揃うと危険なためフェールセーフの観点から検出・排除を行わなけれ

ばならない状況がある。タイミング・同期分析向きネットでは、タスクの起動時に、同時に複数の入力信号や情報のどれを必要とするか、あるいは揃ってはいけないのかの待合せや、タスクの実行後に同時にどの複数のタスクに出力データを送信するかなどの出力の同時分配を分析する。

3.4 プログラム構造設計向きネット

以上の三種類のネットを用いて実行の順序関係や必要な入出力が判明する。プログラム構造設計向きネットでは、全体のプログラム構造として実現する処理群を設計する。

3.5 性能分析向きネット

以上の4種類のネットにより、計測制御システムがもつべき機能を決定し、初めに想定したひな型のアーキテクチャを具体化する。この具体化の**実現可能性 (feasibility)**を、かなり綿密に分析する必要がある。計測制御システムでは、応答時間、スループット、負荷、資源の割付けなどの**性能分析**が必要不可欠な作業（このような項目に対して厳密な性能分析を必要とするシステムは特に**ハードリアルタイムシステム**と呼ばれる）で、これが機能分析・設計の実現可能性を保証する。この目的のため、性能分析向きネットはきわめて有効である。

4. ネットの適用事例

4.1 機能分析・仕様化向きネット

(a) コンテキスト図 (context diagram)

DeMarco によると、コンテキスト図は、システムのすべての入出力を描いた最上位レベルのデータフロー図である。このレベルでは、システムの機能自体の分割はまだなされない<sup>〔DeMarco 79〕</sup>。

図-4 では、ゾーン監視装置の主要な機能・役割を空調制御と火災検出 (Air Conditioning and Fire Detection) ととらえる。この機能を果たす上で必要な入出力情報と信号を確認する。また、それらの発生源と吸収源を認識する。機能を中心に置いて、認識されたものすべてとアークで関係付ける。

(b) 実体関連図 (entity relationship diagram: ERD)

ERD では、システム内の構成要素と互いの論理的/物理的関連の列挙を行う。

図-5 の ERD では、図-4 のコンテキスト図で列挙した情報、信号、それらの発生源、吸収源を総称して**実体 (entity)** ととらえる。実体どうしを結び付けるアクションで、実体を関連付ける。これらのアクションの主なものは、図-4 で認識した装置の主要な機能・役割を実現する部分機能である (図-5 右の Rates, Actuates, Controls)。

列挙された実体に対して、アクションを行うほかの実体を一つずつつづる式に認識し、可能な限

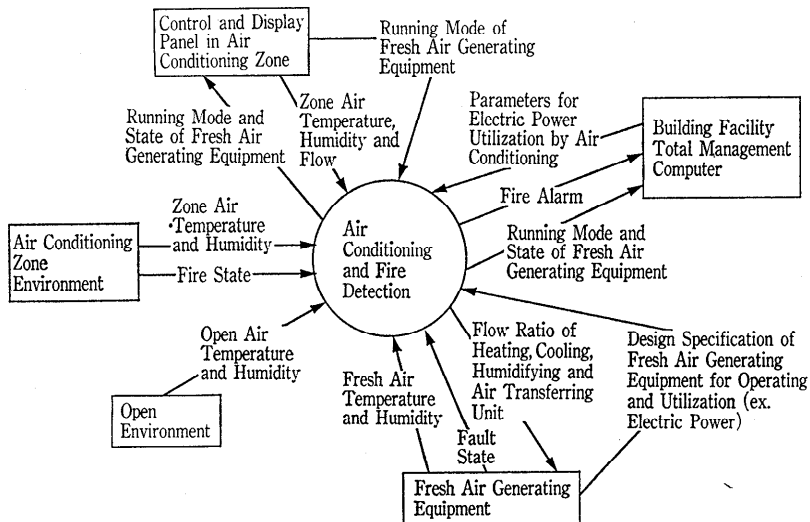


図-4 Context Diagram—Air Conditioning Zone Control System

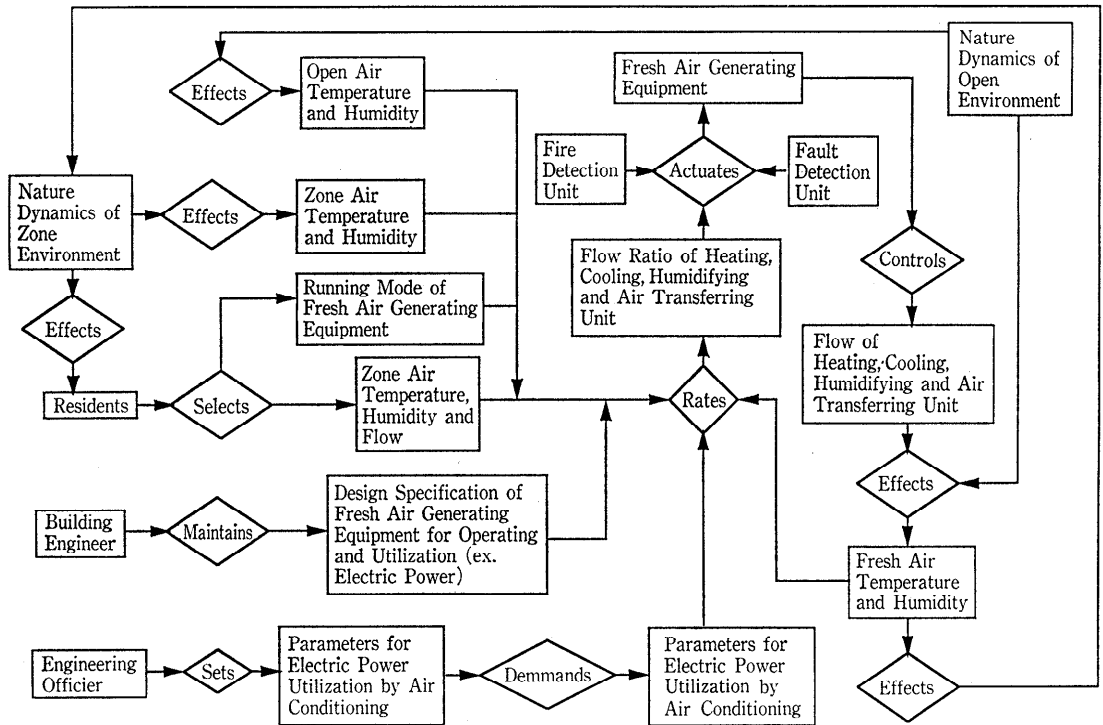


図-5 Entity Relationship Diagram

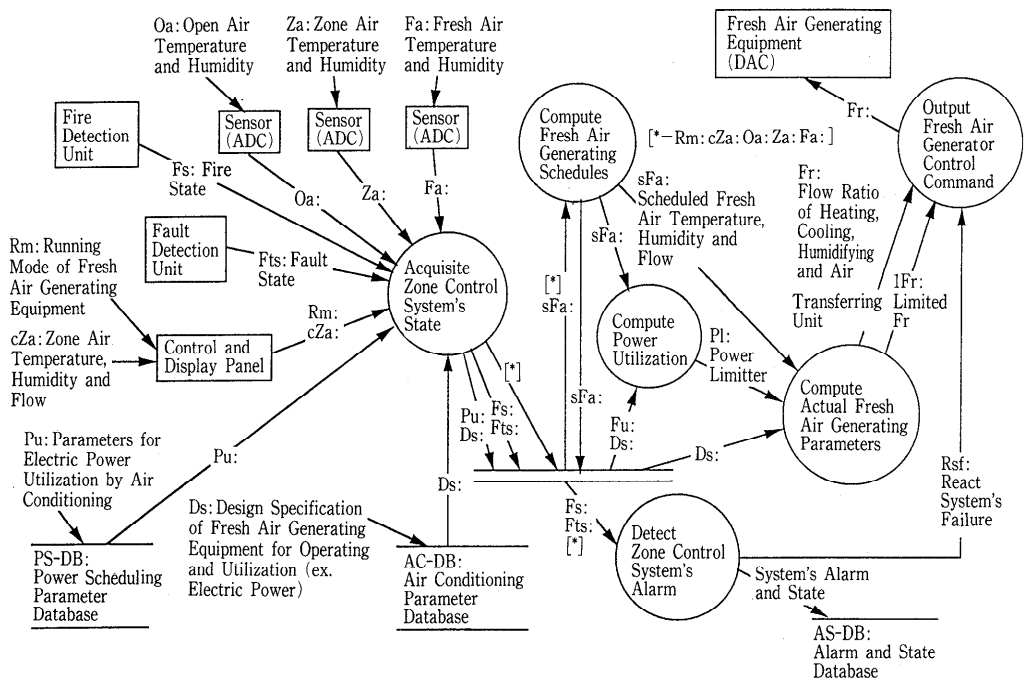


図-6 Data Flow Diagram—Air Conditioning Zone Control System

り ERD を増強してよい (図-5 左の Residents, Building Engineer, Nature Dynamics of Open Environment などの実体. Selects, Maintains, Effects などのアクション).

(c) データフロー図 (data flow diagram: DFD)

DFD では、情報・信号の発生源や吸収源、バッファ、処理をノードとして、ノード間に流れる情報・信号のフローをアークとして表す。

図-6 の DFD は、図-5 の ERD で認識したアクションからスーパーバイザのアクションを選別して、そのアクションを行うタスク群を列挙する (図-6 右のたとえば、Compute Fresh Air Generating Schedules, Detect Zone Control System's Alarm, Output Fresh Air Generator Control Command)。

タスク間のリアルタイムな通信メッセージを認識し、入出力アークに記述する (図-6 右のたとえば、Power Limiter, React System's Failure)。

システムの過去の履歴や多重のデータを蓄積しておき、その傾向を使ってタスクが起動されるとき、それらのデータを保存するバッファを列挙する (図-6 下のたとえば Power Scheduling Parameter Database, Air Conditioning Parameter Database)。

外界とタスクのインタフェースを実現するセンサ・アクチュエータを列挙する (図-6 左のたとえば、Sensor, Control and Display Panel, Fresh Air Generating Equipment)。計測制御システムではこのインタフェースの認識はきわめて重要である。

(d) リアルタイム SA (Real-Time Structured Analysis) 図

リアルタイム SA 図は、DFD を拡張して、機能 (タスク) 間のアークとして、これまでと同様にデータを記入する実線アークに加えて、起動のタイミングを表すイベントを記入する破線アークを導入し、リアルタイムシステム向きにした。

図-7 は、空調機の通常の制御サイクルと、火災検出時の処理サイクルを示している。

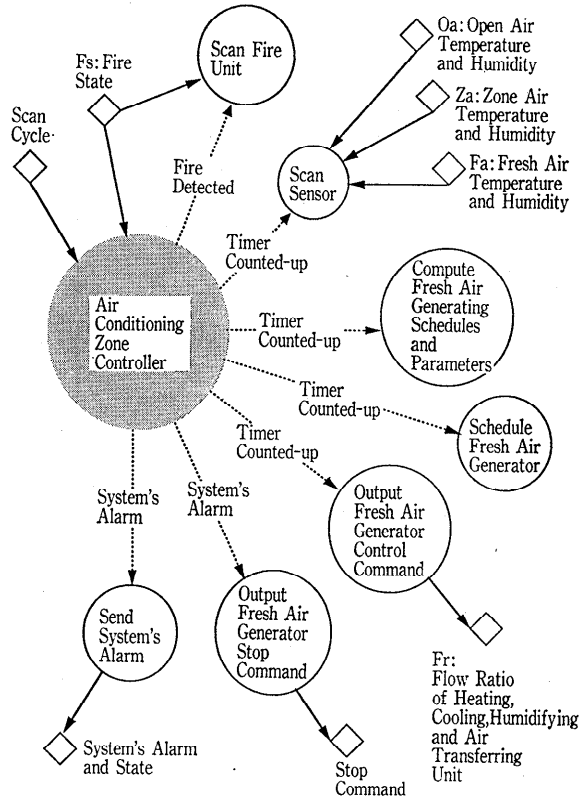


図-7 Real-Time SA

4.2 挙動分析向きネット

(a) 状態遷移図 (state transition diagram: STD)

STD では、各イベントがシステムの状態にどのように変化を与えるかを表す。ノードは状態を、アークはイベントを表す。

図-6 の DFD では、スーパーバイザを構成するタスクと、必要な入出力を列挙するが、タスクの実行順序関係は明示されていない。これに対して、図-8 の STD は、タスクの実行にともなう変化を状態ととらえて、状態を列挙した後、実行順序関係を状態遷移として定義する。

計測制御システムでは、スーパーバイザの状態を初期状態 (Initial State) と、イベント監視状態 (Monitoring State) および、各イベントごとのイベント処理状態群 (たとえば、Scan Panel, Send System's Alarm) として、定義することが有効である。

(b) ステートチャート (Statechart)

Statechart<sup>[Harel 87, 88, 90]</sup> では、計測制御システムを、外界からのイベントに反応する機能集合で構

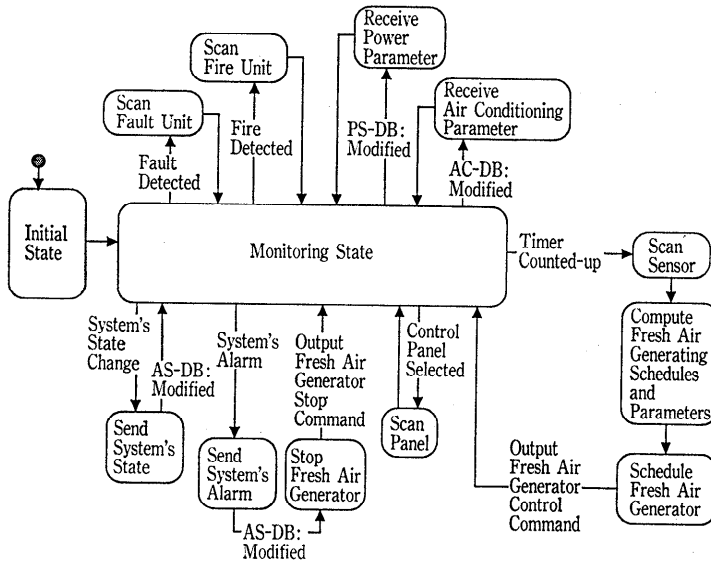


図-8 State Transition Diagram

成されたシステムと認識する。イベントに反応するシステムを順次詳細に分割する手順を与える。この分割のために、STDの階層化設計を行う。

上位の機能は一つのSTDで表され、その機能を実現するためには、下位の部品群を列挙する。各部品の機能もSTDで表す。図-9では、ゾーン監視装置の機能のSTDが上位のSTDとなる。下位のSTDは、空気生成装置、制御パネル、タイマ装置である。

STD どうして状態遷移の共有が可能である。すなわち、あるSTDの状態遷移が、ほかのSTDで

の状態遷移を引き起こせる。図-9では、ゾーン監視装置の状態遷移 (Oc: Output Fresh Air Generator Control Command) が空気生成装置の状態遷移 (Start-cmd, Stop-cmd のいずれか) を引き起こす。

(c) 三群状態遷移図 (STD Triad)

スーパーバイザは、プラントを目標とする状態にするための強制的なアクションをもつ。プラントは、プラント自体の自然な性質や状態の変化に応じて自律的に行われるアクションをもつ。この認識の下に、杵嶋・伊藤の Asdreas (Kishima 92) ではSTD Triad を考案

した。図-10 のとおり、STD Triad は、(1) 自然状態群 (State Group by Nature Dynamics), (2) 作用状態群 (State Group by Nature Dynamics), (3) 目標状態群 (State Group by Heuristic Threshold) からなる。

プラントの Nature Dynamics (ゾーン内で発生する負荷や外界の状態の変化) を明確に認識するために、自然状態群を用いる。たとえば、プラントの温度の自然上昇などはこの状態群に属する。

プラントの Nature Dynamics のみではプラントが維持されない。作用状態群は、目標に向けて

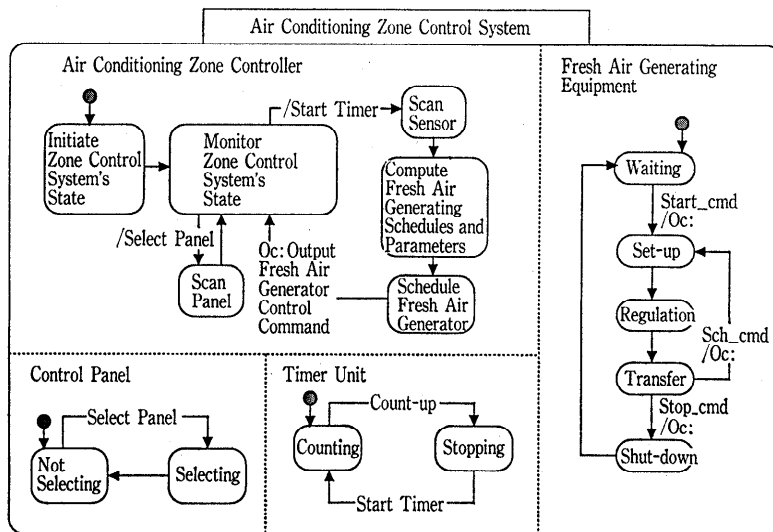


図-9 Statechart

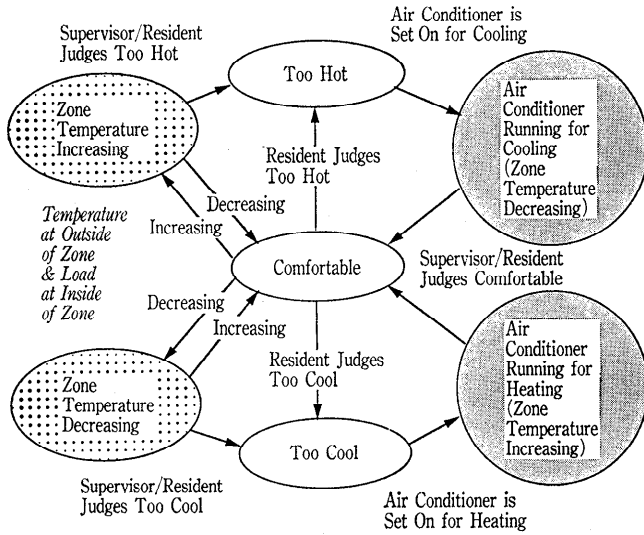


図-10 STD Triad—Air Conditioning Reactive Plant

プラントを強制的に制御するためにスーパーバイザの機構がもつ状態群である。プラントの自然な温度上昇に対して働く冷却機構などはこの状態群に属する。

目標状態群は、目標値に関わる経験的なしきい値で類別された状態群を表す。たとえば、プラントの目標温度に対する High Temperature はこの状態群に属する。計測制御システムには、必ず感覚的・経験的な判断のしきい値があると Asdreas では認識する。

4.3 タイミング・同期分析向き ネット

(a) ペトリネット (Petri Net)

ペトリネット(たとえば [Peterson 84]) には、システムの中で起こる動作(事象)を表すトランジションと、その動作が起こるための条件を表すプレースの二つのノードがある。プレースの中にトークンがあることが条件の成立を表す。

図-5 の ERD では、タスクと入出力をすべて列挙するが、タスクのある起動のときに、同時にどの複数の入力データを必要とするか、または、あるタスクの実行後に同時にどの複数のタスクに出力データを送信するかが、明らかではない。

図-11 のペトリネットでは、複数の入力データの待合せや出力の同時分配を明示できる。たとえば、上部の火災検出ユニットが、火災を検出し、かつスーパーバイザがモニタしていた場合は、スーパーバイザは、火災ユニットのスキャンを行う。また、スーパーバイザが、火災を同定した場合は、空調機停止コマンドとアラームが同時に送信される。

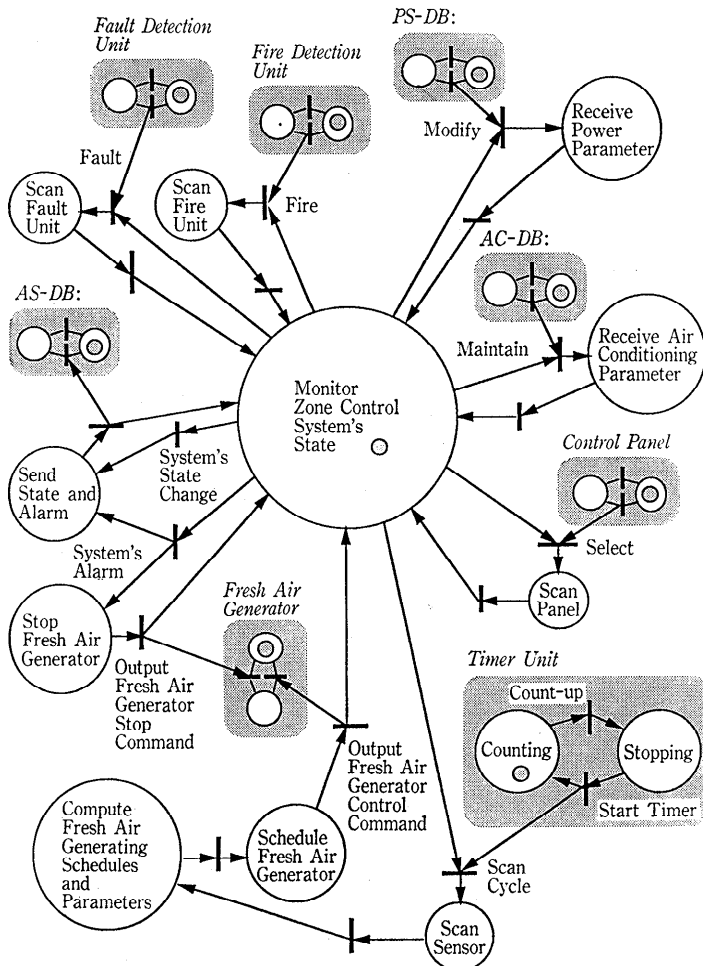


図-11 Petri Net



(b) 時間/確率ペトリネット (Stochastic Petri Net)

初期のペトリネットには、時間の概念は存在しない。すなわち、トランジションは、そのすべての入力プレースにトークンがあれば発火が可能であるが、その発火に時間はかからない。現実的なシステムを正確にモデル化するためには、時間の概念が必要である。時間や確率変数を導入した時間/確率ペトリネット〔Florin 89〕,〔Razouk 85〕が考案され、トランジションの発火時間間隔、トランジションの発火に要する時間やプレースの滞在時間などを、確率分布に従う乱数で与える。

(c) R-Net

R-Net〔Bell 77〕では、機能の流れを意識して、リアルタイムシステムへの入力信号に対する応答を発生する処理系列(スレッド)を記述する。R-Netは、シミュレーションプログラムに変換され、その機能の正しさを検証できる。

図-12の左のスレッド群は、空調機の通常のサイクルセンサのスキャン、空調パラメータのスケジュール、空調機への指示一を表す。右は、火

災を検出し、空調機を停止し、アラームを送出するスレッドを表す。

さらに R-Net は、4.5 で述べる性能分析も可能である。この性能の評価の基準として、性能要求を、R-Net 上の任意の点に明記できる。性能要求とは、その点で収集すべき性能情報の指定や、その点で満たすべき性能条件である。

4.4 プログラム構造設計向きネット

(a) シーケンス機能図(sequential function chart: SFC)

制御対象のプラントに設置されるセンサとアクチュエータを使って、一定のシーケンスでプラントの状態を遷移させてプラントの挙動を制御できる場合、スーパーバイザの制御プログラムは、シーケンス機能図で容易に設計できる。SFCは、順序制御を表す STD に、センサとアクチュエータに関わる情報や信号を付加して定義する。

図-13の SFC では、空調機 (Fresh Air Generating Equipment) を対象プラントとして、その状態を、一定のシーケンスで遷移させて制御する空気生成制御装置 (Air Generating Controller) の制御プログラムを設計する。

たとえば、空気の調整のために関連するゲートをセットアップする状態 (Air Flow Gate Setting-up for Regulation) では、空調機プラントの空気吸入側ダンパのアクチュエート (→ Air Input Damper) と、空気排出側ダンパのアクチュエート (→ Air Output Damper) が、空調機プラントの設備にアクセスする手続き処理として定義される。

また、このゲートのセットアップを開始するためには、空調機起動コマンドの実際の受信が前提となるが、起動コマンド待ち状態 (Waiting Start Command) からの遷移条件を、起動コマンドの受信 (Start Command Received) と定義して、空調機プラントの設備への手続き処理と関係付ける。

SFC は以上のように順序制御を表すために記法をもっているが、これに対して、センスした入力情報で、複数の状態のうち一つの状態に遷移する状況がより実用的である。これに対処するために SFC のいくつかの拡張が試みられている (たとえば、〔Niibori 92〕, 〔Kozima 92〕, 〔Mizutani 92〕 など)。

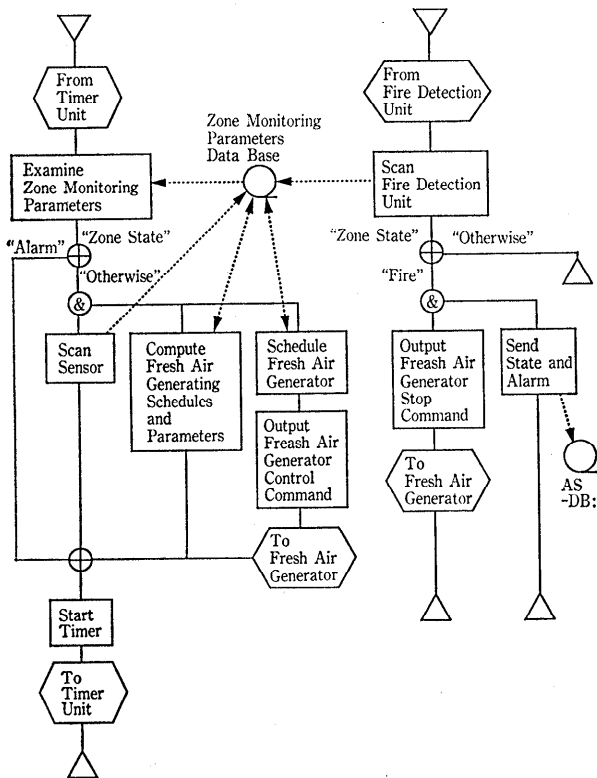


図-12 R-Net

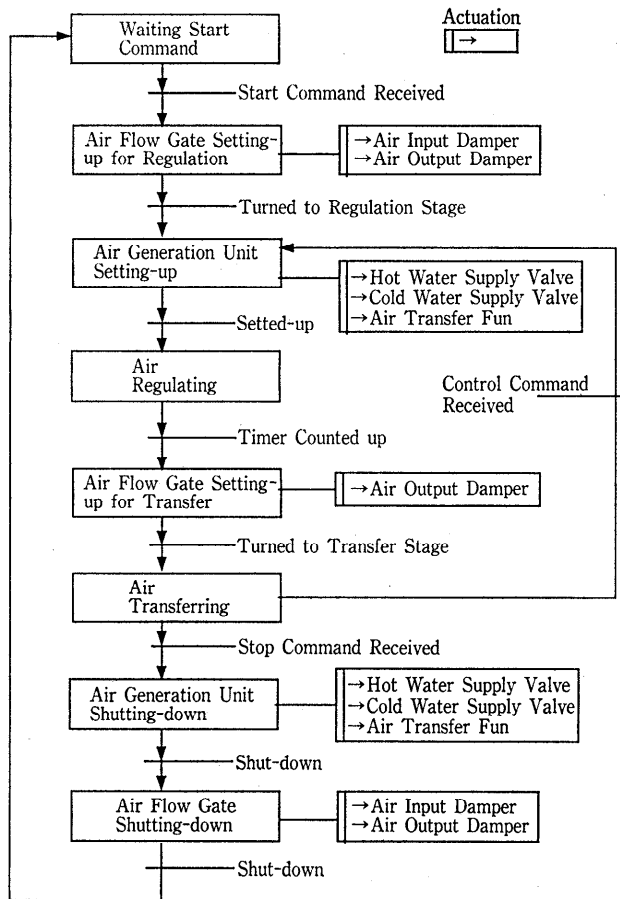


図-13 Sequential function chart—Fresh Air Generating Equipment Control

(b) ジャクソン構造図 (Jackson structure diagram)

SFCの拡張を意識したものではなく、通常用いられているものが、ジャクソン構造図([Jackson 75])などのプログラム図である。これは、イベント駆動の手続きの集合として制御プログラムを表現する。

図-14のジャクソン図では、空調ゾーン (Air Conditioning Zone) を対象プラントとして、各ゾーンの環境を快適に維持するための空調制御と火災の検出を行うゾーン監視装置 (Air Conditioning Zone Controller) の制御プログラムを設計する。

制御プログラムは、初期設定 (Initiate Zone Control System's State) とイベント監視ループ (Monitor Zone Control System's State) に分かれる。Run Air Conditioning Zone Control System

が起動されると、初期設定の実行の後、イベント監視ループが呼び出され、なんらかのイベントが発生するまで監視ループ内で割込み待ちの状態になる。

イベントが発生した場合、監視ループで、イベントの種別を判別 (図では、IF Token == ?? の?? に対応) し、イベントの種別ごとに定義された処理 (図では、IF Token == ?? THEN pp のpp に対応) を実行する。イベント処理の実行が終わると、再び割込み待ちの状態になる。

以上のように、ゾーン監視装置の制御プログラムは、イベント駆動の手続きの集合として設計される。

ジャクソン構造図と同様なネットとしてワーニエ図([Warnier 81]), アクション図([Martin 85]), ヨードン図([Yourdon 78]) などがある。

4.5 性能分析向きネット

(a) 待ち行列ネットワーク (queueing network: QN)

QNは、図-15に示すとおり複数のサーバ (server) が結合したネットである。このQNの上をトランザクション (transaction: Xact) が処理を受けながら流れる。図-15は、図-2の左にある中央監視装置内の状態監視タスク群のつながりと、互いの間の信号やメッセージのやりとりを表している。

図-15の右下にある Building Control Networkは、図-2の中央監視装置とゾーン監視装置群との間のネットワーク装置である。

中央の上部のハッチ部分は、左のタスクからの出力が右の三つのタスクに順次直列に伝播される場合を、通常のQN構造で表す。

QNの個々のサーバは5つのパラメータをもつ。

- $\lambda$ : 単位時間当りにサーバに到着する Xact の平均個数 (到着率: arrival rate)
- $\mu$ : 単位時間当りにサーバで処理できる Xact の平均個数 (サービス率: servicing rate)
- $t$ : 単位時間当りにサーバから出てくる Xact の平均個数 (スループット: throughput)
- $\rho$ : サーバが実際に Xact を処理する時間の割合

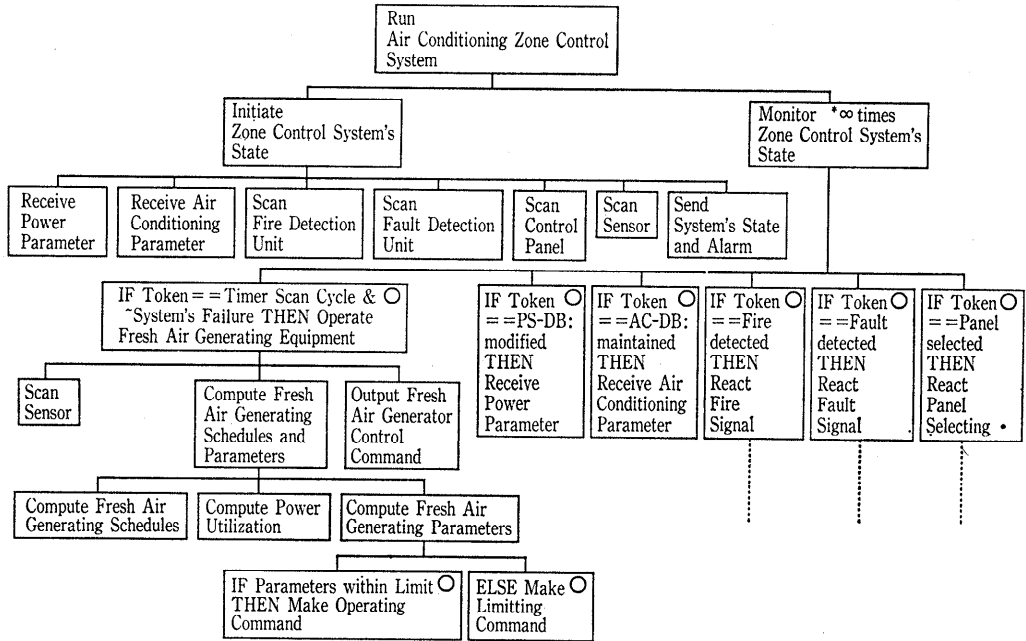


図-14 Jackson Structure Diagram

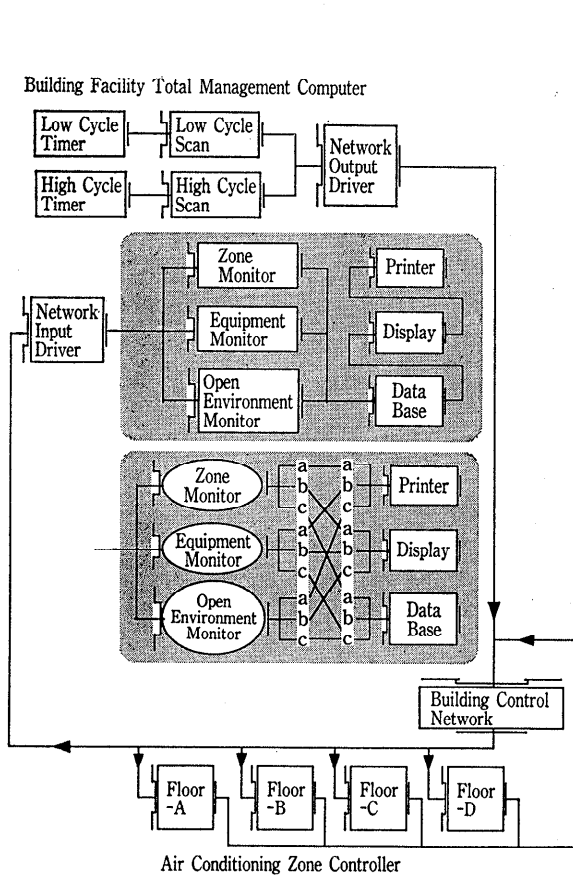


図-15 Queuing Net

(稼働率: utilization rate)

$q$ : サーバの前で処理を待つ  $X_{act}$  の平均個数  
(待ち行列長: queue length)

表-1 第1項に示すとおり,  $\lambda$  と  $\mu$  が与えられるとその大小関係により性能を表す  $\rho$ ,  $t$  が決まる。

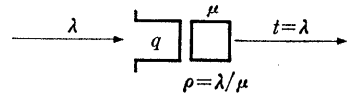
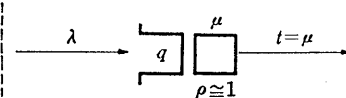
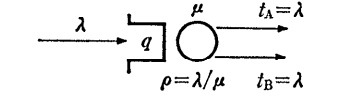
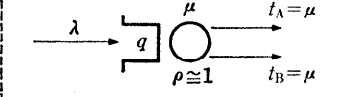
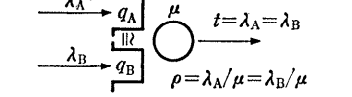

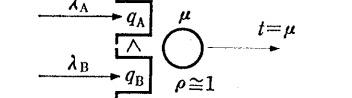
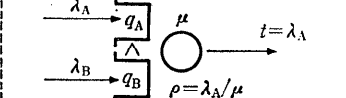
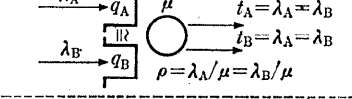
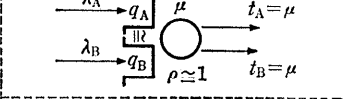
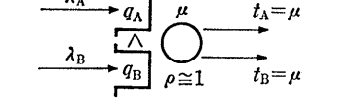
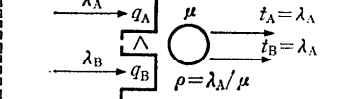
第1項の左は,  $\lambda$  が  $\mu$  より小さい場合, すなわち, 平均到着時間間隔 ( $\lambda$  の逆数) が平均サービス時間 ( $\mu$  の逆数) より大きい場合, サーバはよどみなく  $X_{act}$  を処理できる. サーバから出る  $X_{act}$  の量 ( $t$ ) は, サーバへの  $\lambda$  に等しくなる.  $\rho$  は,

$$\rho = (\text{平均サービス時間}) / (\text{平均到着時間間隔}) \\ = (1/\mu) / (1/\lambda) = \lambda/\mu \text{ となる.}$$

第1項右は,  $\lambda$  が  $\mu$  より大きい場合, サーバはよどみなく  $X_{act}$  を処理できない. すなわち, ある  $X_{act}$  に対するサービスを行っている間に次の  $X_{act}$  が到着する. サーバからの  $t$  は, サーバ  $\mu$  に等しい. サーバが常に働いているので,  $\rho \approx 1$  となる。

これらのQNは, シミュレーションモデルとして使われる. あるいは, いわゆる順問題 (direct problem)—QNでは, 外部からの  $\lambda$  や内部の  $\mu$  を与えて,  $\rho$ ,  $t$ ,  $q$  を求める問題—

表-1 Four Types of Servers and Their Performance Parameters

<p>1) normal server 1-a) case of <math>\lambda &lt; \mu</math></p> 	<p>1-b) case of <math>\lambda \geq \mu</math></p> 
<p>2) split server 2-a) case of <math>\lambda &lt; \mu</math></p> 	<p>2-b) case of <math>\lambda \geq \mu</math></p> 
<p>3) merge server 3-a) case of <math>\lambda_A = \lambda_B &lt; \mu</math></p> 	<p>3-b) case of <math>\lambda_A \geq \mu</math> and <math>\lambda_B \geq \mu</math> 3-b-1) case of <math>\lambda_A = \lambda_B \geq \mu</math></p> 
<p>3-b-2) case of <math>\mu \leq \lambda_A &lt; \lambda_B</math></p> 	<p>3-c) case of <math>\lambda_A &lt; \lambda_B</math> and <math>\lambda_A &lt; \mu</math></p> 
<p>4) match server 4-a) case of <math>\lambda_A = \lambda_B &lt; \mu</math></p> 	<p>4-b) case of <math>\lambda_A \geq \mu</math> and <math>\lambda_B \geq \mu</math> 4-b-1) case of <math>\lambda_A = \lambda_B \geq \mu</math></p> 
<p>4-b-2) case of <math>\mu \leq \lambda_A &lt; \lambda_B</math></p> 	<p>4-c) case of <math>\lambda_A &lt; \lambda_B</math> and <math>\lambda_A &lt; \mu</math></p> 

について、待ち行列理論 (queueing theory) (たとえば文献 [Kleinrock 75], [Gelenbe 80]) によって、平均値だけではなく、到着時間間隔やサービス時間にさまざまな分布を仮定し、 $\rho$ ,  $q$  などの性能パラメータの厳密解や近似解を求める解析問題として使われる。この場合、各性能パラメータの分布上の制限事項は必ずしも現実を常に正しくモデル化しているとは限らない。

伊藤・田村は、リアルタイムシステムのプロトタイプピング ([Itoh 87]) の評価フェーズで、システムを QN とみなしてその性能を評価する Trans Obj ([Itoh 92 ab]) を開発した。

一方、逆問題 (inverse problem) ([SUU 86], [SIM 90])—望ましい  $\rho$  や  $t$  を与えて、 $\lambda$  や  $\mu$  を求める問題、パラメータチューニング問題—は、現状ではあまり対象とされていないため、伊藤・沢村・志田・本

位田は、 $\rho$  が 0.7 以上のボトルネックの可能性があるサーバを改善するチューニングに定性推論を導入して効果的に逆問題を解くエキスパートシステム BDES ([Sawamura 89], [Itoh 90 ab, 91 ab]) を開発した。

(b) 同期型待ち行列ネットワーク (synchronized QN: SQN)

伊藤・志田は、SQN を提案した ([Shida 92])。これは、通常の QN の一つの入力路と一つの出力路をもつ NORMAL 型サーバに加えて、一つ以上の入力部と一つ以上の出力路をもつ 3 種類の同期型サーバから構成される。通常の QN に比べて現実のリアルタイムシステムを表現するモデルとして SQN の有効性は高い。

SQN は、4 種類のサーバが複数個結合したものである。(2)~(4) を同期型サーバと総称する。

(1) ノーマル (NORMAL) 型サーバ (表-1 項目 1)

Xact は FCFS の規則に従ってサーバに入り、サービス時間を経過した後サーバを出る。

(2) スプリット (SPLIT) 型

サーバ (表-1 項目 2)

並行的に実行可能なタスクの生成やメッセージの送信をこのサーバで表現する。

(3) マージ (MERGE) 型サーバ (表-1 項目 3)

それまで実行していた一方のタスクの消滅や、メッセージの受信をこのサーバで表現する。

(4) マッチ (MATCH) 型サーバ (表-1 項目 4)

二つのタスクが協同して仕事を行う場合、歩調を合わせるために信号のみをやりとりするタスク間同期や、メッセージのやりとりをともなうタスク間通信をこのサーバで表現する。

4 種類のサーバは、すべてサーバをトランジション、キューをプレース、Xact をトークンに置き換えると時間/確率ペトリネットグラフになるが、

QN に準拠しているため、性能の測定が容易である。

図-15 中央の下部のハッチ部分は、左のタスクからの出力が右の三つのタスクに同時分配される、実際的なタスク構造を表す SQN である。左のタスクは SPLIT サーバである。

伊藤・志田は SQN のボトルネックに対してもチューニングを行うシステム BDES-S<sup>[Itoh 92 c], [Shida 91]</sup> を開発した。

## 5. おわりに

計測制御システムの実際開発では、アーキテクチャのひな型を想定して、その上でシステムの構造と機能を具体化している。本稿では、5 種類に類別したネットをこのために用い、空調系の実用的な計測制御システムに適用した。現状のネットについて、一種類だけ用いるのは不十分であり、互いに相補的に用いるのが有効であるとの印象をもった。また、既存のネットでは、実際のリアルタイムシステム内の設備、装置などの金物のイメージが伝わりにくいという印象をもった。

## 参考文献

- [Bell 77] Bell, T. et al.: An Extendable Approach to Computer-Aided Software Requirements Engineering, IEEE trans. S.E., Vol. se-3, pp. 49-60 (1977).
- [Boussinot 91] Boussinot, F. et al.: The ESTEREL Language, Proc. IEEE, Vol. 79, No. 9, pp. 1293-1304 (1991).
- [Cooling 91] Cooling, J.E.: Software Design for Real-Time Systems, Chapman and Hall (1991).
- [DeMarco 79] DeMarco, T.: Structured Analysis and System Specification, Prentice-Hall (1979). 翻訳: 高梨, 黒田監訳: 構造化分析とシステム仕様, 近代科学社 (1986).
- [Florin 89] Florin, G. and Natkin, S.: Necessary and Sufficient Ergodicity Condition for Open Synchronized Queueing Networks, IEEE Tr. S. E., Vol. 15, No. 4, pp. 367-380 (Apr. 1989).
- [Gelenbe 80] Gelenbe, E. et al.: Analysis and Synthesis of Computer Systems, Academic Press (1980).
- [Gray 93] Gray, J. and Reuter, A.: Transaction Processing: Concepts and Techniques, Morgan Kaufmann (1993).
- [Harel 87] Harel, D.: Statecharts: A Visual Formalism for Complex Systems, Science of Computer Programming, Vol. 8, pp. 231-274 (1987).
- [Harel 88] Harel, D.: On the Visual Formalism, CACM, Vol. 31, No. 5, pp. 514-536 (1988).
- [Harel 90] Harel, D. et al.: Statemate: A Working Environment for the Development of Complex Reactive Systems, IEEE Trans. S. E., Vol. 16, No. 4 (Apr. 1990).
- [Hatley 87] Hatley, D. J. and Piribhai, I. A.: Strategies for Real-Time System Specification (Dorset House Publishing, 1987). 立田訳: リアルタイムシステムの構造化分析, 日経 BP 社 (1989).
- [Itoh 87] 伊藤, 内平, 本位田: ソフトウェア開発のためのプロトタイプングツール, 啓学出版 (1987).
- [Itoh 90 a] 伊藤, 本位田, 沢村, 志田: 定性推論と定量推論を導入した待ち行列ネットワークのボトルネック診断と改善法, 人工知能学会誌, Vol. 5, No. 1, pp. 92-105 (1990).
- [Itoh 90 b] Itoh, K., Honiden, S. et al.: Role of Qualitative and Quantitative Reasoning in Diagnosis and Improvement for Queueing Network Bottleneck, InfoJapan '90, Vol. 2, pp. 171-178 (Oct. 1990).
- [Itoh 91 a] 伊藤, 本位田: 定性推論のパラメータチューニングへの応用, 情報処理, Vol. 32, No. 2 (Feb. 1991).
- [Itoh 91 b] Itoh, K. et al.: Qualitative Reasoning-Based Parameter Tuning on Bottleneck of Synchronized Queueing Network, Compsac '91, pp. 307-314 (Sep. 1991).
- [Itoh 92 a] Itoh, K. et al.: TransObj: Software Prototyping Environment for Real-Time Transaction-Based Software System Applications, International Journal of S. E. and K. E., Vol. 2, No. 1, pp. 5-29 (Mar. 1992).
- [Itoh 92 b] Itoh, K.: Systematic Integration of Qualitative and Quantitative Parameter Tuning Methods for Improving Real-Time System Prototypes, IEEE ICSI '92, pp. 54-65 (June 1992).
- [Itoh 92 c] Itoh, K. and Konno, T.: An Integrated Method for Parameter Tuning on Synchronized Queueing Network Bottlenecks by Qualitative and Quantitative Reasoning, IEICE Trans. on Information and Systems (Sep. 1992).
- [Jackson 75] Jackson, M. A.: Principles of Program Design, Academic Press (1975). 鳥居訳: 構造的プログラム設計の原理, 日本コンピュータ協会 (1980).
- [Kishima 92] 杵嶋修三, 伊藤 潔: リアクティブシステム開発向き分析・設計環境: Asdreas, 情報処理学会ソフトウェア工学研究会, WGSE 89-11, pp. 57-64 (Dec. 1992).
- [Kleinrock 75] Kleinrock, L.: Queueing Systems, John Wiley & Sons, Inc. (1975).
- [Kozima 92] 小島 透他: 三菱総合一元化システム, 電気学会研究会, MID-92-4 (Sep. 1992).
- [Laplante 93] Laplante, P.: Real-Time Systems Design and Analysis, An Engineer's handbook, IEEE Computer Society Press (1993).
- [Martin 85] Martin, J. and McClure, C.: Diagramming Techniques for Analysts and Programmers, Prentice-Hall (1985) 国友, 渡部訳: グラム法によるソフトウェア構造化技法 (1986).

- [Mizutani 92] 水谷博之他：シーケンス制御プログラム自動生成システム，電気学会研究会，MID-92-5 (Sep. 1992).
- [Niibori 92] 新堀俊明他：制御用プラントコントローラのプログラム自動生成，電気学会研究会，MID-92-3 (Sep. 1992).
- [Otsuki 90] 大槻 繁：制御用ソフトウェアの仕様化技法，計測と制御，Vol. 20, No. 11, pp. 8-14 (Nov. 1990).
- [Peterson 84] Peterson, J. L., 市川, 小林(訳)：ペトリネット入門，共立出版 (1984).
- [Razouk 85] Razouk, R. R. and Phelps, C. V.: Performance Analysis Using Timed Petri Nets, in Protocol Specification, Testing and Verification IV (1985).
- [Sawamura 89] 沢村, 本位田, 志田, 伊藤：知識工学的手法を用いた待ち行列ネットワークのボトルネック診断，情報処理学会論文誌，Vol. 30, No. 8, pp. 990-1002 (Aug. 1989).
- [Shida 91] 志田, 伊藤：同期型待ち行列ネットワークのボトルネックに対する定性的なパラメータチューニング法，人工知能学会誌，Vol. 6, No. 6, pp. 891-903 (Nov. 1991).
- [Shlaer 92] Shlaer, S. and Mellor, S.J.: Object Lifecycles, Modeling the World in States (1992). 本位田, 伊藤監訳：続・オブジェクト指向システム分析，オブジェクトライフサイクル，啓学出版 (1992).
- [SIM 90] コンピュータ利用による逆問題解析，シミュレーション，9.1 (Mar. 1990).
- [SUU 86] 逆問題，数理科学，Vol. 274 (Apr. 1986).
- [Warnier 81] Warnier: Logical Construction of Systems, Van Nostrand Reinhold (1981).
- [Yamamoto 90] 山本修一：制御ソフトウェアを対象としたCASEの現状と動向，情報処理，Vol. 31, No. 8, pp. 1057-1067 (Aug. 1990).
- [Yourdon 72] Yourdon, E.: Design of On-Line Computer Systems, Prentice-Hall (1972). 大野, 落合監訳：オンラインシステムの設計，丸善.
- [Yourdon 78] Yourdon, E. and Constantine, L.:

Structured Design, Yourdon Press (1978) 原田, 久保訳：ソフトウェアの構造化設計，日本コンピュータ協会.

(平成5年3月9日受付)



伊藤 深 (正会員)

1951年生。1974年京都大学工学部情報工学科卒業。1979年同大学院情報工学専攻博士課程修了。京都大学工学博士。1979年より上智大学に勤務。1985年より助教授。現在、同理工学部機械工学科情報システム講座所属。ソフトウェア工学，シミュレーション手法，エキスパートシステム，定性理論の待ち行列ネットワークへの応用，三面図からのソリッドモデルの自動合成法の研究に従事。情報処理学会論文誌編集委員。著書「ソフトウェア開発のためのプロトタイプングツール」(共著)，「システムプログラム」，訳書「並行処理と Unix」(共訳)。人工知能学会，IEEE，ACM など各会員。ドメイン分析・モデリング研究グループ主査。



杵嶋 修三 (正会員)

1948年生。1968年佐世保高専機械工学科卒業。同年山武ハネウエル(株)入社，現在アドバンステクノロジーセンター主任研究員。研究テーマ：システムインテグレーション，アーキテクチャ，オペレーティングシステム，言語処理，記号処理，システム分析/モデリング，CASE，リアクティブシステム，ソフトウェア工学。ACM 会員。ドメイン分析・モデリング研究グループ幹事。