

リアクティブシステム開発向き分析・設計環境：Asdreas

杵嶋修三： 山武ハネウエル(株) アドバンステクノロジーセンター
伊藤 潔： 上智大学 理工学部 一般科学研究室 情報科学部門

リアクティブシステムの開発環境Asdreas (Architecture for the Specification and Design of REActive System)の目的は、リアクティブシステムの要求分析と基本的なタスクの設計をサポートする事である。リアクティブシステムは、プラントの計測・制御・監視を目的としたリアルタイムシステムである。Asdreasを構成する際に、現実のリアクティブシステムの問題領域とシステムにまつわる様々な要因について、十分にドメインアナリシスを行った。その結果、リアクティブシステム向きの要求分析とタスク決定の新しい方法である、三群状態遷移モデル(STD Triad)を考案した。Asdreasでは、リアクティブシステムは、制御する側のリアクティブスーパーバイザと、制御される側のリアクティブプラントから構成されるものと認識する。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。

Asdreasの要求分析・設計手法の特徴は次の3点である。

- A) プラントの挙動をシステムの分析と設計上の問題として明確に認識。
- B) リアクティブシステムの挙動を、STD Triadの枠組みに基づいて (1)自然状態群, (2)作用状態群, 及び (3)目標状態群の3つの状態群に類別されたプラントの状態遷移図(Plant STD)として分析。
- C) Plant STDを縮退する事で、制御機構として備えるべきスーパーバイザの状態遷移図(Supervisor STD)を導出。

本稿では、空調制御リアクティブシステムを事例とする。

Asdreas: Analysis and Design Environment for Reactive System Development

Shuzo Kishima
Yamatake-Honeywell, Advanced Technology Center
4-28-1, Nishi-rokugou, Ohta-ku, Tokyo 144, Japan
Tel. +81-3-3730-2301
e-mail: kishima@atc.yamatake.co.jp

Kiyoshi Itoh
Sophia University, Natural Sciences Center
7-1, Kioi-cho, Chiyoda-ku, Tokyo 102, Japan
Tel. +81-3-3238-4143
e-mail: itohkiyo@hoffman.cc.sophia.ac.jp

<Keywords> domain analysis, reactive system, state transition diagram, system development environment.

The main purpose of Asdreas (Architecture for the Specification and Design of REActive Systems) is to support system requirements analysis and primary task definition. The target environments of Asdreas are reactive systems which measure, control and monitor target plants. In constructing Asdreas, we performed the domain analysis fully on the problem domain of actual reactive systems and the features of the systems. As a result, the State Transition Diagram (STD) Triad was devised. Asdreas regards Reactive Systems as consisting of a Reactive Plant and a Reactive Supervisor. Each part reacts according to the action of the other part in real time. The following three characteristics of development methodologies are emphasized in Asdreas.

(A) In Reactive Systems, the specification and design of the total system includes the behavior of the plant.

(B) The Reactive Plant is specified in the form of a STD Triad (named Plant STD) which is categorized into three sets of states: (1) Set of Uncontrolled States (i.e., State Group by Nature Dynamics), (2) Set of Controlled States (i.e., State Group by Artificial Enforcements), and (3) Set of Decision States (i.e., State Group by Heuristic Thresholds).

(C) The essential mechanisms of the Reactive Supervisor (Supervisor STD) can be obtained by direct reduction of the Plant STD.

The process of specification and design is exemplified using an Air Conditioning Control Reactive System.

1. リアクティブシステムとAsdreasの開発アプローチ

近年、リアクティブシステムという概念に基づいたアプローチが、リアルタイムシステムとソフトウェア生産技術の研究で一つの新しい流れを形成しつつある。リアクティブシステムは、プラントの計測・制御・監視を目的としたリアルタイムシステムであり、コンピュータシステムとその制御対象となる外界(プラント、プロセス、ライン、装置等)とを、密接に相互干渉するものとして明確に認識する。

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を中心としたデータシステム(EDP, オンラインバンキング等)を対象とした。そこでは、主に、コンピュータシステム内部でのタスクの機能設計と性能評価の観点から、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、リアクティブシステムのアプローチでは、コンピュータシステム内部でのタスクの設計に加えて、計測・制御・監視の対象となるプラントを明確に認識する。そのプラントの中で関与する人間や各種装置の役割も対象とする。リアクティブシステムの制御対象とするプラントの状態は、様々な物理的特性に支配される。例えば、空調制御システムのプラントの状態は、温度・湿度・塵埃・におい等に支配される。

Asdreas (Architecture for the Specification and Design of REActive Systems) は、リアクティブシステムのための、システム要求の分析と設計の環境である。

Asdreasでは、対象プラントとその制御コンピュータシステムを全体として分析の対象とする。Asdreasは、現実の開発作業から遊離しない事を絶対的な制約条件とする。しかし、これは、現実に行われている開発作業を、単にそのままの手順でCASE的に実現するために分析しようとするものではない。Asdreasでは、複雑で大規模な現実の開発対象システムの問題領域に含まれている、対象問題自身の本来的な各種の性質の十分な分析(ドメインアナリシス)を起点とする。このドメインアナリシスの結果として得られた、対象問題自身の本来的な各種の性質を積極的に利用する。そして、現実のリアクティブシステムの要求分析とシステムの主要なタスク決定を連動して行うための新しい開発スキームを構成した。

2. リアクティブシステムに対するAsdreasのドメインアナリシス

ドメインアナリシスは、[Arango91]や[Tracz92]等の多くの文献で述べられている通り、ソフトウェアの開発における生産性の向上と再利用の促進のために必要な作業である。[Arango91]によると、"ドメインアナリシスは、問題のクラス、即ち問題領域についての知識を認識し組織化するプロセスである。これにより、個々の問題領域に固有な概念構造が得られる。"

2. 1 リアクティブシステムの構造

リアクティブシステムについては、この分野の研究者の間で、統一された概念に基づいた議論が必ずしも展開されていない。Asdreasの定義を次に述べる。

[定義] リアクティブシステム：リアクティブシステムは、制御する側のリアクティブスーパーバイザと、制御される側のリアクティブプラントからなる。<Fig.1参照> 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。

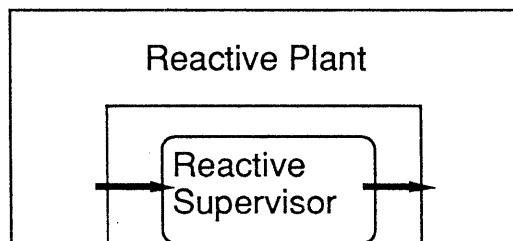


Fig.1 Reactive System's Organization

[定義] 作用・反作用関係：リアクティブシステムでは、スーパーバイザを制御対象となるプラントとの関わりとして認識する。スーパーバイザとプラントは、互いに、相手の作用(Action)に応じて必要な反作用(Reaction)を行う事ができる。この関係を作用・反作用関係と呼ぶ。

スーパーバイザがプラントを制御する場合、その機能はプラントに対して作用的(Active)に動作すると見る。この際に、プラントはスーパーバイザからのアクションに応じて反作用的(Reactive)に動作すると見る。例えば、冷却機構は、スーパーバイザのアクションである。プラントは、このアクションに反応して温度が下降する。

一方、プラントがその性質や状態に応じて何らかの自律的な挙動を行う時、プラントは、スーパーバイザに対して作用的(Active)に動作すると見る。この際に、スーパーバイザはプラントからのアクションに応じて反作用的(Reactive)に動作すると見る。例えば、プラントの自然温度下降は、プラントのアクションである。

2. 2 リアクティブシステムのドメインの性質

リアクティブシステムのプラントとスーパーバイザの挙動の分析し、その挙動を実現する主要なタスクを決定するためには、現実のリアクティブシステムのドメインに存在する、例えば、次の様々の性質を考慮しなければならない。

●プラントは、放置しておく、自然にその状態が変化する可能性がある。(例えば、プラントの自然な温度上昇、反応の自然停止)

●プラントは、予期された状態変化速度を越えて、急激な状態変化を起こす可能性がある。(例えば、異常な反応・発酵による急激な温度や圧力の上昇)

●プラントは、必ずしも理想的な環境に置かれて
いるわけではなく、外界から影響を受けて、状態が
少なからず変化する可能性がある。(例えば、外気から
の塵埃の流入、壁面からの熱の伝導・放射、日射)

●スーパーバイザは、あらかじめ組み込まれたプラ
ントに従って、プラントの状態を変化させる(もしくは
維持する)ために、プラントに対してアクションを行
う。

●スーパーバイザは、あらかじめ組み込まれたしき
い値で、プラントの状態を判断する。

●プラントには、スーパーバイザではセンサする事
が容易ではない情報が存在する。(例えば、ゾーン内
の居住者の感覚、その感覚の個人差、醸造工場の杜
氏が発酵プロセスの総合的な状況判断を行うために
センサする情報)

2. 3 Asdreasの分析の観点

リアクティブシステムの開発に対する、Asdreasの
主要な分析の観点は以下の3点である。

(1)リアクティブシステムにおいて、プラントの
挙動をシステムの分析と設計上の問題として明確に
認識する。このためには、現実の開発対象システム
の問題領域に含まれている、対象問題自身の本来的
な各種の性質の十分な分析(ドメインアナリシス)が
必要である。

(2)プラントとその制御コンピュータシステム(ス
ーパーバイザ)を全体として、システム構造を決定する。
このためには、プラントとスーパーバイザの間での作
用・反作用関係を認識する事が必要である。

(3)現実のリアクティブシステムの開発のために、
システムの要求分析とシステムの主要なタスクの決
定を連動して行う。このためには、(1)のドメインア
ナリシスの結果と、(2)のプラントとスーパーバイザの
間での作用・反作用関係を基礎とした新しい開発ス
キームが必要である。

Asdreasの要求分析は、プラントとスーパーバイザを
包括したシステムの挙動を捉えて、リアクティブシ
ステムを構成する主要なタスクを認識し列挙する事
である。このためには、プラントとスーパーバイザの
挙動を、各々単独に定義するのではなく、相互のイ
ンタラクションで結合されたものとして分析定義す
る必要がある。

リアクティブシステムを構成する主要なタスクの
認識と列挙は、スーパーバイザ側から見た場合、プラ
ントを目標とする状態に推移させるための強制的な
アクションの認識と列挙である。一方、プラント側
から見た場合は、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の認識
と列挙である。

3. AsdreasのSTD Triad

以上に示したリアクティブシステム内のアクショ
ンを認識し列挙するために、AsdreasではSTD Triadを
考案した。

Fig.2の通り、STD Triadではプラントとスーパ
バイザの挙動を、次の3つの状態群に類別した状態遷

移図を作成し、リアクティブシステムの挙動を分析
し主要なタスクを決定する。STD-Triadは、(1)自然状
態群(Set of Uncontrolled States; State Group by Nature
Dynamics)、(2)作用状態群(Set of Controlled States;
State Group by Artificial Enforcement)、(3)目標状態群
(Set of Decision States; State Group by Heuristic
Threshold)から成る。

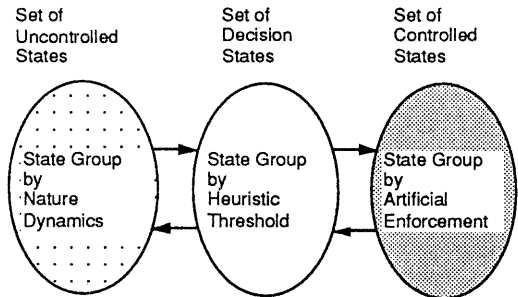


Fig.2 STD-Triad for Reactive Systems

<自然状態群>

プラントが自然に変化する事がリアクティブシ
ステムの必要条件である。即ち、プラントの
Dynamicsを明確に認識する事が必要である。これを
表すのが自然状態群である。例えば、プラントの温
度の自然上昇等はこの状態群に属する。

<作用状態群>

プラント内に存在する Nature Dynamicsのみでプ
ラントが維持されるとすれば、プラントが達成すべ
き目標値からのずれが生じる。更に、プラント内に
関与する人の行為も目標に影響を与える。

作用状態群はプラントが達成すべき目標を維持す
るための機構が有すべき状態群である。例えば、プ
ラントの自然な温度上昇に対して働く冷却機構等は
この状態群に属する。

<目標状態群>

目標状態群は、目標値に関わるヒューリスティッ
クなしきい値で類別された状態群を表す。例えば、
プラントの目標温度に対して High Temperatureはこ
の状態群に属する。即ち、リアクティブシステムに
は、必ず、何らかのしきい値が存在するとAsdreas
では認識する。

リアクティブシステムは、プラントとスーパバイ
ザを全体で統合して構成したシステムである。

Asdreasの要求分析設計では、この統合を目標状態群
の定義から行う。これは、目標状態群が、プラント
とスーパーバイザの双方で共有され、相互にインター
フェイスを取るための唯一の状態群であるためであ
る。即ち、プラントが、その自然な性質や状態の変
化に応じて行う自律的なアクションの結果として到
達する状態群と、スーパーバイザが、プラントを次
に移行するための目標としての状態群とを、目標状態
群は同時に表す。

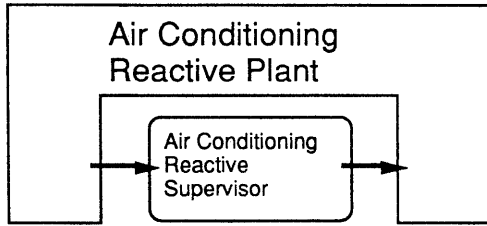


Fig.3 Reactive System's Organization in Air Conditioning

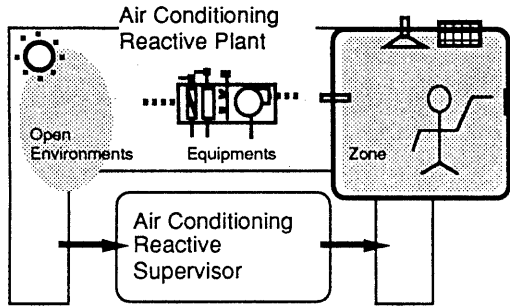


Fig.4 Reactive System's Components in Air Conditioning

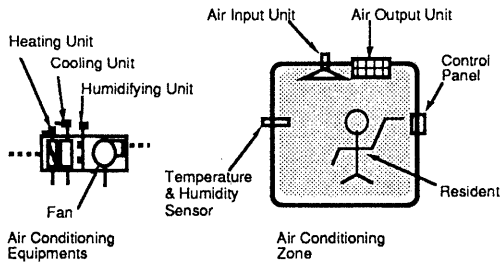


Fig.5 Explanation of Components in Figure 4

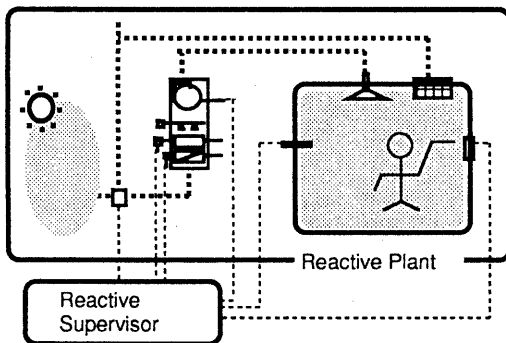


Fig.6 System Chart for Inter-Relationship between Reactive System's Components in Air Conditioning

4. 空調制御リアクティブシステムのシステムチャート

Fig.3は、空調制御リアクティブシステムのプラントとスーパーバイザを示す。スーパーバイザは、プラントの状態をセンサからの入力に基づいて同定し、プラントの状態が一定の目標になるように必要な制御出力を行う。

Fig.4の通り、空調制御リアクティブシステムの制御対象とするプラントは、(1)外界、(2)空調装置、(3)空調ゾーンの3つの基本要素から構成される。通常、空調制御のリアクティブプラントは複数のゾーンを有する。

Fig.5の通り、空調装置は(a)Heating Unit, (b)Cooling Unit, (c)Humidifying Unit, (d)Fanから構成される。また、ゾーンは、(a)Temperature & Humidity Sensor, (b)Air Input Unit, (c)Air Output Unit, (d)Control Panel, (e)Residentから構成される。

Fig.6は、システムチャートである。スーパーバイザは、ゾーンの状態を温湿度センサからの入力に基づいて同定し、常に快適な状態になるようにゾーンのAirを空調機を介して循環させて制御する。

空調機には冷房・暖房の運転モードがあり、これらの切り替えはスーパーバイザの指示により行われる。

ゾーン内の居住者はプラントの中のアクティブな主体であり、スーパーバイザとは独立した個人的な感覚に基づいてゾーン環境が快適か否かを判別し、独自に冷暖房の切り替えをスーパーバイザに指示する。

ゾーン内の状態は、ゾーン内で発生する負荷(ex. 人の数、人の活動量、人の着衣量、照明)と、外界の状態(ex. 外気温度、日射量、方位)に依存して常時変動する。

5. 空調制御リアクティブシステムの分析と設計

Asdreasの要求分析とシステム設計は、システムチャートに対してプラントとスーパーバイザの挙動を詳しく分析し、システム全体の挙動として主要なタスクを決定する。その後、プラントの挙動を制御するスーパーバイザの制御機構を導出する。

5.1 要求分析とタスク決定の基本的な戦略

Asdreasでは、以下のステップにより要求分析とシステム設計のプロセスを行う。

(1)システムチャートに対して、システムの挙動をSTD Triadを使って、Plant STDとして詳細に同定する。

(2)Plant STDを縮退して、プラントの挙動を制御するスーパーバイザの制御機構となるSupervisor STDを導出する。

5.2 Plant STD とその構成手順

Asdreasで使用するPlant STDは、Fig.7の通り、以下の点で通常のSTDと異なった性質を持つ。

A)Plant STDは、STD Triadの枠組みに基づいて構成される。STD Triadの中の全ての状態は、(1)作用状

態群, (2)自然状態群, (3)目標状態群の三つのいずれかに類別される。

B)作用状態群と自然状態群の間での状態の遷移は, 目標状態群を経由して行われる。

作用状態群は, スーパーバイザの指示でプラントの挙動が強制的に制御を受けている事を示す。

自然状態群は, スーパーバイザの指示とは独立してゾーン内で発生する負荷や外界の状態の変化により, プラントが自然に変動する事を表す。

目標状態群は, スーパーバイザ又はプラント内の居住者が, 感覚的な基準に基づいて判断したプラントの状態を表す。

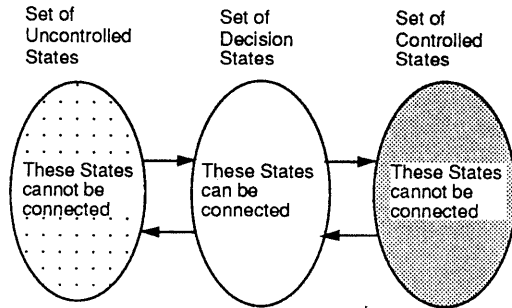


Fig.7 Categories of Plant States in the form of STD-Triad

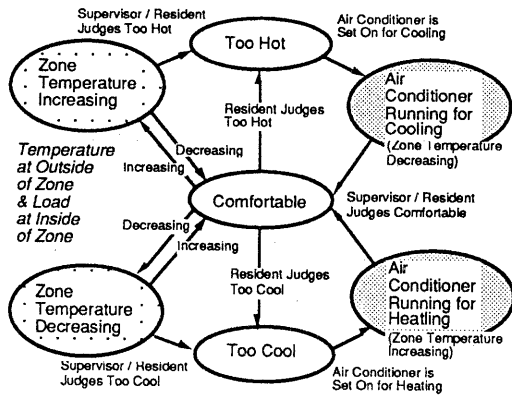


Fig.8 State Transition Diagram of Air Conditioning Reactive Plant in the form of STD-Triad

Fig.8のPlant STDは, 以下の手順で構成される。

(1)目標状態を列挙して, 各状態に感覚的な基準に従った状態名を付ける。(Fig.8では Too Hot, Comfortable, Too Coolの3状態がある。)目標状態間の直接的な状態遷移の有無を分析してSTD内にアークを記入し, アークに対するアクションを記述する。多くの場合, このアクションはプラント内に存在する人の行為である。(Fig.8ではResident Judges Too Hot, Resident Judges Too Coolの2つのアクションがある。)

(1)で作ったアーク毎に, 以下の(2)~(5)の手順を繰り返す。

(2)アークで結合された2つの目標状態に着目し, その左側に, この2つの目標状態間の状態遷移に関連して, プラントの自然な挙動を認識し, 自然状態群として定義する。状態には, プラントの自然な状態変化を表す名称を付ける。(Fig.8では, Zone Temperature Increasing, Zone Temperature Decreasingの2つの状態がある。)

(3)自然状態と, 2つの目標状態間の状態遷移を分析してアークを記入し, アークに対するアクションを記述する。このアクションは人とスーパーバイザの行為, 及びプラントの自然な挙動である。

(4)アークで結合された2つの目標状態に着目し, その右側に, この2つの目標状態間の状態遷移に関連して, 制御として働く強制作用を認識し, 作用状態として定義する。状態にはプラントの強制的な状態変化を表す名称をつける。(Fig.8では, Air Conditioner Running for Cooling, Air Conditioner Running for Heatingの2つの状態がある。)

(5)作用状態と, 2つの目標状態間の状態遷移を分析してアークを記入し, アークに対するアクションを記述する。このアクションも人とスーパーバイザの行為である。

以上の通り構成されたPlant STDで表現されるアクションは, コンピュータシステムのソフトウェア(プログラムやデータとして保持), 及び, プラント自体の機構(センサとアクチュエータを含むプラントの装置)として実現される。

5.3 Plant STDのトポロジカルな性質

Plant STDは, 次のトポロジカルな性質を持つ。

(a)作用状態同士をアークで直接連結できない。即ち, 作用状態間での直接的な状態遷移は存在しない。

(b)自然状態同士をアークで直接連結できない。即ち, 自然状態間での直接的な状態遷移は存在しない。

(c)作用状態群と自然状態群とをアークで直接連結できない。即ち, 作用状態群と自然状態群の間での直接的な状態遷移は存在せず, 必ず, 目標状態群を介した間接的な状態遷移が行われる。

(d)2つの目標状態をアークで直接連結できる。即ち, 目標状態間では直接的な状態遷移が存在する可能性がある。

STD Triadの右側の作用状態群は, スーパーバイザの強制的なアクションを示し, その際, 空調プラントは, リアクティブに反応して指示された状態に向かった強制的な推移を行う。

STD Triadの左側の作用状態群は, 空調プラントの性質や状態の自然な変化に対する自律的なアクションを示す。

STD Triadの中央の目標状態群をSTD Triadの右側の作用状態群から見た場合, 各目標状態は, スーパーバイザが空調プラントを次に移行させたい目標となる状態である。

STD Triadの中央の目標状態群をSTD Triadの左側の自然状態群から見た場合、各目標状態は、空調プラントが、その自然な性質や状態の変化に応じて行った自律的なアクションの結果として、到達する状態を示す。

リアクティブプラントとリアクティブスーパーバイザが全体で統合されたリアクティブシステムを構成するためには、STD Triadの左右から見た目標状態群はユニークな同一の状態群でなければならない。即ち、左右、各々の状態群からの遷移は、必ず、中央の目標状態群を経由して相互のインタラクションを形成する。

これは、上記の(a), (b), (c)の性質を満たすようにSTDを構成するための原理として働く。

目標状態群は、プラントの連続的な状態変化を幾つかのしきい値で定性的に類別した状態群である。この目標状態間の状態遷移は、間接的には作用状態群と自然状態群を介して行われる。プラント内で、人等の主体が自身で判断した場合に、直接的な状態遷移が行われる。

これは、上記の(d)の性質を満たすようにSTDを構成するための原理として働く。

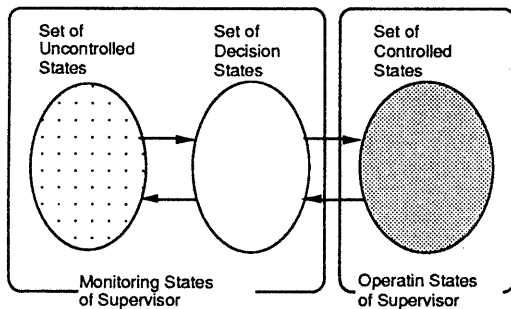


Fig.9 Segmentation of Plant STD (How to determinate Supervisor STD)

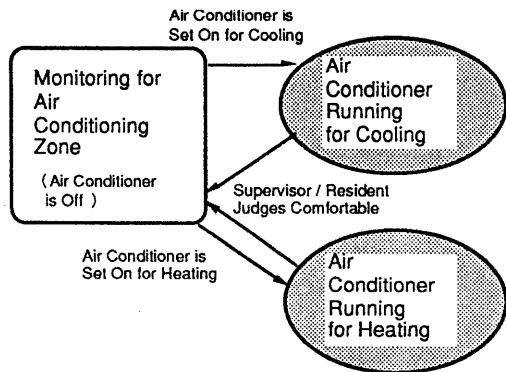


Fig.10 State Transition of Air Conditioning Reactive Supervisor

5.4 Supervisor STD とその構成手順

Supervisor STDの状態は、(1)監視状態(Monitoring States)と、(2)運転状態(Operating States)に類別される。

Fig.9は、Fig.7のPlant STDからスーパーバイザの制御機構となるSupervisor STDを決定するための縮退法を示す。

監視状態は、プラントの状態を同定するために働くスーパーバイザの制御機構を表す。即ち、監視状態は運転状態に対するトリガの役割をもつ。

プラントの状態を同定するために働くスーパーバイザの制御機構は、状態監視タスクの設計の問題として認識される。

一方、運転状態は監視状態からのトリガを受けて現在のプラントの状態を目標とするプラント状態に誘導するために働くスーパーバイザの制御機構を表す。現在のプラントの状態が目標とするプラント状態に到達した場合は、再度、監視状態に移行する。

プラントの状態を目標とするプラント状態に誘導するために働くスーパーバイザの制御機構は、状態制御タスクの設計の問題として認識される。

Supervisor STDは、以下の手順で構成する。

(1)スーパーバイザの運転状態は、プラントの作用状態群に対応させる事で構成される。

(2)スーパーバイザの監視状態は、Plant STDから作用状態群を取り除いた状態遷移図から構成される。

この結果、Plant STDからSupervisor STDが縮退できる。

Fig.10では、監視状態としてMonitoring for Air Conditioning Zoneの状態が記述される。ここでの監視状態は、Fig.8のプラントの挙動を表す自然状態群とプラントの状態を表す目標状態群から成る。

一方、運転状態としてAir Conditioner Running for Cooling, Air Conditioner Running for Heatingの2つの状態が記述される。この運転状態は、Fig.8でプラントに作用する制御を表す作用状態群から成る。

監視状態から運転状態に対するトリガとしてはAir Conditioner is Set On for Cooling, Air Conditioner is Set On for Heatingの2つのアクションが各々のアークに記される。

プラントが目標とする状態に到達した場合には、運転状態から監視状態への移行が行われる。Supervisor / Resident Judges Comfortableのアクションがアークに記される。

6. 他研究機関の方法論や研究との違い

Real-Time Structured Analysis (Real-Time SA)

[Hatley87, Ward85], Object-Oriented Analysis and Design (OOA/OOD) [Abbott83, Booch82, Booch86, Booch90, Coad90, Coad91, Shlaer88, Meyer88], ADISSA [Harel88], TransObj [Itoh92], Statechart [Harel87, Harel88, Harel90], 及び ESTEREL [Boussinol91]との違いを、(1)理論的な基礎、(2)グラフィックス記法、(3)モデル化のプロセスの観点から述べる。最後の2つのStatechartとESTERELは、Asdreasと同じくリアクティブシステムの開発のために作られたものである。他は、通常のリアルタイムシステム用である。この

中でStatechartは、道具としてSTDを用いる事も同じなので、以下の全般的な比較の後に更に詳しく比較する。

<理論的な基礎>

Real-Time SAの理論的な基礎は、データフローモデルと有限状態機械(FSM)である。OOA/OODは、意味データモデル、FSM、メッセージ送受に、ADISSAはFSMに、TransObjはトランザクション指向のシミュレーションの概念と記法に、StatechartはFSMに、ESTERELはリアクティブシステムの同期性に基づく。Asdreasは、リアクティブシステムに対するオリジナルな概念であるSTD Triadに基づく。

<グラフィックス記法>

Real-Time SAとOOA/OODは、グラフィックス記法として、データフロー図(DFD)、状態遷移図(STD)、及びエンティティ・リレーションシップ図(ERD)を用いる。ADISSAは、DFD、STDとトランザクショングラフを、TransObjは、テキスト言語に対応するグラフィックス記法を、Statechartは拡張されたSTDを、各々用いる。ESTERELは、グラフィックス記法でなくテキスト言語を用いる。Asdreasは、3つの状態群に類別されたSTD記法としてSTD Triadを用いる。

<モデル化のプロセス>

Real-time SAでは、まず、対象システムを刺激と応答でモデル化する。次に、データフローに基づいて機能分割を行う。

OOA/OODでは、まず、オブジェクトを列挙する。次に、オブジェクト内の主なデータの値の変化に基づいてSTDを定義する。更に、オブジェクトの振る舞いを詳細に表すために、その他のデータの値を用いて詳しくSTDを増強する。最後に、STDの各状態のオブジェクトの詳細な機能を定義する。結果として、OOA/OODを用いると、ソフトウェアは、メッセージを互いに送受するオブジェクトの集合体として定義される。

ADISSAは、SAのトランザクション指向の増強版である。トランザクション仕様はFSMに基づく形式仕様に変換される。

TransObjはソフトウェアの設計プロセスをトランザクション指向のパラダイムからオブジェクト指向のパラダイムへの変換ととらえる。

ESTERELは制御システムをリアクティブシステムとしてとらえる。

Asdreasは、計測と制御の専門家の経験的な知識を用い、プラントとスーパーバイザの相互のインタラクションに基づいて、システム全体としてリアクティブシステムを分析する。この分析の結果、システムの全体モデル(Plant STD)が構成され、主要なタスクが決定される。更に、このモデルを縮退する事で、スーパーバイザの制御機構(Supervisor STD)が導出される。

<StatechartとAsdreasの比較>

Statechartは、リアクティブシステムを、外界からのイベントに反応(Reaction)する、コンピュータの内部の機能集合(ソフトウェア)で構成されたシステムと認識する。即ち、リアクティブシステムの設計を、イベント処理ソフトウェアシステムの設計と考えて行う。これは、外界の挙動は、ソフトウェアシステム

の設計のために、既に与えられている事を前提とする。外界がイベントを発生するに至った元々の挙動、即ち、外界そのものの自然な挙動(ダイナミクス)については、関知しない。

Statechartでは、このイベントに反応するソフトウェアを順次詳細にブレークダウンする手順を与える。このブレークダウンのために、STDの階層化設計を行う。

機能は1つのSTDで表され、その機能を実現するためには、部品群を列挙する必要がある。各々の部品の機能を表すために、同様にSTDを用いる。このように上位のSTDの機能を実現する下位の部品のSTD群を、上位の機能を実現するSTDのグループと呼ぶ。あるグループ内の複数のSTDの状態遷移を結合するために、1対多のイベントアークを用いる。このようにして、イベント処理ソフトウェアシステムを組織化する。

Asdreasは、リアクティブシステムを、プラントとコンピュータの内部の機能集合(ソフトウェア)を包括した挙動として捉える。Asdreasでは、プラントをコンピュータに対する外界からのイベントと同時に、外界がイベントを発生するに至った元々の挙動、即ち、外界そのものの自然な挙動(ダイナミクス)を含んだものとして捉える。

このシステム認識に基づいて、プラントとコンピュータ双方のタスクを包括的に決定するために、AsdreasではSTD Triadを用いる。STD Triadでは、リアクティブシステムの状態を3つの状態群、即ち、(1)プラントのダイナミクスを表す状態群、(2)プラントが達成すべき目標を維持するための強制的な機構が有すべき状態群、(3)その目標値に関わるヒューリスティックなしきい値を表す状態群とに類別する。そして、リアクティブシステムの根幹となるプラントとコンピュータ双方のアクションを、類別された状態群の間でのSTD Triadのトポロジカルな性質で、制約されたアークで表現する。

Asdreasでは、このようにしてSTD Triadを構成して、リアクティブシステムをプラントとコンピュータ双方のタスクが統合されたシステムとして組織化する。

7. 結び

プラントの計測・制御・監視を目的としたリアクティブシステムの要求分析と基本的なタスクの設計をサポートするAsdreasの方法について詳しく述べた。Asdreasでは、要求分析とタスク決定の方法として、三群状態遷移モデル(STD Triad)を新たに考案した。STD Triadは、現実のリアクティブシステムの問題領域とシステムにまつわる様々な要因について、十分にドメインアナリシスを行って得たものである。Asdreasでは、リアクティブシステムは、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構成されるものと認識する。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。

Asdreasの要求分析・設計手法の特徴は次の3点である。

A) プラントの挙動をシステムの分析と設計上の問

題として明確に認識。

B)リアクティブシステムの挙動を、STD Triadの枠組みに基づいて(1)自然状態群、(2)作用状態群、及び(3)目標状態群の3つの状態群に類別されたプラントの状態遷移図(Plant STD)として分析。

C)Plant STDを縮退する事で、制御機構として備えるべきスーパーバイザの状態遷移図(Supervisor STD)を導出。

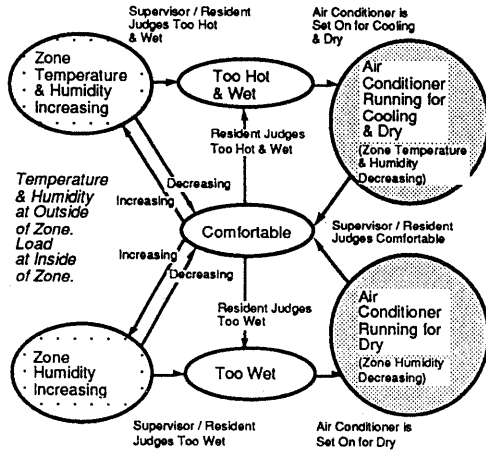


Fig.11 Plant STD of Air Conditioning Reactive Plant in the case of Temperature and Humidity Combination Control

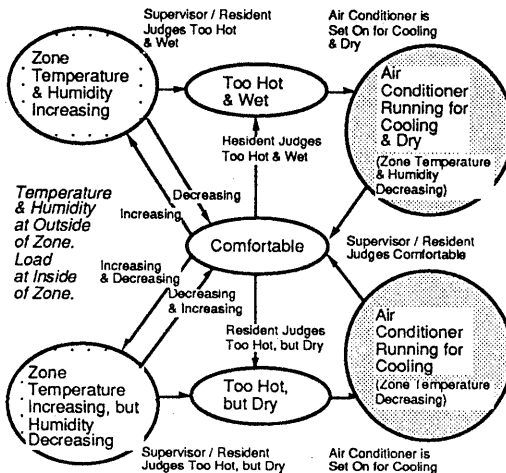


Fig.12 Plant STD of Air Conditioning Reactive Plant in the case of Temperature and Humidity Combination Control (con't)

本稿では、空調制御リアクティブシステムの開発を事例とした。これに、湿度制御、塵埃制御、におい制御等のその他の環境制御プラントを含める事も比較的容易である。例えば、Fig.11とFig.12は、温度制御と湿度制御を併用した空調制御プラントの基

本的なPlant STDの構造を示す。

現在、Asdreasが様々な分野でのリアクティブシステムに適用可能である事を確認するための実証作業を行っている。同時に、Asdreasに基づくリアクティブシステムの開発支援のためのCASEシステムの設計を行っている。

参考文献

- [Abbot83] Abbot, R. : Program design by informal English description, CACM, Vol. 26, No. 11 (1983).
- [Arango91] Arango, G. and Diaz, R.P. : Introduction and Overview: Domain Analysis Concepts and Research Directions, in Tutorial on Domain Analysis and Software Systems Modeling, pp.9-32, IEEE Computer Society Press, (1991).
- [Babin91] Babin, G., et al. : Specification and Design of Transactions in Information Systems: A Formal Approach, IEEE Trans. S.E., Vol. 17, No. 8, pp. 814-829 (Aug. 1991).
- [Booch82] Booch, G. : Object-oriented design, Ada letters, Vol. 1, No. 3 (1982).
- [Booch86] Booch, G. : Object-oriented development, IEEE Trans. S.E., Vol. 12, No. 2 (1986).
- [Booch90] Booch, G. : Object-oriented design with applications (1990).
- [Boussinot91] Boussinot, F., et al. : The ESTEREL Language, Proc. IEEE, Vol.79, No.9, pp.1293-1304 (1991).
- [Coad90] Coad, P., et al. : Object-oriented analysis, Yourdon Press (1990).
- [Coad91] Coad, P., et al. : Object-oriented design, Yourdon Press (1991).
- [Harel87] Harel, D. : Statecharts: A visual formalism for Complex Systems, Science of Computer Programming, Vol.8, pp.231-274 (1987).
- [Harel88] Harel, D. : On the visual formalism, CACM, Vol. 31, No. 5, pp. 514-536 (1988).
- [Harel90] Harel, D., et al. : Statemate: A working environment for the development of Complex Reactive Systems, IEEE Trans. S.E., Vol. 16, No. 4 (April 1990).
- [Hatley87] Hatley, D.J. and Pirbhai, I.A. : Strategies for real-time system specification (Dorset House Publishing, 1987).
- [Itoh92] 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, (March 1992).
- [Meyer88] Meyer, B. : Object-oriented software construction (Prentice-Hall, 1988).
- [Shlaer88] Shlaer, S. and Mellor, S.J. : Object oriented system analysis (Prentice-Hall, 1988).
- [Tracz92] Tracz, W. : Domain Analysis Working Group Report - First International Workshop on Software Reusability, ACM SIGSOFT Software Engineering Notes, Vol.17, No.3, pp.27-34, (July 1992).
- [Ward85] Ward, P. and Mellor, S. : Structured development for real-time systems, Vol. 1-3 (Prentice-Hall, 1985).