

TSC21 技術資料

1. 全体概要

TSC21 (*Tool of Solution & Communication for BEMS*) は、真の意味でのマルチベンダー環境を実現した 21 世紀のビルのエネルギー & 環境マネージメントを拓くソフトウェアです。

発端は 1991 年に東京電力と日建設計が蓄熱システムの制御のために開発に着手し、長年の実証試験や検証で自由度と拡張性を極限まで追求した制御プログラムであり、1999 年から有志企業によるコンソーシアムを組織し、更なる汎用化、ネットワーク化そしてオープン化を指向した通信機能とデータの命名規則の規定を付加してきました。

エネルギー&環境マネージメントのためには、的確な情報に基づき適切な運転制御・運転管理を行なう必要があります。BEMS^{*1}、BOFD^{*2}などのマネージメントシステムやコミッションングなどの研究成果が上がってきています。しかし、従来の制御システムはブラックボックスであり、修正が困難などの問題がありました。

TSC21 では、そのソフトウェア構造をアプリケーション層、通信層、機器層の三階層に分けることで物理的な制約から開放させました。これによりフィールドデータは、理論上世界中の何処からでも活用できるようになっています。しかし従来のようにフィールドデータは単に現在値だけを扱うというのでは、的確で適切な情報とは言えません。過去の履歴データや未来の予測データ、また、運転モードとインターロックされたデータのようにリアルタイムに加工されたデータが必要になってきます。ビルオーナーやビル管理者、設計者、施工者などのビルの建設から運用に関わる様々な人々が自由にフィールドデータを経営情報にリンクし活用できるための環境が求められています。

TSC21 ではこのデータに意味の理解できるコードを付けることで、表計算の式を作る感覚で誰でもが簡単に高度アプリケーションを作れるようにしました。

2. TSC21 により実現する / 可能なソリューション

エネルギーは使い方次第で効率性や経済性が大きく変動するという特性をもっています。そして益々複雑になってくる料金体系の元でのエネルギー & 環境マネージメントの実現は、使用量やデマンドなどの予測をリアルタイムに行い最適な運用を行ないたいというニーズを増大させることが考えられます。

(1) ビルの統合 BEMS ソリューション

熱源・空調・電気・搬送・照明等のサブシステム毎に TSC21 を、イーサネット上に TSC21 をインストールした DB サーバーを設置することで統合 BEMS が構成できます。Web 状態監視画面作成ツールで簡単に監視制御用インターフェースを簡単に作成できます。

(2) 遠隔監視ソリューション

WAN 或いはインターネットを活用した遠隔監視・制御、エネルギー管理などが簡単に構成できます。警報も携帯電話の e-mail 機能、状態監視も PDA でといったモバイル遠隔監

*1 BEMS: Building Energy Management System (ビル・エネルギー・マネージメント・システム)

*2 BOFD: Building Optimize and Fault Detection (最適化と不具合検出)

視(Web監視)も可能になります。

(3) ノンストップ・コントローラ・ソリューション

従来のコントローラでは、複雑化する料金体系の変更には新たなプログラムの開発が必要でしたが、TSC21 ではそのロジックの変更部分だけのカスタムファイルを変更して遠隔から配信することで、コントローラは停止することなく直ちに最新のロジックで運転されます。

3. ソフトウェアの構成

このようなソリューションの実現のためには、ビルのライフサイクルで多くの専門家との共働を必要とし、自由に様々な知識(ロジック)を組み合わせる使用することが大切になります。TSC21 のソフトウェアの構成を図2に示します。利用者は「利用者が必要なところ」のロジック記述をするだけで、複雑な制御や解析ができます。

(1) 動作環境

TSC21 のソフトウェア群のほとんどは Java により実装です。よって OS の違いを吸収し、既存の Workstation や汎用 Personal Computer はもちろん、組込機器での動作も視野に入れています。

(2) TSC/prog

TSC/prog では、カスタムファイルとそれを実行するロジックエンジンとに役割を分けることで、制御の自由度と拡張性を確保し、機能を自由に組み合わせて制御の流れを作ることで、ニーズに合わせた制御システムの構築を可能にしました。データ収集・制御・診断などに対し、自由度と汎用性を追求し、リアルタイムシステムに対して

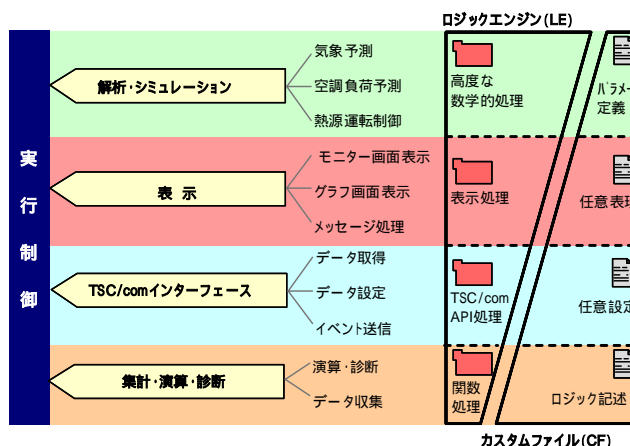


図1 ロジックエンジンとカスタムファイル

表計算ソフトのようにロジック記述だけでプログラム知識を不要としました。また、シミュレーションのような高度ソフトや定評のソフトのプラグインも簡単です。

(3) TSC/codes

従来のシステムにおいて、エネルギー管理を行う上で重要な問題があります。データが蓄積されていても、そのデータの持つ意味を理解することに多大な労力を要することです。たいていは「日本語ポイント名称」が大きな頼りとなりますが、違うポイント名称で同じ意味だったり、逆に同じポイント名称で違う意味を持っていたりということが起こっているのが現状です。日本語では冗長になってしまうポイント名称がある法則に従って記号を組み合わせることで、計測ポイントの意味するところを分かりやすく表現することができ、エネルギー管理の本来の目的である分析・改善へと円滑に進ませることが TSC/codes の目的です。

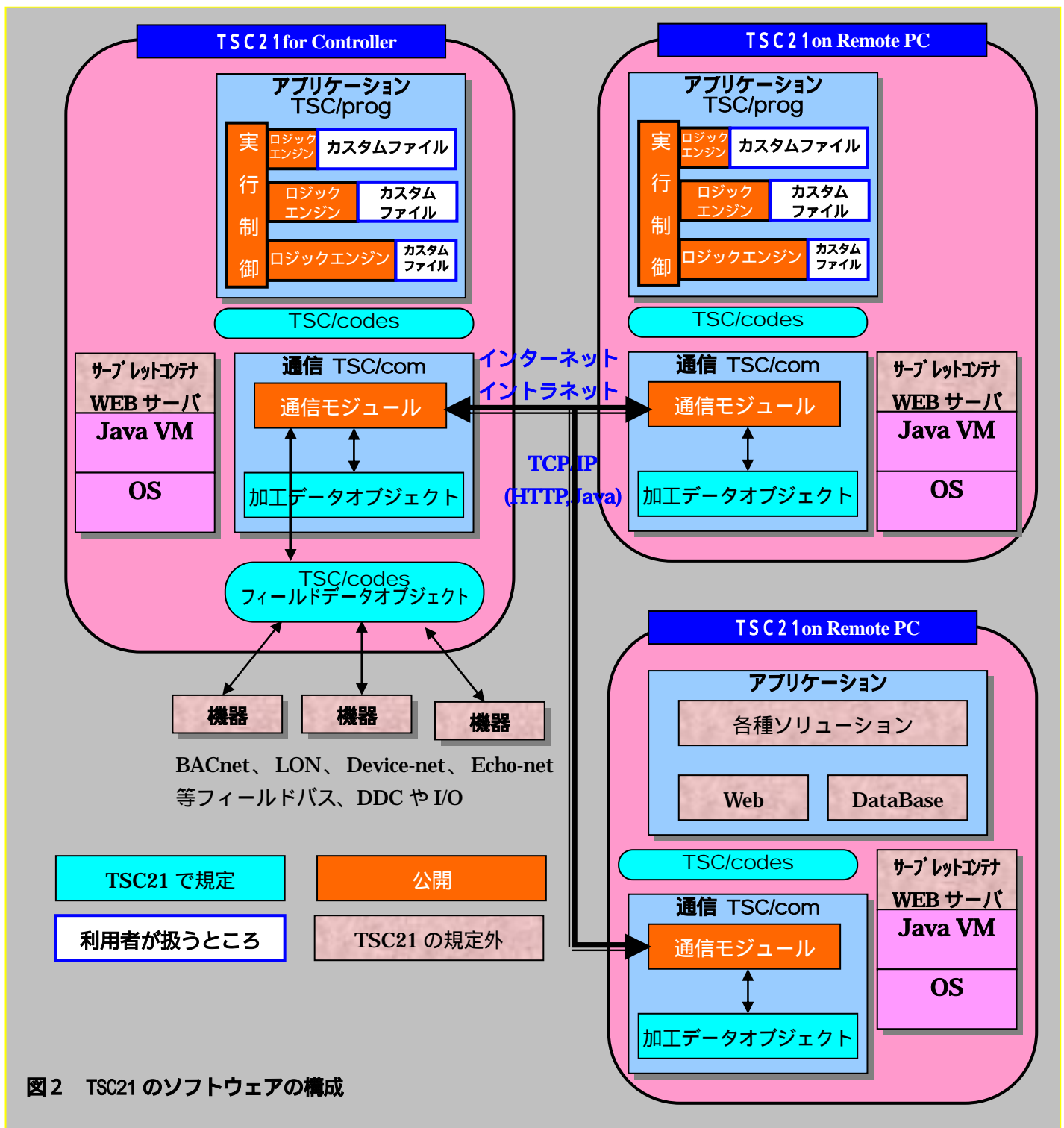


図2 TSC21 のソフトウェアの構成

(4) TSC/com

Java により実装された通信ソフトウェアです。フィールド機器の情報をデータ管理と呼ばれるデータベースにより、オブジェクトとプロパティにカプセル化し、この API を定めることで TSC/prog 等のアプリケーションからの自由なアクセスを可能にします。

4．ネットワーク構成

TSC21 は、機器層を持つコントローラと機器層を持たないリモート PC 構成が可能となり、ネットワーク構築の自由度を確保します。(図2 参照)

TSC21 for controller

蓄熱コントローラとしての必然性としての「負荷予測制御」を実現するために考えられたアーキテクチャーは、蓄熱コントローラに止まらず、空調、照明、電気、防犯、防災等様々なビル内或いは産業プロセスなどに広く使用できます。

TSC21 on Remote PC



機器層を伴わない構造はコントローラと LAN 接続することでコントローラとしての機能分担ができます。例えば予測制御のようなデータ処理部と I/O 処理部を分けることが可能に成ります。フィールド即ち手足の部分は組込型機器として頑健な仕様とし、データ処理即ち頭の部分は監視室内の PC に分担するなど用途により自由に組み合わせて使用することができます。

TSC21 on Remote PC

アプリケーションに TSC/prog を使わずに Web アプリケーションやデータベースとの連携によるアプリケーションを使うことも、TSC/codes により一元的に扱えます。大規模システムや既存のシステムとの連携も簡単に行えます。

5．おわりに

TSC21 は BEMS の今後の方向性を示す次世代のエネルギーマネジメントツールです。このような機能をより多くの場面で、様々な方々に使っていただきたいと考え、TSC21 を広く公開することと致しました。その活動母体である TSC21 推進協議会は TSC21 を利用する様々な立場の企業・団体が集まる「場」とし、TSC21 の普及、相互のレベルアップ、TSC21 の継続的な改良、新たな機能の拡充などを検討していきます。皆様のご参加をお待ちしております。

TSC21 推進協議会事務局	
	www.serl.co.jp/tsc21
	tsc21@serl.co.jp
〒166-0003 東京都杉並区高円寺南 3-47-8	
株式会社システック環境研究所内	
Tel : 03-5305-3701 Fax : 03-5305-3700	

2004.09

TSC/com 技術資料

1. 全体概要

TSC21 は、アプリケーション層、通信層、機器層の三階層に分かれています。上位のアプリケーション層のインターフェースも備えているので、高度な制御や診断や評価など様々なアプリケーションとの情報授受が自由に行えます。(図1)

機器層とアプリケーション層はネットワークで接続されてさえいれば、同一のハードウェア上でも、異なるハードウェア上でも構わなく、アプリケーション層を海外に持つていくことさえも理論的には可能です。

通信層を独立させたことで、物理的な制約が軽減され、各ベンダーはそれぞれ得意な分野に特化して

参入できるので、真の意味でのマルチベンダー化が可能となりました。また、将来、通信方式の仕様が変わったとしても、アプリケーション側に影響を与える心配がありません。

省エネルギー性の更なる追求など、今後、高度化するアプリケーションに対し、シングルベンダーで全てを対応することは困難になるでしょう。様々なアプリケーションを組み合わせる自由でシステムを構築できるマルチベンダー環境を実現することが重要であると考えます。

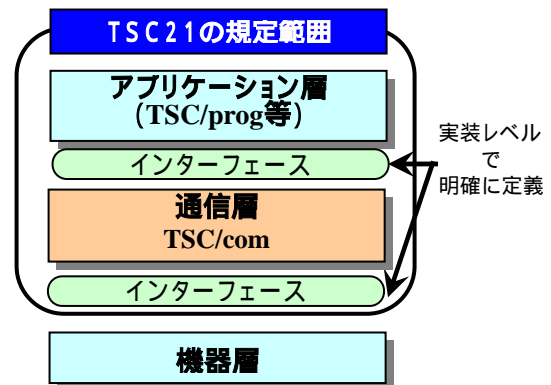


図1 TSC21の通信階層

2. TSC/com の構成・機能

図2がTSC21の構成になります。この中で通信層について実装したものがTSC/comです。TSC/comは、通信ライブラリと通信モジュールという2つの機能で構成されています。通信ライブラリは、アプリケーションからのリクエストを受け付ける機能で、Java言語用のインターフェース(API)が用意されています。

TSC21ではデータをオブジェクトとプロパティという単位で扱います。オブジェクトとプロパティは実機器に対応づけられるだけではなく、蓄積された時系列データや、データを2次利用した加工データにも対応づけられます。

通信モジュールは、通信ライブラリからの通信を受け取り、データ管理^{*1}とデータをやりとりする機能です。アプリケーションから通信ライブラリを利用する場合、通常はオブジェクトとプロパティを指定して利用しますが、それらは後述する名前命名規則であるTSC/codesを用いて指定します。

また、TSC/comにおいてアプリケーションからイベント発報機能を利用することが出来ます。発報されたイベントはTSC/comにおけるイベントサーバが起動されたマシンへと送られます。

TSC/comはJava VM上で動作します。内部通信プロトコルとしてHTTP(Hyper Text Transfer Protocol)を利用していますが、TSC/comのユーザーはそれを意識することなく利用することが出来ます。また、プロトコルにHTTPを利用しているため、FireWallの透過も容易です。TSC/comはインターネット経由のデータ授受を想定しているため、ユーザー認証によりセキュリティにも十分な配慮を行っています。

^{*1} 実機器の計測値や設定値といったフィールドデータオブジェクトを管理する機器層の機能(1次データ管理)やフィールドデータを加工したものなどの加工データオブジェクトを管理する機能(2次データ管理)

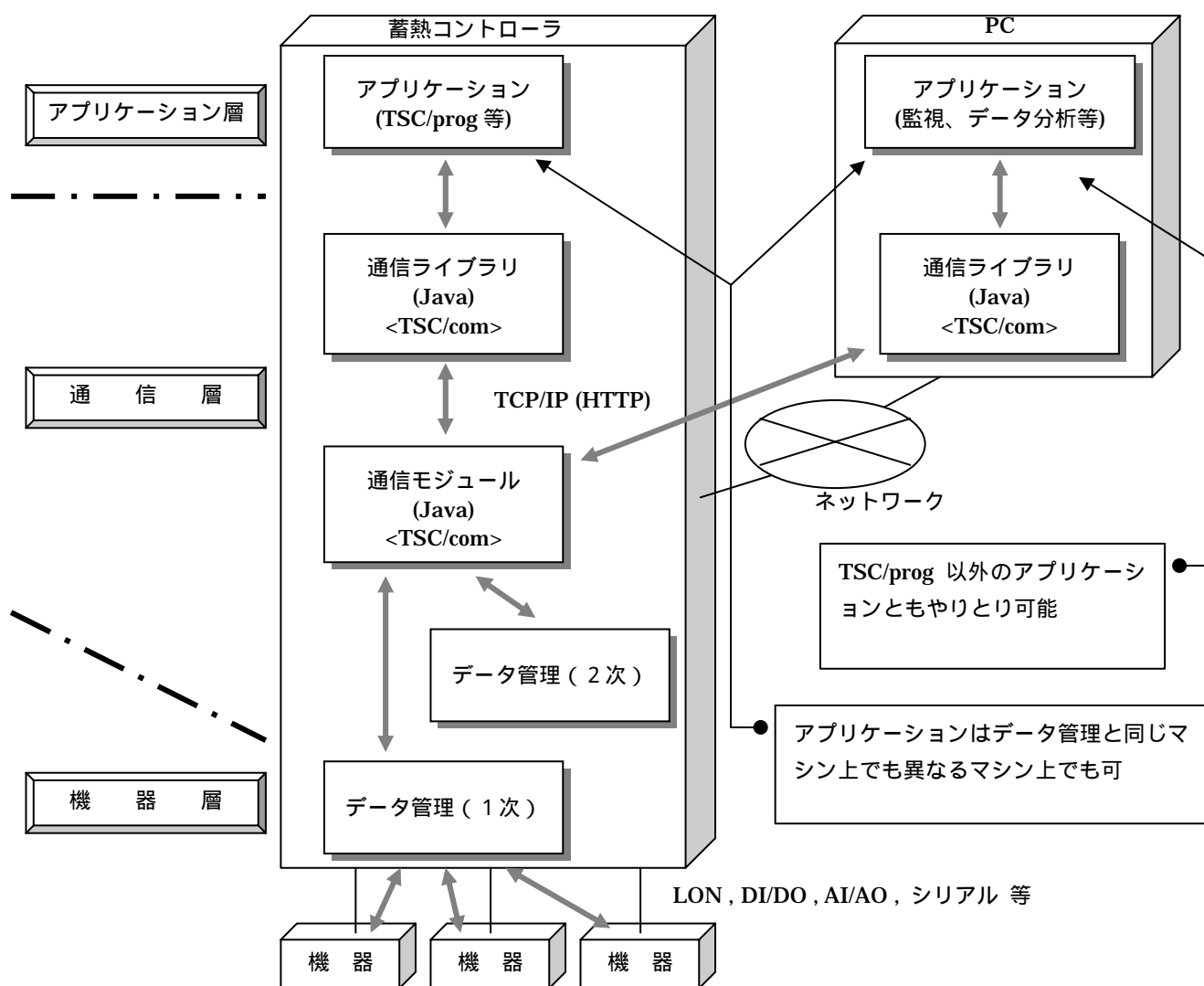


図2 TSC21 構成図

3. TSC/com の振る舞いと特徴

あるデータの現在値を読む場合を例にして、TSC/com の振る舞いを説明します。(図3)

- (1) アクセスしたいオブジェクトをあらわすグローバル名、データ管理名、オブジェクト名、プロパティ名を引数として、データ読み込み用のサービス(readProperty)を、アプリケーションから呼び出す(図3の)。
- (2) 通信ライブラリでは、グローバル名を、対応する蓄熱コントローラの IP アドレスに変換する。その後、アクセスするオブジェクトのデータ管理名、オブジェクト名、プロパティ名が該当する通信モジュールに伝えられる(図3の)。
- (3) 通信モジュールは、データ管理名に該当するデータ管理に対して、オブジェクト名、プロパティ名を指定し値を要求する(図3の)。
- (4) データ管理では、オブジェクト名、プロパティ名に対応した値を戻り値として通信モジュールに返す(図3の)。さらに、通信モジュールは通信ライブラリに(図3の)通信ライブラリはアプリケーションに値を返す(図3の)。

以上の通り、アプリケーションから見ると、全て名前による指定で処理することができます。この名前は TSC/codes に基づいてユニークに決められている論理的な名前です。

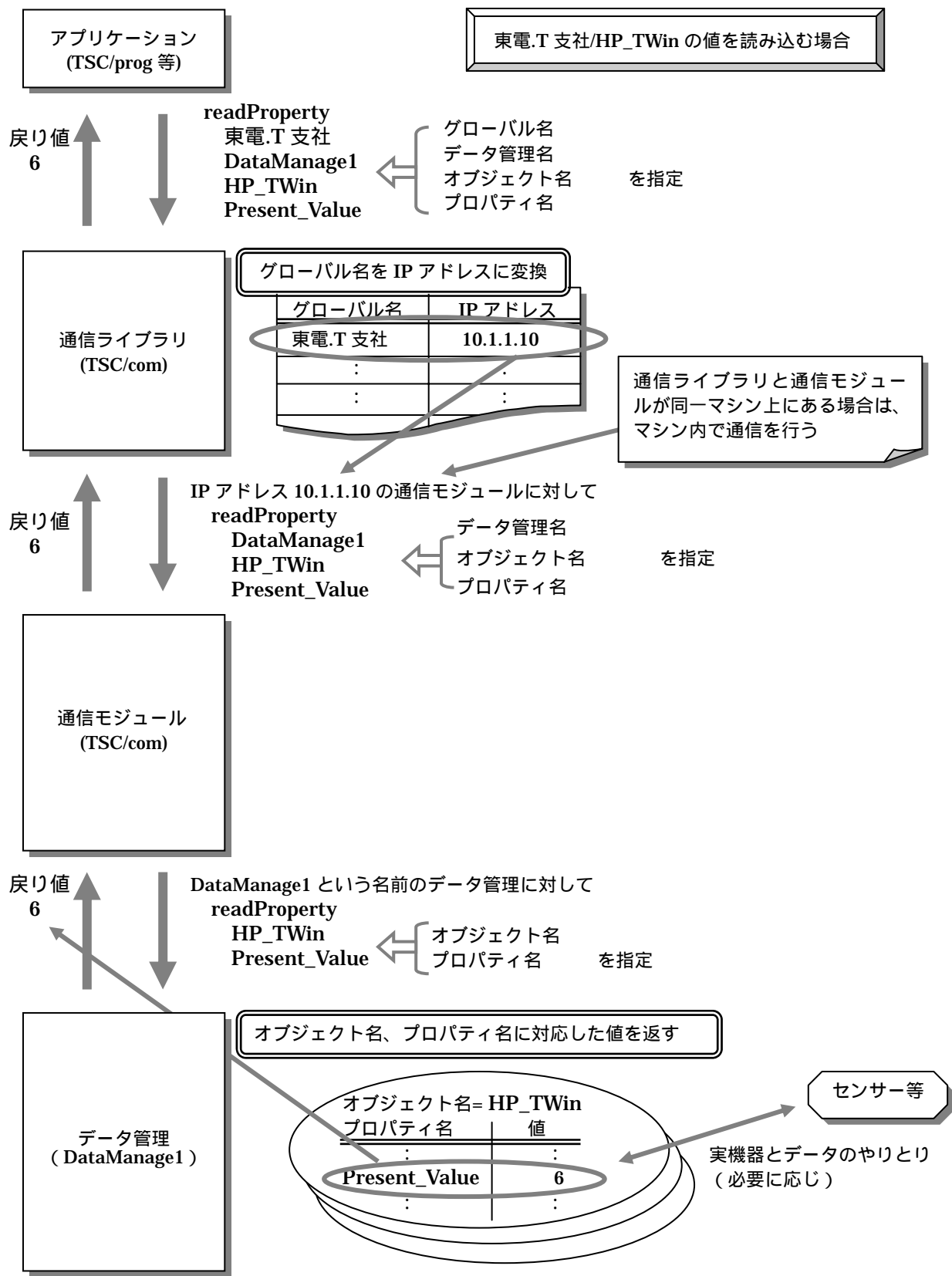


図 3 TSC/com の仕組み

4．サービスAPI

表1 サービスAPI 一覧(抜粋)	
サービス	機能
readProperty	指定した単一オブジェクトの単一プロパティの値を取得する
readPropertyMultiple	単一オブジェクトの複数プロパティや複数オブジェクトのプロパティの値を一度のサービスで取得する
readRange	指定したオブジェクト名を持つオブジェクトの時系列データを取得する
writeProperty	指定した単一オブジェクトの単一プロパティに値を書き込む
writePropertyMultiple	単一オブジェクトの複数プロパティや複数オブジェクトのプロパティの値を一度のサービスで書き込む
writeRange	指定されたローカル名を持つオブジェクトの時系列データを書き込む
whatHas	指定したデータ管理の持つ全てのオブジェクトを取得する
whatPropertyHas	指定したオブジェクトのプロパティ情報を取得する
eventNotification	イベントを受信確認要求なしで通知する

表1 がアプリケーション（TSC/prog 等）向けに提供している主なサービスAPI です。これらを用いてアプリケーションを作成可能です。（図4）

```
public static void main(String args[]){
    /* readProperty を用いデータ管理からデータを取得する */
    ObjectUnit ou = ServiceProvide.readProperty(" 東 電 . T 支 社 ", "DataManage1","HP1_ 温 度 ",
"present_value");
    System.out.println("オブジェクト名 = " + ou.getObject_name());
    for (Enumeration e = ou.getPropertyUnitList().elements(); e.hasMoreElements(); ) {
        PropertyUnit pu = (PropertyUnit) e.nextElement();
        System.out.println("   プロパティ名 = " + pu.getPropertyName());
        System.out.println("   プロパティ値 = " + pu.getPropertyValue());
    }
}
```



図4 プログラムサンプル(抜粋,エラー処理を省いています)

この他にも、データ管理用の各種APIも用意しており、ユーザーが独自のデータ管理を実装することが可能です。

5．おわりに

通信部分を独立の階層として定義しTSC/comとして実装することで、アプリケーション側も機器側も実装レベルで通信と独立となり、実装負荷をあまり上げずに、データを名前管理することができるようになります。シングルベンダーでは対応不可能な高度なアプリケーションも、このようなマルチベンダー環境の仕組みの中で実現可能となります。今後は、実証物件の状況もふまえながら、TSC/comの機能アップをさらに図っていく予定です。

TSC21 推進協議会事務局


www.serl.co.jp/tsc21

tsc21@serl.co.jp

〒166-0003 東京都杉並区高円寺南 3-47-8
 株式会社システック環境研究所内
 Tel: 03-5305-3701 Fax: 03-5305-3700

TSC/prog 技術資料

1. TSC/prog の構成

TSC/progはBEMSを構築するためのフレームワークです。従来の制御システムは、制御の中身がブラックボックスで、機能の追加・修正が困難であるため、予め決められた枠の中での制御になってしまいます。TSC21では、カスタムファイルとそれを実行するロジックエンジンとに分けることで、制御の自由度と拡張性を確保し、機能を組み合わせで制御の流れを作ること、ニーズに合わせた制御システムの構築を可能にしました。

TSC/prog は次の構成から成っています。



- ・ 機能単位毎に独立したプログラム = ロジックエンジン (以下 **LE**)
- ・ この動作を規定するテキストファイル = カスタムファイル (以下 **CF**)
- ・ 全体を統合する実行制御プログラム

図1のようにロジックエンジンはその機能により4種に分類されます。解析・シミュレーションの **LE** は、ロジックがある程度固定されているため、ユーザーは建物毎に異なる情報(所在地、規模、機器仕様など)のみを **CF** に記述することで空調負荷予測や熱源の運転制御を行うことができます。一方、集計・演算・診断の **LE** は、プログラムには固定されたロジックが無いため、**CF** に演算式や診断式を記述することが必要になります。これにより演算の中身をオープンにする事が可能になります。これらの **LE** の動作を規定しているのが実行制御プログラムです。実行制御プログラムは **LE** がどのような条件・順番で起動するのかをコントロールしており*1、この情報は実行制御 **CF** に記述します。

LE はユーザーが独自に作成する事も可能です。たとえばTSC/progではARX モデルに基づく空調負荷予測 **LE** を搭載していますが、独自の手法(例えばニューラルネットワーク手法)に変更する事もできます。つまり **LE** はC++、Javaなどの言語で作成された各々独立したプログラム*2あり、処理内容に応じて並列処理もしくは順次処理を行います。単機能の **LE** を、実行制御プログラムの指示により起動しているため、ユーザーは自由にTSC/progのロジックの組替え、カスタマイズを行うことが容易にできるのです。

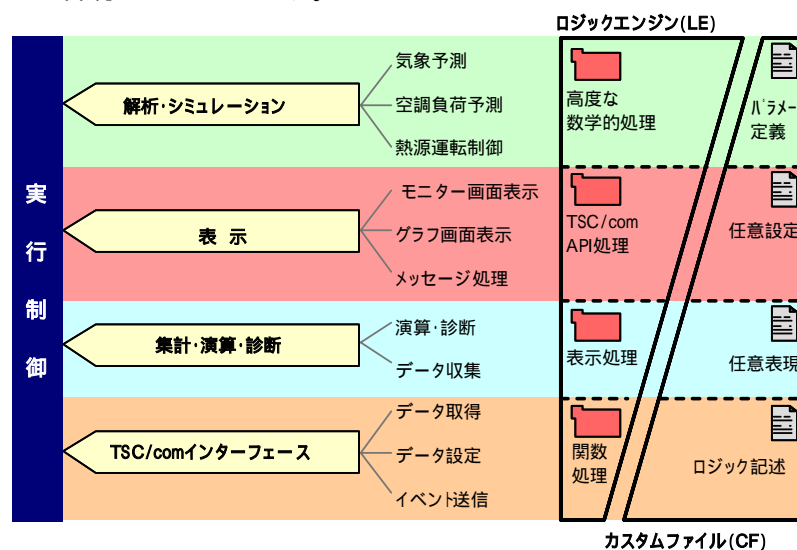


図1 TSC/progの構成

*1 unixにおけるcronやWindowsにおけるタスクスケジューラのような動きをしています。 *2 WSH等のスクリプトを呼び出すことも可能です。これにより簡易にログの回転や過去の蓄積データの削除等も行なう事ができます。

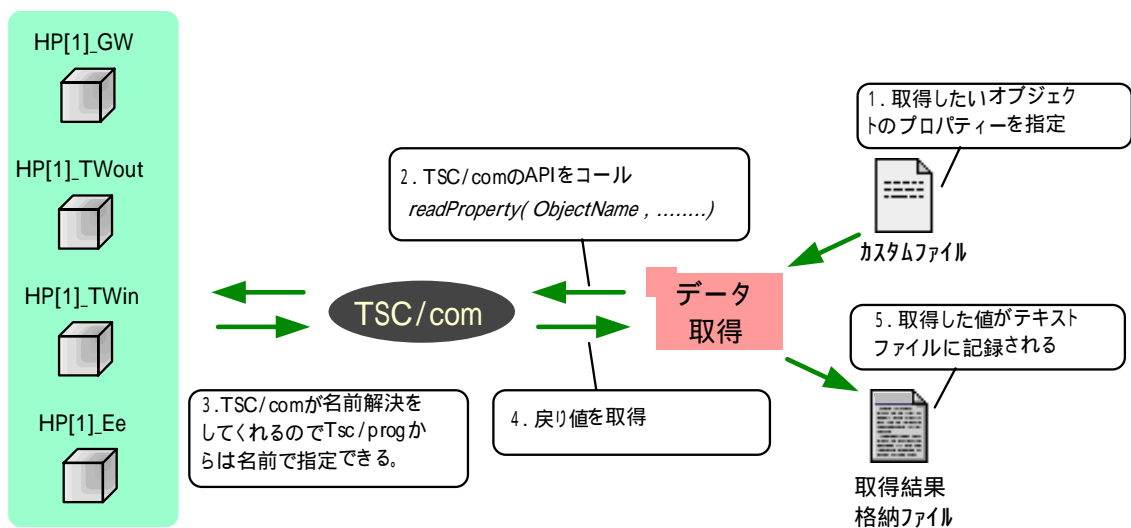


図2 オブジェクト状態取得

2. オブジェクトとの通信

TSC21では計測情報や制御指令の1つ1つがオブジェクトとなっており、オブジェクトの通信基盤としてTSC/comが用意されています。TSC/progにはTSC/comを利用してオブジェクトの値を取得するためのLEが用意されています。図2のように、オブジェクトIDの名前解決はTSC/comが受け持ってくれるため、ユーザーはオブジェクトの「名前」さえ分かれば、そのオブジェクトの持つ情報を取得する事ができます。例えば、ヒートポンプ（HP1）の出入口水温、流量、消費電力を取得する場合、表1のようにCFを作成するだけです。TSC/comのAPIはLEが内部的に呼び出してくれるため、ユーザーからはTSC/comを意識することはありません。

これにより、取得結果は表2のようなCSV形式のテキストファイルに格納されます。このファイルを利用して後述する集計・演算等を行います。また、機器発停信号など、オブジェクトへのデータ設定に関しては、データ設定用LEを用いて、設定したい値をCFに記述するだけで、取得と全く同様な手順で処理することができます。

表1 オブジェクト状態取得のためのカスタムファイル記述例

*No	変数	コメント	単位	API	オブジェクト名	プロパティ名	A 係数	B 係数
001	HP1_TWi	HP1 入口水温		P	東電 豊島/HP[1]_TWin{o}	Present_Value	1.0	0.0
002	HP1_TWo	HP1 出口水温		P	東電 豊島/HP[1]_TWout{o}	Present_Value	1.0	0.0
003	HP1_GW	HP1 流量	Lit	P	東電 豊島/HP[1]_GW{o}	Present_Value	1.0	0.0
004	HP1_Ee	HP1 消費電力	kW	P	東電 豊島/HP[1]_Ee{o}	Present_Value	1.0	0.0

【解説】変数はTSC/prog 内部で用いるローカルな名前であり、自由につけて良い。オブジェクト名はネーミングコードに準拠した正式名である。グローバルネームは省略できる。A係数、B係数は取得した値の1次式補正係数である。

表2 オブジェクト状態取得結果ファイル例

date	time	HP1_TWi	HP1_TWo	HP1_GW	HP1_Ee
				Lit	kW
20010720	130000	15.0	7.0	180.0	30.1

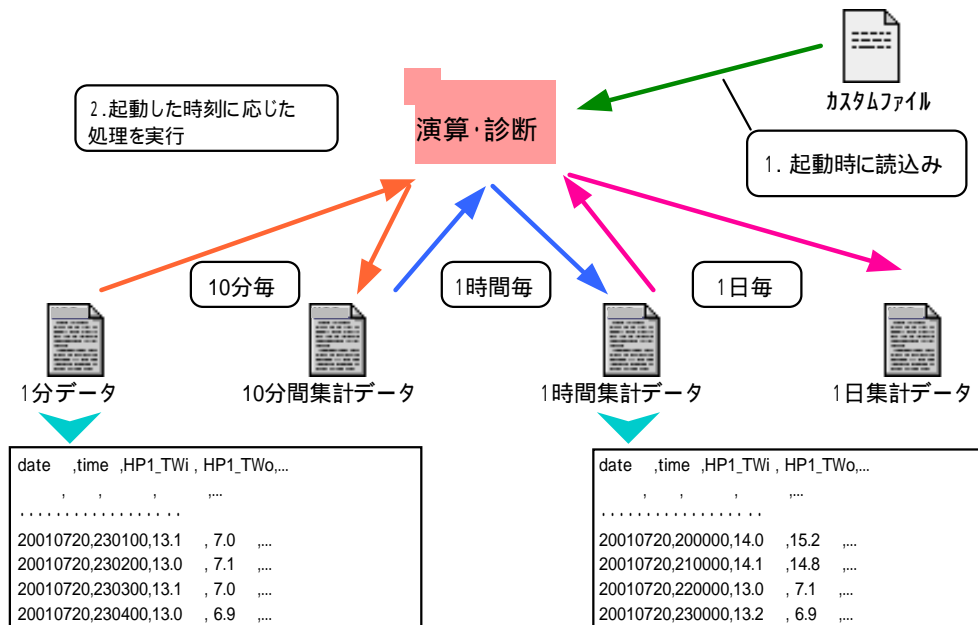


図3 データ処理の流れ

3. カスタマイズ可能なデータ処理

TSC/progの演算・診断LEでは診断したい事項や、処理ロジックをプログラムに書くのではなく、プログラムの外のカスタムファイルに置きます。こうすることで、ロジックがオープンになり、ノウハウの共有や蓄積が可能になります。演算・診断LEの中身は、+ - * /の四則演算、exp,ln,sin,cos,max,min等の数学関数、and,or,if等の論理演算関数、スケジュールを扱う配列関数、空気線関数、などの関数機能の集まりです。表3は、データ処理を行うカスタムファイルの例です。表2で得たオブジェクトの値をもとに、HP製造熱量やCOP、夜間移行率などの評価値を演算しています。演算・診断LEでは表計算ソフトのように、演算式や積算/平均、補間/補正などを定義すると、図3のように10分毎、1時間毎、...に上位のデータに集計演算を行うことができます。これを利用すれば日報や週報、月報などを思い通りに作成する事ができるのです。

表3 データ処理カスタムファイルの記述例

*No.	変数	コメント	定義式	集計法	補間	単位	出力	実行レベル
001	HP1_Ee	HP1 電力	#HP1_Ee	cal	中間	kW	1	*****
002	HP1_EEe	HP1 積算電力量	HP1_Ee/m	sum	中間	kWh	1	*****
003	HP1_EEe[ngt]	HP1 夜間電力量	HP1_Ee * fJngt	sum	中間	kWh	1	*****
004	HP1_TWi	HP1 入口水温	#HP1_TWi	ave	中間		1	*****
005	HP1_TWo	HP1 出口水温	#HP1_TWo	ave	中間		1	*****
006	HP1_GW	HP1 流量	#HP1_GW * HP1_GW * uGW	ave	中間	Lit	1	*****
007	HP1_QW	HP1 製造熱量	(HP1_TWi- HP1_TWo)*HP1_GW*uQW	ave	中間	kW	1	*****
009	HP1_fEEe[ngt]	HP1 電力量夜間移行率	@DIV(HP1_EEe[ngt], HP1_EEe)	cal	中間	%	1	-----
010	HP1_COP	HP1 COP	@DIV(HP1_QQW, HP1_EEe * 3.6)	cal	中間	-	1	-----
011	uGW	流量換算係数	1.0	nop	中間	-	0	*****
012	uQW	負荷換算係数	4.186 * 60	nop	中間	-	0	*****
014	HP1_Ee[peak]	HP1 ピーク電力	HP1_Ee	max	中間	kW	1	-----
015	Jngt	夜間時間定義	@ARY_SET([1:24] 1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,1,1)...					
016	fJngt	夜間フラグ	@ARY_GETELMT(Jngt, JH)	nop	中間	-	0	*****
017	JH	時刻	@TIME()/10000	nop	中間	-	0	*****
018	m	インターバル	10	nop	中間	-	0	*****
*End								

【解説】定義式は、#は正式なオブジェクト名、@は内部定義済み関数である。実行レベルは左から毎分、10分毎、1時間毎、1日毎、1月毎の実行フラグを表しており*は実行、-は実行しない。15行目は配列設定関数で、16行目は配列要素取得関数である。各時間帯に対応するJHは夜間なら1の値を取り、そうでない時は0を返す。sumで自動積算、aveで平均処理、calはその都度演算される。Noは空白可で、記述順序は不同であり、自由に定義して良い。

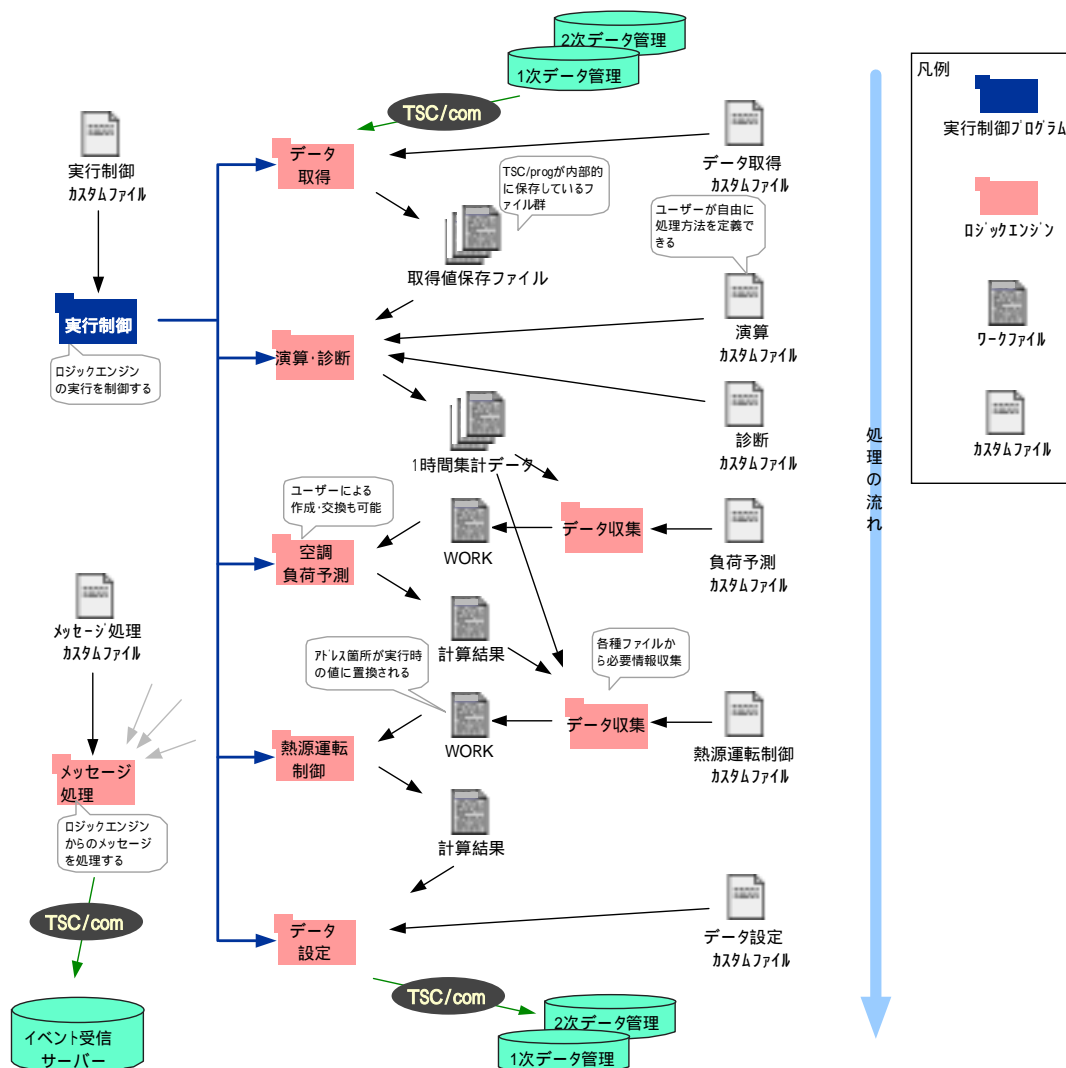


図4 蓄熱制御システムとしての実装例



4. 熱源（蓄熱）コントローラとしての実装例

TSC/prog は新世代の蓄熱コントローラとして開発がスタートした経緯から、蓄熱制御に関わる **LE** が特に充実しています。TSC/prog による制御システムの実装例として熱源（蓄熱）コントローラとしての機能を持たせた場合のファイル構成を図4に示します。この場合はTSC/progに含まれている、気象予測、空調負荷予測、有効蓄熱量に基づく熱源運転制御などの **LE** を順次起動するように設定します。建物固有の情報やパラメータは **CF** に記述します。

5. おわりに

TSC/prog は高度な制御システムソフトウェアを実現するためのフレームワークとして位置づける事ができ、カスタムファイルによる柔軟なカスタマイズ性と、ロジックエンジンによる高度な拡張性により様々なニーズへの適用が可能です。

TSC21 推進協議会事務局


www.serl.co.jp/tsc21

tsc21@serl.co.jp

〒166-0003 東京都杉並区高円寺南 3-47-8

株式会社システック環境研究所内

Tel: 03-5305-3701 Fax: 03-5305-3700

TSC/codes 技術資料

TSC/codes では、オブジェクトに関する約束事である TSC/object code とデータの命名方法に関する約束事である TSC/naming code を定めています。これにより、情報を統一的に扱うことができ、システム構築が合理化され、情報通信も容易になります。

I TSC/object code

2 種類のオブジェクト

1) 1 次データオブジェクト (フィールドデータオブジェクト)

- ・計測値、状態値、制御指令などの情報で、物理的に存在する“もの”に直接的に対応するオブジェクトをいいます。
- ・ベンダー側で用意し、TSC/com では「1 次データ管理」として扱います。

2) 2 次データオブジェクト (加工データオブジェクト)

- ・TSC/prog などの上位側のアプリケーションで作られる情報で、計算結果や診断結果、何らかの修正値・補正值、予測値など何でも構いません。TSC/com ではこれらは「2 次データ管理」として扱います。
- ・2 次データオブジェクトは、現在値 (Present Value) の他、過去の時系列データ、未来の予測データなどを 1 つのオブジェクトに包含し、これらはプロパティとして扱います。

オブジェクトに対する情報アクセス

- ・アプリケーション側からは「1 次データ」「2 次データ」を特別に意識する必要はありません。
- ・Read Property、Write Property などの“サービス (関数)”が TSC/com に標準的に用意してありますので、これを利用すれば、通信システムの知識が無くてもアクセスできます。
- ・更に、TSC/prog のカスタマイズ機能を利用すれば、これらのサービス (関数) は TSC/prog がカバーするので、プログラムの知識が無くてもアクセスできます。
- ・2 次データオブジェクトに関しては、オブジェクトの生成・修正・削除などのユーティリティ機能が TSC/com に標準的に用意されています。これを利用すれば、オブジェクト指向の高度で専門的な知識が無くてもオブジェクトの生成等ができます。

BACnet との関係

- ・BACnet は下位の計測・制御指令をターゲットにしたものです。この範囲では、TSC21 の 1 次データは BACnet の概念と概ね一致します。
- ・TSC21 の 2 次データは BACnet の範囲を超えた概念です。負荷予測などの高度な制御や、BEMS や BOFD などのマネジメントの分野では 2 次データの役割がより重要になります。

LON との関係

- ・TSC21 や BACnet ではオブジェクトの単位はデータですが、LON のオブジェクトは機器が単位であり、データはその中のプロパティです。

BACnet や LON とのインターフェース

- ・BACnet や LON とのインターフェースについては、ベンダー側で TSC 用のインターフェースを用意しなければなりません。
- ・ベンダー側のシステムがオブジェクトであれば、TSC21 のオブジェクトをインターフェースとして用意することは容易です。

なお、最も簡便な方法は TSC/com の 2 次データ管理を利用することです。この場合は、TSC21 のオブジェクトを開発する必要がありません。(2 次データと同じ扱いになります)

表 1 TSC/com で扱うオブジェクトのタイプ

		TSC21	BACnet (参考)
プロトコル		T C P / I P	BACnet Protocol
オブジェクトの識別		Object Name で識別する	Object ID (番号) で識別する
	Object ID	TSC/Naming Code による (桁数に制限はない)	4 桁の 16 進数 (LON は 6 桁)
	Object Name		決まりはない
データ形式		オブジェクト内部は文字列。 授受するデータ形式も文字列	整数、実数などのタイプがオブジェクト毎に決められている。
1 次データ オブジェクト		計測値、状態値、制御信号 などで、物理的な存在に 対応するもの。(物理オブ ジェクト)	<ul style="list-style-type: none"> • AnalogInput (計測値) • AnalogOutput (制御値) • AnalogValue (設定値等) • BinaryInput (0/1 のスイッチ) • BinaryOutput (0/1 のリレー) • BinaryValue (制御パラメータ) • TOT (通算値) • Command • Device • EventEnrollemnt • Loop • Muti-StateInput • Muti-StateOutput • NotificationClass
2 次 デ ー タ オ ブ ジ ェ ク ト	複合型データ オブジェクト	アプリケーション側が生 成する論理的なオブジェ クト 物理的に同じ意味のもの を 1 つのオブジェクトと してまとめて扱う複合型 のオブジェクト • Data Object	Present Value の他 に時系列配列の Property を持つ <ul style="list-style-type: none"> • m_Trend(1 分) • m10_Trend(10 分) • m30_Trend(30 分) • h_Trend (時間単 位) • d_Trend(日単位) • W_Trend(週単位) • M_Trend(月単位)
	等価オブジェクト	• Equivalent Object 1 つの情報に複数の名前 を付けて仮想的に別のオ ブジェクトとして扱う	TSC/com のカスタ ムファイルで ” 等価 ” を定義す る
	集合オブジェクト < 将来 >	• GroupObject 複数のオブジェクトを集 合名でまとめて扱うオブ ジェクト	TSC の方が BACnet よりも汎用性があ る。
	仕様オブジェクト < 将来 >	• Specification Object 機器仕様を扱うオブジェ クト	• GroupObject (集合)
			< 該当するものはない >

II TSC/naming code

TSC では、計測・計量データを「どのような目的で使うのか」ということを主眼にして基本構造が作られています。従って、システムの理解を容易にすることとどのように加工されたデータが必要なのか naming_code を見ただけで想像が付くように考慮され、階層構造を採用しています。TSC/naming_code の基本構造と基本ルールを以下に示します。

(1) オブジェクト名（全体階層構造）

a. **グローバル名/ローカル名**

- ◆ グローバル名とローカル名の区切りを / で行う
- ◆ **場所_システム**/**機器_データ** が基本
- ◆ 全てのオブジェクト名はユニークでなければならない

(2) グローバル名（制御系を表す）

a. **場所_システム**の階層構造

- ◆ 場所は、国名、地域名、地区名、企業名、建物名、棟名などの区分等詳細分類が出来き、文字制限はしない（漢字も可）
- ◆ システムも階層構造を持たすことが出来る
- ◆ 各名前の接続には _ を使用し、階層構造を取る
- ◆ 階層構造についての詳細の規定は設けていない
- ◆ TCP/IP の IP アドレスに対応する

b. **センター名**： **データ管理名**

- ◆ 多くのビルの統括管理などのため、複数のデータ管理（建物データベース）を持つことができる
- ◆ データ管理名には、a の **場所_システム** そのものとする

(3) ローカル名（機器とデータを表す）

a. **機器_データ**の階層構造

- ◆ 機器には階層構造を持たすが、データには階層構造を持たさない
- ◆ 同種のデータに同じ名前を与えることで、グローバル名の違いによる串刺し（横並び比較）が可能となる

(4) 機器名

a. **機器_部品_子部品_孫部品**などの階層構造

- ◆ 各名前の接続には _ を使用する
- ◆ 機器、部品等には各々配列（ [] ）を付けて、その位置や場所、役割を、順番などを表す（考え方は（ 6 ）配列を参照）

(5) データ名

a. **接頭語** **コアデータ** **添字**

b. データ名はポイントの概念と同様にデータを規定するものである

- ◆ 接頭語は小文字でデータタイプを規定できる
- ◆ アナログデータは、コアデータは大文字で表し、物理量と媒体でデータの意味を表す（制御や評価指標については固有名）
- ◆ デジタルデータは、接頭語とコアデータの組み合わせで必ず表現する
- ◆ 添字は小文字で表し、役割や位置などを説明できる、_ を使用し複数説明を可能とする（階層構造ではない）
- ◆ 1つの機器に多くのセンサーがある場合のみ例外としてデータに配列（ [] ）を追加できる

(6) 配列

- a. 各々の名前（グローバル名、機器名、データ名）には配列を持つことができ同種の別物（別のオブジェクト）を表す

- b. 配列は [] で表す
- c. 小文字のみ規定されている
- d. 大文字は任意に命名できる（名前付けの自由度を確保）
- e. 位置や場所、役割、順番など様々に使用される

(7) ラベル

- a. ラベルは 1 次データとのやり取りを示すものであり、{ } で規定する
 - ◆ {c} command 1 次データへの書き込み、即ち指令・設定
 - ◆ {o} observed 1 次データからの読み込み

(8) プロパティ名

- a. プロパティ名はデータの属性を示すもので、. (ドット) で規定する
 - ◆ .Present_Value .cUnits .m_Trend など大文字と小文字の組み合わせで表す
 - ◆ プロパティ名は自由に追加できる
 - ◆ 配列要素の属性は [] にて規定する

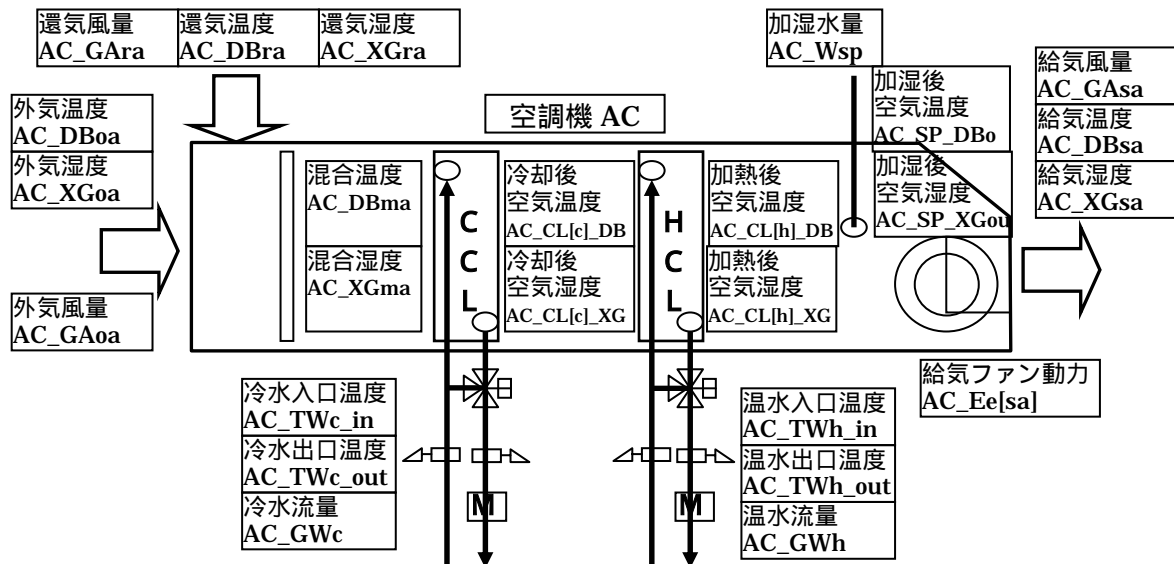




図 1 TSC/codes によるローカル名（空調機）

補足) 空調調和・衛生工学会の「建築設備システムの性能評価方法の標準化調査研究委員会」にて、室内環境および設備システムの運転状況やエネルギー消費の実態を把握するために、計測・計量に用いる変数名として、変数名のわかり易さを勘案し TSC/codes (Naming Code Ver.2) で統一を図った。

TSC21 推進協議会事務局


www.serl.co.jp/tsc21

tsc21@serl.co.jp

〒166-0003 東京都杉並区高円寺南 3-47-8

株式会社ステック環境研究所内

Tel: 03-5305-3701 Fax: 03-5305-3700

2004.09