外皮・躯体と設備・機器の総合エネルギーシミュレーションツール「BEST」の開発（その 23）

入力データ XML 構成と JPA の適用

Development of an Integrated Energy Simulation Tool for Buildings and MEP Systems, the BEST (Part 23)

Structure of input data with XML form and the application of JPA

正 会 員　○上 田 博 嗣　（大林組）　　特別会員　村 上 周 三　（建築研究所）
正 会 員　石 野 久 彌　（首都大学東京名誉教授）　正 会 員　郡　公 子　（宇都宮大学）
学生会員　木 下 泰 斗　（日本板硝子）

Hirotsugu UEDA[*1]　Shuzo MURAKAMI[*2]　Hisaya ISHINO[*3]　Kimiko KOHRI[*4]　Taito KINOSHITA[*5]
[*1] Obayashi Corporation　[*2] Building Research Institute　[*3] Tokyo Metropolitan University
[*4] Utsunomiya University　[*5] Nippon Sheet Glass

In the BEST development, we adopt XML form as input data. Therefore, in input data acquisition, it is necessary for associating object data and XML data. This is called the mapping, and the mapping is very complex work. For improvement of this problem, we adopt JPA. In this paper, we describe the merit that the application of JPA can reduce the work at the mapping.

１．はじめに

建築物の空調エネルギー消費量をシミュレーションするソフトウェアとして国内の HASP/ACSS，BECS をはじめ，海外の EnergyPlus，TRNSYS などがある．これらの入力に使用されるデータ形式は，ソフトウェア独自の配列を定義したテキスト形式データやバイナリ形式データが慣用的であった．しかし，入力データを読込む際には，プログラム内でデータ配列の定義をしなければならず，かつ第三者にとって入力データの内容が理解困難という問題があった．それ故，プログラム改良やメンテナンスが困難な状況にあった．このような背景を受け，BEST では改善策として XML[*1] と JPA[*2] を適用することとした．

本報では，BEST 開発における建物側入力データの XML 構成について述べる共に，JPA の概要およびその適用法について報告する．

２．データ形式としての XML の優位性 [1)]

従来のデータ形式が効率や性能に重点を置くのに対して，XML は拡張性，構造化，正当性の検証を重点課題として開発された．データ処理の環境が飛躍的に進歩した今日では，性能や効率は必ずしも重要ではなく，むしろ，個々の処理系で異なるデータ形式を用いることの弊害の方が目立ってきた．そこで，BEST ではマクロデザインの段階から XML データの利用を宣言してきた．データ形式を XML にすることによって，処理系の内容を全く知ることなく，データの内容を理解することができ，この故にデータの再利用効率が高まる．データの再利用効率が高いので，BEST 以外の処理系でもデータを利用することが出来る（**図 1**）．この点は HASP/ACSS，BECS，EnergyPlus，TRNSYS などと大きく異なる点である．

```
OutsideWall   owall  OWALL  0  0  S    0.7  0.9  24.6
InternalWall  floor  FLOOR  1  3  98.  0.   0.   south
```

a）テキスト形式データ例

```
- <外壁>
    <クラス名>OutsideWall</クラス名>
    <外壁名>owal</外壁名>
    <壁体構造名>OWALL</壁体構造名>
    <部位タイプ>0</部位タイプ>
    <屋外条件>0</屋外条件>
    <外表面名>S</外表面名>
    <日射吸収率>0.7</日射吸収率>
    <長波放射率>0.9</長波放射率>
    <外壁面積>24.6</外壁面積>
    <固定温度>0</固定温度>
  </外壁>
- <内壁>
    <クラス名>InternalWall</クラス名>
    <内壁名>floor</内壁名>
    <壁体構造名>FLOOR</壁体構造名>
    <部位タイプ>1</部位タイプ>
    <隣室タイプ>3</隣室タイプ>
    <内壁面積>98</内壁面積>
    <隣室温度差係数>0</隣室温度差係数>
```

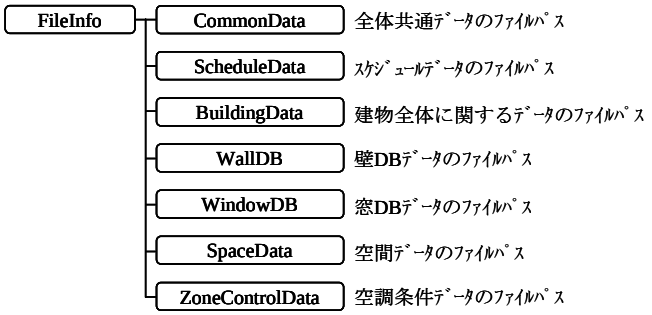
b）XML 形式データ例

図 1　テキスト形式データとXML形式データの比較例

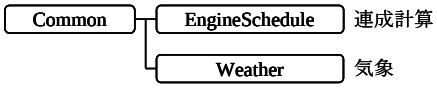
３．入力データXML構成

図 2 に建物側入力データ XML 構成を示す．入力データは大別すると以下の項目から成り立っている．

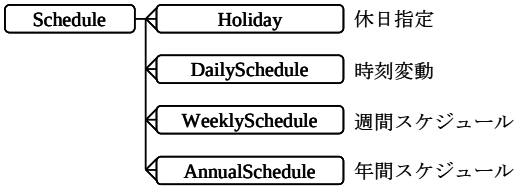
<共通・建物データ入力ファイルデータ>



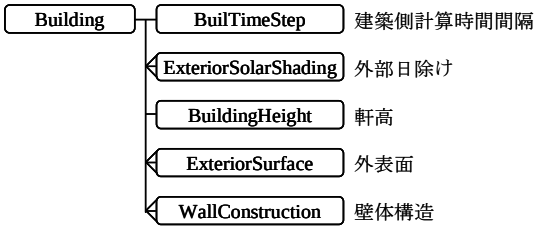
<全体共通データ>



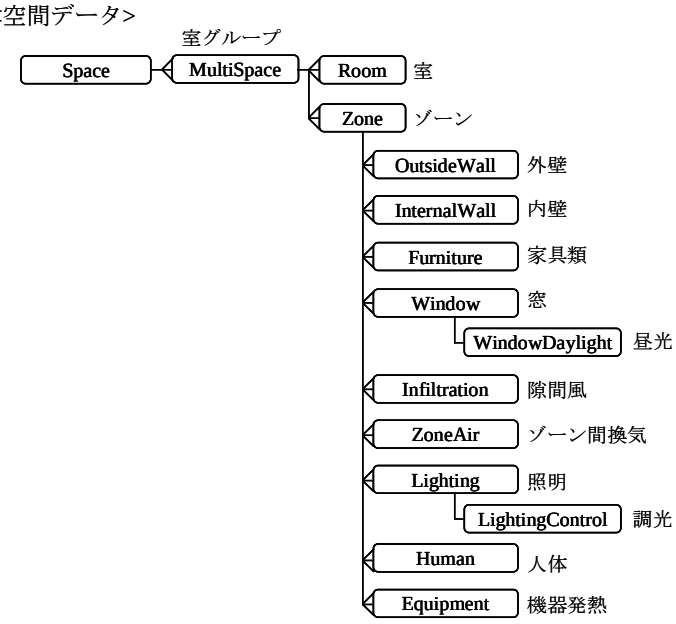
<スケジュールデータ>



<建物全体に関するデータ>



<空間データ>



<空調条件データ：空調システムとの連成計算を行わないとき>

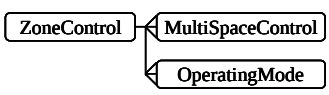


図 2　建物側入力データ XML 構成

1) FileInfo：プログラムで使用する入力データのファイルパス，データ形式を指定する入力項目である．

2) Common：建築・空調・電気・衛生計算の有無，使用する気象データを指定する入力項目である．

3) Schedule：スケジュールに関する入力項目である．休日指定，時刻変動スケジュール，週間スケジュール，年間スケジュールを指定する．

4) Building：建物全体に関する入力項目である．建物側計算時間間隔，外部日除け，軒高，外表面，壁体構造を指定する．

5) Space：空間に関する入力項目である．**図 3** に Space の概念図を示す．室グループ(MultiSpace) - 室(Room) - ゾーン(Zone)の階層から成り，ゾーンには外壁，内壁などの熱負荷要素が子要素に入る．

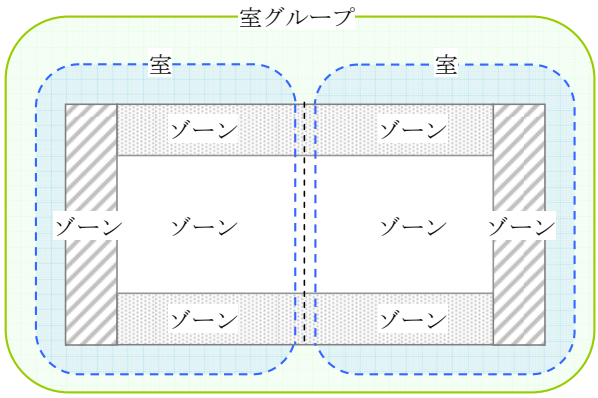


図 3　Space の概念図

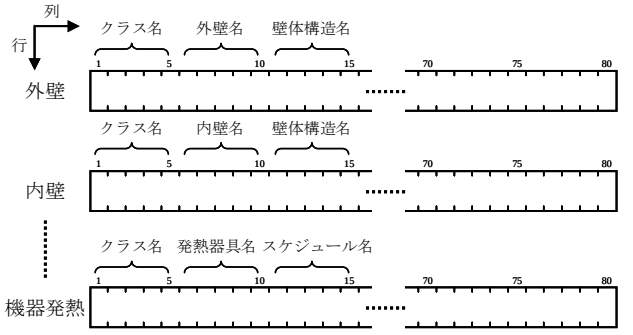
6) ZoneControl：空調システムとの連成計算を行なわない場合に，室内設定温湿度，装置容量などの空調条件を指定する入力項目である．

上記のようなテキスト，バイナリ形式データでは表現しにくかった階層構造を持つデータを扱う上でも，XML は適したデータ形式といえる．

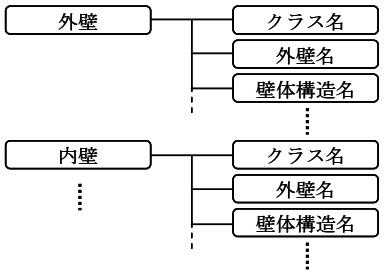
4．従来のマッピング手法の問題点 [2)~4)]

各種データ形式に保存されている入力データを Java プログラム内で扱えるデータ（オブジェクト）として読込む際には，データ間の対応関係を定義する必要がある．この対応付けのことをマッピングという．このマッピング作業は，オブジェクト - リレーショナルデータ間では O/R（オブジェクト・リレーショナル）マッピング，オブジェクト - XML データ間ではデータバインディングという．オブジェクト指向言語で各種データ形式を扱う際に最も煩わしい作業は，このマッピング作業である．**図 4** に各種データ形式のイメージを示す．オブジェクトと各種データ形式の間には，表現力や自由度に大きなギャップ（インピーダンスミスマッチ）があり，複雑な構造を持つオブジェクト（デー

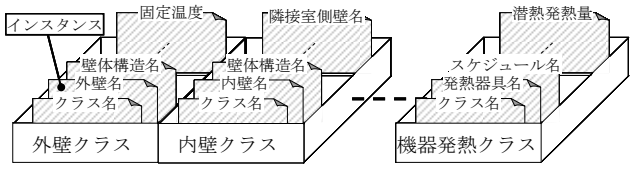
タ）を，RDB（リレーショナルデータベース）やXMLなど各種データ固有の形式に対応付けていかなければならないため，どうしても不都合が生じる．また，プログラマーは多数の入力データを一つ一つマッピングする作業を強いられることになる．それ故，マッピングの際のコーディングには大変煩雑な作業が伴い，プログラム開発工程の中でも大きな割合を占める．さらに，単調なコーディングの繰り返しを強いられるため，誤ったマッピングをしてしまうなど，発見しにくいミスを生みやすいという問題があった．この問題を改善するには，オブジェクト指向言語と各種データ形式の表現力のギャップを埋める技術を導入する必要がある．JPAもその手法の一つである．



a) RDBとしてのデータ



b) XMLとしてのデータ



c) オブジェクトとしてのデータ

図4 各種データ形式のイメージ

5．JPA(Java Persistence API)の適用

5．1 JPAの概要[5),6)]

JPAとは，テキスト，XML形式などにファイル化されたデータをオブジェクトとして扱うためのJava用フレームワーク[*3]である．主にRDBをオブジェクトとして扱うために利用されているが，BESTでは，さらに広範なデータ形式を統一的に扱うために利用している．このフレームワークを用いることで，現在は未だデータベースプログラムを用いていないが，データベースプログラムを導入してもプログラム側の変更は殆ど発生しないように出来ている．これはJPAが豊富なマッピング機能を有しており，一般的な設定においては適切なデフォルトが自動的に適用され，容易にマッピングが可能となるためである．また，XML形式データとテキスト形式データ，さらにExcelファイルの混合も認めている．一例として， RDBをオブジェクトとして扱う場合のデータ間の対応関係を述べる．RDBでは「データ」を「テーブル」と呼ばれる「表」に相当する形式で扱っている（**図4**）．従来はデータ取得に当って，RDBのテーブルとオブジェクトの定義をしなければならず，マッピングに伴う煩雑かつ冗長な作業が発生したが，JPA の適用により自動的にマッピングが行なわれ，「テーブル(表)」を「クラス」，「レコード(行)」を「インスタンス」としてマッピングされる．つまり，テーブルで表記された固有のデータ内容をプログラム内で記述する作業が解消される．これは他のデータ形式においても同様である．

5．2 適用例[6)]

BESTでは入力データをXML形式で扱っているため，XMLとオブジェクトのデータバインディング（マッピング）が必要となる．データバインディングに当たって，JAXB2.0[*4]を利用したマッピングのためのフレームワークを構築している．**図5**に建物側における適用例を示す．データバインディングに必要な情報は，XML Schemaのみである．XML Schemaとは，XMLデータの構造を記したものであり，マッピングに関する情報がコーディングされたクラスを自動生成することができる．フレームワークの内容を以下1)～3)に示す．

1) XMLデータをオブジェクトとして扱うためのクラスを利用し，XML形式の入力情報をメモリー上にインスタンスの集合として変換する．ここで，このクラスは，JAXB2.0を利用するため，プログラマーのコーディング作業は軽微になっている．

2) 入力情報のインスタンスを取得し，XML Schemaで記した構造に沿ってデータを格納する．その際に必要となるXMLの構造を示したクラスは，JAXB2.0を利用することにより，XML Schemaから自動生成できる．

3) 自動生成したクラスから，プログラム実行に必要な入力情報を取得する．

自動生成したクラスは各XMLのタグ名と同じ名前を持つクラスとしてデータを有している．そのため，例えば外壁データを取得する場合，クラス名がOutsideWallなど，直感的にデータ内容が連想可能な形で取得でき，取得するデータを取り違えるリスクを軽

減できる．さらに，今後予想される頻繁なスキーマ変更に対しても，XML Schema のみを変更すれば容易にマッピングが可能となっている．それ故，プログラマーはオブジェクトと XML 形式データのマッピングを，XML を強く意識することなく可能となる．その結果，システム間のデータのやり取りに柔軟に対応でき，データ取得やプログラム改良などが従来のマッピングと比べ自由度が高く，開発生産性の向上に繋がっている．

6. まとめ

プログラム開発において従来のマッピング時に生じていた非効率的な作業を，JPA の適用により軽微な作業量で解消できるメリットがあることについて述べた．

*1：XML（eXtensible Markup Language）は，タグ（tag）と呼ばれる情報をデータに埋込んで「データ」を表す．タグは自由に定義できる為，タグによりデータ内容を類推できる利点がある．

*2：JPA（Java Persistence API：永続化の為の標準仕様API）は，RDB のデータを扱うアプリケーションを開発する際に，データ授受（マッピング）を簡易に行なうためのJava用フレームワークである．

*3：フレームワークは，再利用可能な標準的なプログラムをまとめたものである．多くの標準的なプログラムを再利用することにより，プログラマーの作業量を軽減することができる利点がある．

*4：JAXB2.0（Java Architecture for XML Building 2.0）は，JSR222 として標準化されているデータバインディングAPIである．これを用いることにより，オブジェクトとXMLのマッピングを簡易にできる．

※JPA と JAXB2.0 の関係について

JPA は，主に RDB のデータをオブジェクトとして扱う作業を，一方，JAXB2.0 は，XML のデータをオブジェクトとして扱う作業を簡易化するために開発された．JPA と JAXB2.0 はオブジェクトと RDB 若しくは XML をマッピングするための技術という点に関しては共通しているが，その開発は各々独自の路線で行なわれている．BEST では，広範囲なデータを統一的に扱うために JPA を利用しており，JAXB2.0 の XML のマッピング技術や，その他，テキスト形式データ，Excel ファイルのマッピング技術を取り込んだものとしている．それ故，本報においては，マッピングに関する技術を JPA と称している．

【謝辞】
本報は，（財）建築環境・省エネルギー機構内に設置された産官学連携による環境負荷削減のための建築物の総合的なエネルギー消費量算出ツール開発に関する「BEST 開発普及事業研究会（村上周三委員長）」ならびにアーキテクチュア検討部会（坂本雄三部会長），建築・空調設備作業部会（石野久彌部会長），クラス構想 WG（石野久彌主査）の活動成果の一部であり，関係各位に謝意を表するものである．クラス構想 WG 名簿（順不同）主査：石野久彌（首都大学東京名誉教授），委員：一ノ瀬雅之（東京理科大学），内海康雄（宮城高専），郡公子（宇都宮大学），長井達夫（東京理科大学），羽山広文（北海道大学），上田博嗣（大林組），木下泰斗（日本板硝子），後藤裕（三機工業），菰田英晴（鹿島建設），芝原崇慶（竹中工務店），平林啓介（新日本空調），松村一誠（清水建設），渡邊剛（NTT ファシリティーズ），協力委員：瀧澤博（元鹿島建設），菅長正光（自営），二宮博史，國吉敬司，篠原奈緒子（以上，日建設計），オブザーバー：野原文男（日建設計），事務局：生稲清久（建築環境・省エネルギー機構）

【参考文献】
1)ソフトウェアエンジニアリング講座４，日経 BP 社，第六章 XML
2)http://www.atmarkit.co.jp/fdb/index/index-db.html#javadb,
Java の DB アクセスを極める
3)http://www.atmarkit.co.jp/fjava/index/index_ormap01.html,
Hibernate で理解する O/R マッピング
4) DB Magazine 2002 November, SE SHOEISHA
5)Seasar2 と Hibernate で学ぶデータベースアクセス JPA 入門，毎日コミュニケーションズ，中村年宏 著
6)Java Expert #01，技術評論社, p144-182

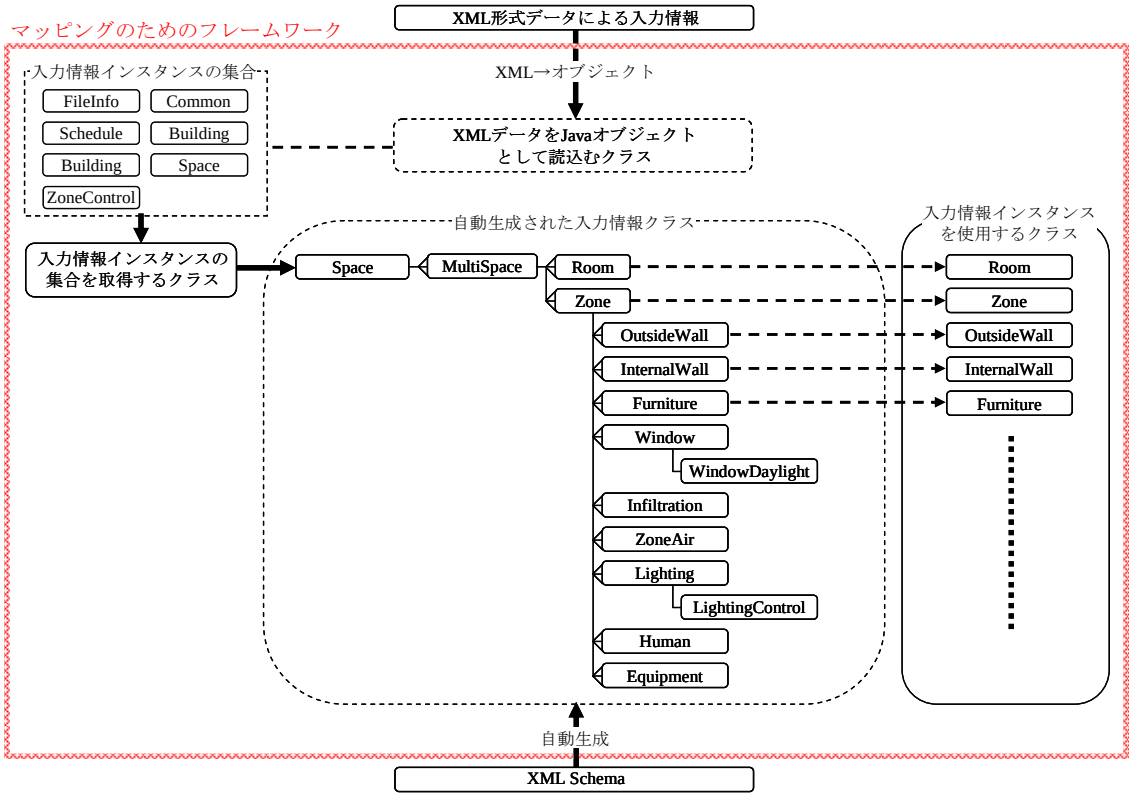


図 5　建物側における JPA 適用例