

xmlExcelHyper¹ と SIB2UI²の開発

¹ Excel 上で複雑な XML を扱うライブラリ

²衛星設計ベース2作成ツール

松崎恵一、馬場肇、宮野喜和
(ISAS/JAXA)

1. はじめに

~~~ 衛星設計ベース、衛星監視制御プロトコル、衛星の機能モデル ~~~

## 衛星の機能モデル (Functional Model of Spacecraft)

- 山田隆弘が提唱している衛星・搭載機器設計のためのモデル。
- UML のクラスダイアグラム・ステートマシンダイアグラムのサブセットに衛星・搭載機器設計固有な定義情報を加えたもの。と、考えればおよそ理解できる。

## 衛星監視制御プロトコル (Spacecraft Monitoring and Control Protocol)

- 山田隆弘が提唱している衛星の監視制御のためのプロトコル。
- 機能モデルに従って設計された衛星に対し、監視制御のためのプロトコルを定めたもの。
- パケットのユーザデータ部に詰めるメッセージを規定。

## 衛星設計ベース (Spacecraft Information Base)

- 衛星の設計情報を記述するデータベース
- ISAS では、「のぞみ」衛星以降の（「れいめい」を除く）全衛星で、テレメトリコマンドを衛星設計ベース (SIB) に登録し、SIBに従って動作する共通システムと共に利用することで、衛星システムレベルの試験・運用を効率的に進めてきた。
- 本来は、SIBへの入力機器の設計のタイミングで行われる。しかし、機器単体の開発に適用する共通システムも存在しなかったこともあり、従来 SIBへの入力は、衛星システムへの統合のタイミングとなっていた。
- 山田、松崎らは、SIBを、衛星設計段階からより積極的に適用することで衛星開発のさらなる効率化を図るため、SIB version 2 (SIB2) の検討を進めてきた。
- SIB2は、衛星の機能モデル、衛星監視制御プロトコルに従い、MMO, Sprint-A, Astro-H 以降の衛星の開発に適用される。

# 1. はじめに

## ~~~ SIB2とXMLの定義の比較 ~~~

- SIB2 は右に示すように、ツリー状のデータ構造で定義される。
- SIB2のデータ構造
  - 項目は値か1~n 種類の子項目のいずれかを持つ。
  - 後者の場合、それぞれの種類の子項目は 0~1個、0~n個、1個、1~n個の多重度を持つ。
  - 子項目の並び順は意味を持たない (が、実際のところ定義順は存在する)
  - これら、項目が持つもののパターンを複数指定し、何れかであることを規定できる。
- XMLのデータ構造
  - また、XMLのデータ構造は、「要素は0~n個の属性、0~n個の子要素、値を持ち、属性は値を持つ」であり、
  - XMLのスキーマである Relax NG では、子要素の多重度を 0~1個、0~n個、1個、1~n個、属性の多重度を0~1個、1個のいずれかで指定する。
  - 要素が持つ子要素、値の並び順に意味を持たせられる。また Relax NG スキーマで並び順を規定できる。
  - Relax NG スキーマでは、これら、要素が持つもののパターンを複数指定し、何れかであることを規定できる。
- SIB2とXMLのデータ構造の対応
  - SIB2 をXMLに 1to1 対応させる直訳は可能だが値を持つ項目の多重度が0~1個、1個の場合は、要素に対応させるか属性に対応させるかの選択が必要。(実際は、1to1対応よりも複雑な対応関係になっている)
- SIB2の定義はXML (& Relax NGスキーマ) で自然に表現できる。

→ SIB2 のデータはXMLを源泉とすることとした  
XML自身は可読性が低いのでユーザインタフェースが必要

### Definition of SIB2

機能オブジェクト定義は、機能オブジェクト毎に以下の事項を定義する。

- 1) 名前(1個)
- 2) サブ機能オブジェクト(0~n個)
- 3) アトリビュート(0~n個)
- 4) オペレーション(0~n個)
- 5) アラート(0~n個)
- 6) 状態遷移図(0~n個)
- 7) 診断ルール(0~n個)
- 8) この機能オブジェクトが有効となる親機能オブジェクトの状態(1~n個)

アトリビュート毎に以下の事項を定義する。

- 1) 名前(1個)
- 2) 外部から値を設定可能か否か(1個)
- 3) データ型(1個)
- 4) その値が有効となる状態(1~n個)
- 5) 初期値(0~1個)
- 6) 注意を喚起するためのリミット値(上限0~1個、下限0~1個)
- 7) 危険を知らせるためのリミット値(上限0~1個、下限0~1個)
- 8) 値が意味を持つ上下限值(上限0~1個、下限0~1個)

# 1. はじめに

~~~ 従来: 設計現場の複雑な表 ~~~

FMS, SMCP, DSIB の様なモデル化の構造は、モデルの仕様が収束せず、デッドロックする危険性 ...

- ユーザの要件が固まらないとツールが作り始められない
- ツールを作らないとユーザのレビューに掛けられない
- ユーザとの議論はモデルをツールでどう表現するか (View の議論) という話題になりがち
- View はユースケースに対する依存度があり、好みにも影響される
- SIB の開発も 2010年 4月にデッドロック

これまでSIB2 を入力するツールとしては、以下が試みられてきた

- 汎用のUMLツールを利用する方法 (山田) ... ユーザ側において汎用の機能を学習する労力が高く非採用
- フレームワークを用いる方法 (Ruby on Rails) (馬場) ... ユーザインタフェースを作りこむコストが高く非採用
- 専用の GUI アプリを構築する方法 (山田) ... ユーザインタフェースを作りこむコストが高く非採用
- Excelを用いて構築する方法 ... 本講演

○ Excelは、各種の設計ツールを開発するためのインフラとして非常に良く利用される

- データの一覧性や、一括操作性、任意の付加情報の記述性をユーザに浸透したツールで実現できるため
- フルのGUIツールを開発するよりは低コストで開発できるため

1. はじめに

~~~ 従来: 設計ツールの開発サイクル ~~~

▽しかし、スプレッドシートが素直に表現できるデータ構造には限界がある。

- ツールのユーザにとっても、どのようなデータ構造を表すのかマスを眺めても理解しにくい

例: SIB2UI (Spacecraft Information Base version 2 User Interface) の (設計途中の版)

| typeName        | kind                | description          | kind property |           |         |         |         |         |         |         |         |  |
|-----------------|---------------------|----------------------|---------------|-----------|---------|---------|---------|---------|---------|---------|---------|--|
|                 |                     |                      | k0            | k1        | k2      | k3      | k4      | k5      | k6      | k7      |         |  |
| /ERR_OK         | enumeration         | err ok 型 (システムが定義)   | 0x00:OK       | 0x01:ERR  |         |         |         |         |         |         |         |  |
| /ENADIS         | enumeration         | ena dis 型 (システムが定義)  | 0x00:DIS      | 0x01:ENA  |         |         |         |         |         |         |         |  |
| /HIGH_LOW       | enumeration         | high low 型 (システムが定義) | 0x00:LOW      | 0x01:HIGH |         |         |         |         |         |         |         |  |
| /PCD_ERROR_CODE | enumeration         | エラーコード               |               |           |         |         |         |         |         |         |         |  |
| #               |                     | error1               | 0x01:Error1   |           |         |         |         |         |         |         |         |  |
| #               |                     | error2               | 0x02:Error2   |           |         |         |         |         |         |         |         |  |
| #               |                     | error3               | 0x03:Error3   |           |         |         |         |         |         |         |         |  |
| #               |                     | error4               | 0x04:Error4   |           |         |         |         |         |         |         |         |  |
| /TCIU_XRX_DOH   | enumeration         | 複数の項目へ同じ表示を割り当てる特殊な例 | 0x00:OK       | 0x02:OK   | 0x04:OK | 0x06:OK | 0x08:OK | 0x0A:OK | 0x0C:OK | 0x0E:OK | 0x10:OK |  |
| #               |                     |                      | 0:TR:ERR      |           |         |         |         |         |         |         |         |  |
| /PCD            | functionalObjectRef | Pxxxx:Oxxxx:Dxxxx    |               |           |         |         |         |         |         |         |         |  |
| ERROR_CODE      | enumeration         | エラーコード               |               |           |         |         |         |         |         |         |         |  |
| #               |                     | error1               | 0x01:Error1   |           |         |         |         |         |         |         |         |  |
| #               |                     | error2               | 0x02:Error2   |           |         |         |         |         |         |         |         |  |
| #               |                     | error3               | 0x03:Error3   |           |         |         |         |         |         |         |         |  |
| #               |                     | error4               | 0x04:Error4   |           |         |         |         |         |         |         |         |  |

▽ ツールを実装するのは大変

- Excel ツールの仕様と XML ファイルの整合性をとるのが大変
- 表の大きさがリニアに効く (下記の例を参照のこと)
- 表が大きくなるとノンリニアな項 (項目と項目の整合性チェック) が登場する

例) SIB2 Excel ユーザインタフェースのこれまでのリリースと開発規模 (単位は任意の相対値)

単純計算してしまえば1カラム, 開発規模 0.5

... いずれも出力機能のみ実装、入力機能を実装すれば規模は倍

FY2005: 表2個, ~8カラム, 開発規模 5 (SIB1.5技術検討)

FY2006: 表2個, ~13カラム, 開発規模 +10 (SIB2技術検討)

FY2007: 表4個, ~30カラム, 開発規模 +6 (MMO EM ミッション総合向け)

FY2008: 表4個, ~30カラム, 開発規模 +3 (MMO EM システム総合向け)

FY2009-2010: 表9個, ~100カラム, 開発規模 +26 (MMO/Sprint-A/Astro-H 向け; 従来の開発手法の場合の見積もり)

Excel ツールの開発方法自体の最適化が必要

# 1. はじめに

~~~ Excel ツールの開発方法 ~~~


- Excel から単純な表形式のXMLを扱う仕組みは確立している (Excel 2003の標準機能¹など)
 - リレーションを設定し複数のテーブルに分割することもできる。

XML インスタンス

ユーザー インターフェイス

([XML ソース] 作業ウィンドウを使用して) Row 要素または Root 要素をセル B2 にドロップ

| Column1 | Column2 | Column3 |
|---------|---------|---------|
| A | B | C |
| X | Y | Z |




```
<?xml version="1.0"?>
<Root
  xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance>
  <Row>
    <Column1>A</Column1>
    <Column2>B</Column2>
    <Column3>C</Column3>
  </Row>
  <Row>
    <Column1>X</Column1>
    <Column2>Y</Column2>
    <Column3>Z</Column3>
  </Row>
</Root>
```

Excel 上で XML の要素や属性名を扱うツールやライブラリも存在。

- SIB2UIの古い版は、ISAS 製の xmlExcel というライブラリを使用していた。
- 同種のアプリも幾つか存在する。

| ROOT | | | |
|------|-----|-----|---|
| A | | | |
| | @A1 | | 1 |
| | B | | |
| | | @B1 | 2 |
| | | C | 3 |
| | B | | |
| | | @B1 | 4 |
| | | C | 5 |



```
<ROOT>
  <A @A1="1">
    <B @B1="2">
      <C>3</C>
    </B>
    <B @B1="4">
      <C>5</C>
    </B>
  </A>
</ROOT>
```

- しかし、Excel を用い、複雑なデータ構造を1枚のスプレッドシート上で取り扱うライブラリは見当たらない

そこで、2010年5月にExcel ツールの開発方針の抜本の見直しを実施した

参照

¹[http://msdn.microsoft.com/ja-jp/library/aa203737\(v=office.11\).aspx](http://msdn.microsoft.com/ja-jp/library/aa203737(v=office.11).aspx)

2. 適用した開発方針

~~~ (1) Excel 表記法の単純化 ~~~

- インタフェースを完全に自由に考えると、作る方も使う方も大変になるため、複雑なデータ構造の Excel 表記法に規則性を作るとよい

[例: SIB2 ユーザインタフェースのモックアップ]

(a) 固定的な階層関係は左右に並べることで示す。

機能オブジェクトはオペレーション・アトリビュートを持つ ...

(b) 可変的な階層関係は入れ子構造で示す。行ごとに表す対象が変わるため表はまたらとなる。オペレーションはパラメータを持つ ...

(c) キャプションをつける

出力データは、ツリー構造 (XML)

- この規則の範囲内で相当複雑なデータ構造まで表現できる (後述)

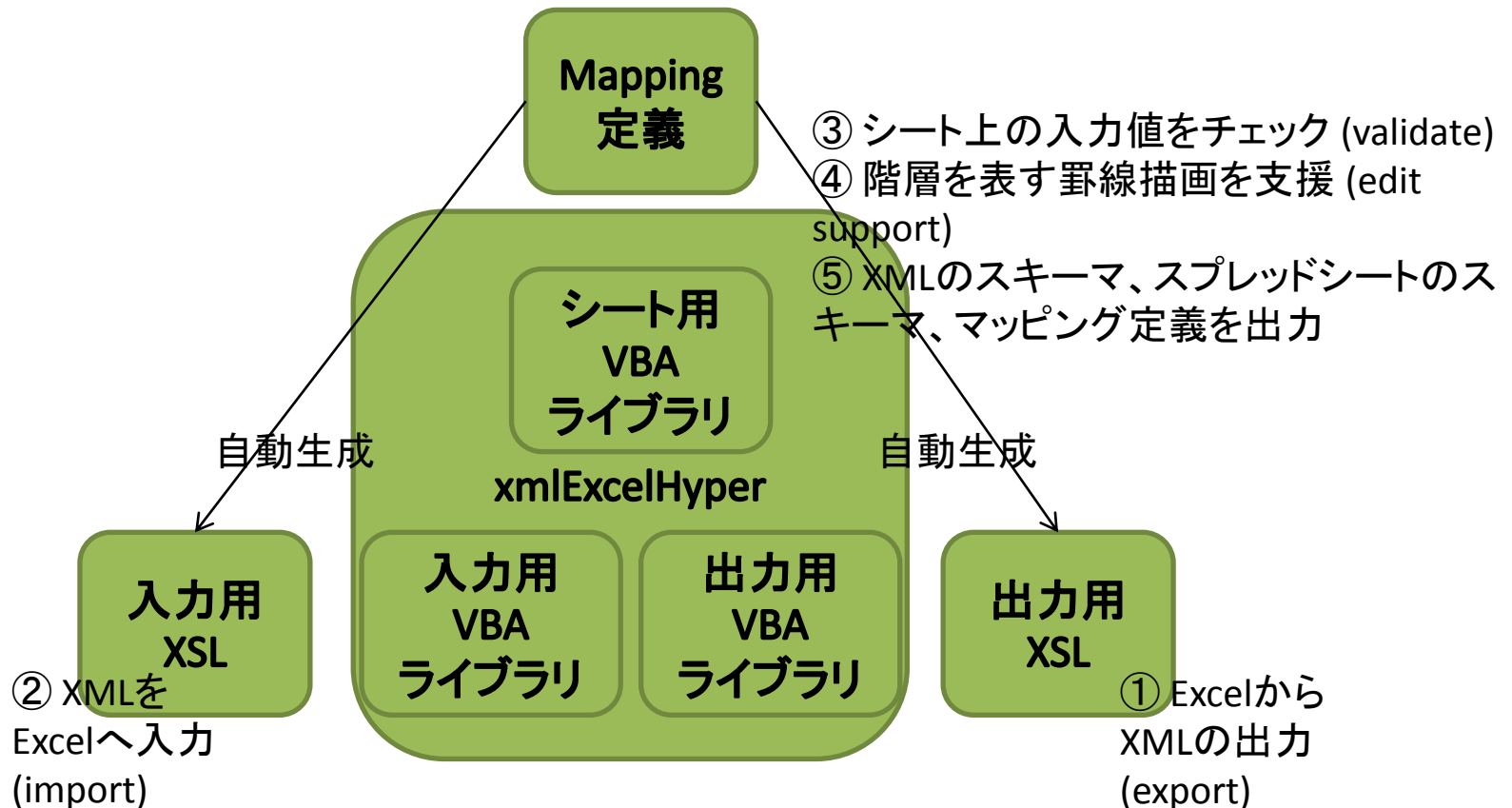




## 2. 適用した開発方針

~~~ (3) xmlExcelHyper ~~~

- Mapping 定義を記述すれば、これを参照しExcel から複雑な XML ファイルを読み書きできるツールを作ることが原理的に可能である。そこで、これを 実現するライブラリを「xmlExcelHyper」を開発し、これを適用しSIB2UIの開発をすることとした。

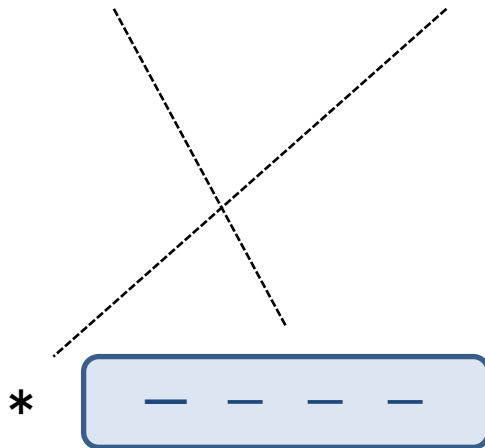


3. 開発で得られた知見

- xmlExcelHyper の初版の開発は 2010年度に終了し、2011/11/E のリリースにおいて SIB2UI もこれを利用する実装に置き換えられた。
 - △「XMLのスキーマ」の出力機能の実装は専用のスキーマ言語となっておりため限定的。原理的にはこのスキーマを RELAX NG に変換できるはずだが未対応。
- SIB2UIへのxmlExcelHyperの組み込みで問題は見いだされなかった。そこで、我々は適用した開発方針が妥当であったと考えている。
 - xmlExcelHyper 自身のバグは幾つか見つかったが修正は実施済み。
- 本講演では、以降のシートで
 - xmlExcelHyper が取り扱うことのできるデータ構造、
 - SIB2UIを実現する上で洗い出された複雑なデータを扱うための機能で xmlExcelHyper で実現されているもの・されていないものについて述べる。

表現可能なデータ構造 (1/4)

- 構造体の配列 (0~n個)



ユーザインタフェースの例: 単純テーブル

| name | attribute values for | | | |
|----------------------------|----------------------|---------------|---------------|-------------|
| attribute limit name [def] | (fatal min) | (caution min) | (caution max) | (fatal max) |
| BATL | 10 | 15 | 25 | 30 |
| BATH | 10 | 15 | 25 | 30 |
| RWA1 | 0 | 10 | 30 | 40 |
| RWA2 | 0 | 10 | 30 | 40 |
| RWA3 | 0 | 10 | 30 | 40 |
| RWA4 | 0 | 10 | 30 | 40 |
| STT | -50 | 0 | 50 | 100 |
| CH08 | -10 | 10 | 30 | 50 |
| STRPA | -10 | 10 | 30 | 50 |
| STRPB | -10 | 10 | 30 | 50 |
| SHNT | -10 | 10 | 30 | 50 |
| PCU | -10 | 10 | 30 | 50 |
| IRU | -10 | 10 | 30 | 50 |
| SAP1 | -50 | 0 | 50 | 100 |
| SAP2 | -50 | 0 | 50 | 100 |
| CH16 | -10 | 10 | 30 | 50 |

- SIB2UI は約40の構造体からなる

ユーザインタフェース上では表として表現

XMLとの対応の記述 (1/1)

- 構造体のメンバとXMLとの対応をスプレッドシート上のマッピング定義部に記述。
 - マッピング定義部はユーザに対し非表示としておく、あるいは、ファイルに出力しExcelから切り離しておく。
- 同一カラムに記し、追加・削除・並び替え時の整合を担保。
- 要素、属性の親子関係が、それなりに分かり易い記法。

- 下記は、sib:attributeLimit 要素⁽¹⁾の
 - @name
 - sib:fatalCondition/sib:minParam
 - sib:warningCondition/sib:minParam
 - sib:warningCondition/sib:maxParam
 - sib:fatalCondition/sib:maxParam

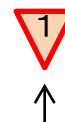
と対応付け⁽²⁾。5個のメンバのうち、4個はオプション⁽³⁾で登場することを「?」で表現⁽³⁾。XML上での要素の並び順をN.要素名で表現⁽⁴⁾。値が取りうる制約も表記⁽⁵⁾。複数の値を列挙することも表記できるが、XMLでの表現とシート上での表現の対応表が必要な場合、ユーザのプログラムが必要。

マッピング定義の例

| | | | | | |
|---------|---------------------------------------|---------------------------------|---------------------------------|---------------------------------|---------------------------------|
| (1) ... | sib:attributeLimit | | | | |
| (2) ... | @name | 1. sib:fatalCondition | 2. sib:warningCondition | | 1. sib:fatalCondition |
| (5) ... | | 1. sib:minParam ? | 1. sib:minParam ? | 2. sib:maxParam ? | 2. sib:maxParam ? |
| (5) ... | <i>\$(nonStructuralName_data-def)</i> | <i>\$(analogValue_data-def)</i> | <i>\$(analogValue_data-def)</i> | <i>\$(analogValue_data-def)</i> | <i>\$(analogValue_data-def)</i> |

(3) (4)

| name | attribute values for | | | |
|----------------------------|----------------------|---------------|---------------|-------------|
| attribute_limit_name [def] | (fatal min) | (caution min) | (caution max) | (fatal max) |



アプリ側の作りこみが必要な項目を示すマーク。

XMLとのトリッキーな関係 (1/2)

- xmlExcelHyper では、XMLの属性・末端要素が optional に指定されていて、これらに対応セルが空欄の場合、XML出力に際し、これらの属性・要素は存在しないものとして扱う。
- XMLにおいて、末端以外の要素を optional に指定すると出力の扱いが定まらなくなる場合がある。
 - 右のような例で fatalCondition, minParam, maxParam のいずれも optional に指定し、値1、値2が存在しない場合、末端以外の要素 (fatalCondition) の扱いが決まらない。
 - そこで、マッピング定義においては末端の要素と属性のみに optional (?) の指定を可能とした。

```
<attributeLimite>  
  <fatalCondition>  
    <minParam>値1</minParam>  
    <maxParam>値2</maxParam>  
  </fatalCondition>  
</attributeLimite>
```

→

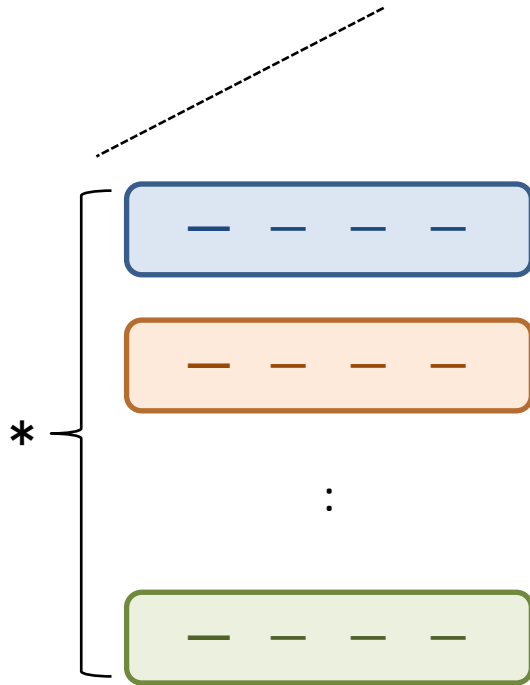
```
<attributeLimite>  
  <fatalCondition>  
</fatalCondition>  
</attributeLimite>
```

or

```
<attributeLimite>  
</attributeLimite>
```

表現可能なデータ構造 (2/4)

- (構造体の選択)の配列 (0~n個)



ユーザインタフェースの例: 選択テーブル

| property of transition | | | | | description |
|------------------------|-----------------|---------------------|-------|------------|---|
| trigger kind | trigger | transitionTime[sec] | | target | |
| command | operation name | (min) | (max) | state name | (of transition) |
| event | event name | | | state name | |
| spontaneous | | | | | |
| command | HCE_ON | | | ON | |
| command | HCE_OFF | | | OFF | |
| command | HCE_ON_ENA | | | ENA | HCE_ON を許可を設定 |
| command | HCE_ON_DIS | 1 | 10 | DIS | HCE_ON を許可をキャンセル |
| event | HCE_ON_DISABLED | | | DIS | 他のコマンドを受信したことにより HCE ON ENA 状態が解除されたことを示すイベント
(現状の SIB では他のコマンドを受信したことは定義できない) |
| spontaneous | | 10 | 20 | DIS | 一つ目のダブルコマンドが実行されてから、一定時間経過によってDISに遷移する。 |

- それぞれの行が構造体の何れを選択したかを表すカラムを設ける。
- 構造体によりフィールドの数や意味がことなるため、表はみだらとなる。
- xmlExcelHyper はマッピング定義に従い、選択用のプルダウンメニューを生成し、ユーザの選択時にと入力不能なセルを色付けを行う。

- SIB2UIは 10 の選択構造からなる

XMLとのトリッキーな関係 (2/2)

- xmlExcelHyper 自身のデータモデルでは、構造体の中身が構造体の登場箇所によって違ってはならない。
- 他方、xmlExcelHyper を用いると要素の中身が取りうるパターンが要素の登場箇所に応じ変更される XML を扱える場合がある。
 - 異なる構造体に XML 上で同一の要素を割り振りその中身が異なるように定義して良い
 - スプレッドシートから XML への出力の処理は単純であるが、XML からスプレッドシートへの入力の処理は単純とはならない。そこで、xmlExcelHyper による読み込みの処理が単純化するため、(冗長ではあるが) マッピング定義において xpath による判定式も記載することとしている。

```
<A>  
  <B1></B1>  
  <B2></B2>  
</A>
```

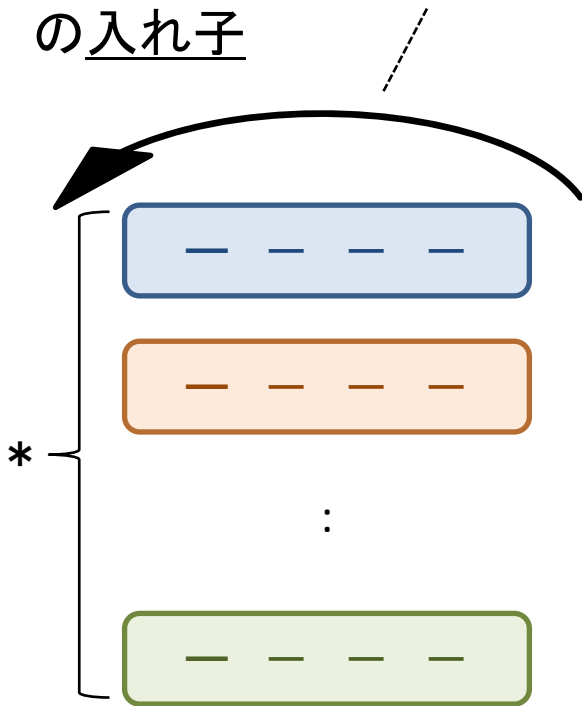
... こちらは A[B1]

```
<A>  
  <C1></C1>  
  <C2></C2>  
</A>
```

... こちらは A[C1]

表現可能なデータ構造 (3/4)

- (構造体の選択)の配列 (0~n個)の入れ子



- SIB2UIの選択構造のうち5は階層

ユーザインタフェースの例: 階層テーブル

| name | kind | type or parameter property | | operation property | | |
|----------------------|--------------------------|---|--------------------------|----------------------------|------------------|--|
| operation name [def] | operation | | | attribute change rule name | (warn, prohibit) | name of condition to be effective or always (状態遷移で規定しない場合) |
| # | constant | | | | | |
| parameter name [def] | operationAnalogParameter | (byte,unsignedByte,short,unsignedInt,long,unsignedInt,float,double,longBinary,variableLengthBinary) | (encode conversion name) | | | |
| | enumParameter | | enumeration name | | | |
| | attributeRefParameter | | attribute name | | | |
| # | reserve | | | | | |
| block name [def] | block | | | | | |
| ... | | | | | | |
| ALL_AUTO | operation | | | | | always |
| CH01.SETTEMP | operation | | | | | always |
| LOTEMP | operationAnalogParameter | short | /TCS.TPS.RANGE1 | | | |
| HITEMP | operationAnalogParameter | short | /TCS.TPS.RANGE1 | | | |
| CH08.SETTEMP | operation | | | | | always |
| LOTEMP | operationAnalogParameter | short | /TCS.TPS.RANGE2 | | | |
| HITEMP | operationAnalogParameter | short | /TCS.TPS.RANGE2 | | | |
| SET_MODE | operation | | | | | always |
| MODE | attributeRefParameter | | MODE | | | |

- 入れ子関係は表で階層的な段差により示す (同一種別の構造体の再帰的な入れ子も可能)
- xmlExcelHyper は階層関係が分かり易いように罫線を描画し、階層関係の可視化を行う。

表現可能なデータ構造 (4/4)

- (構造体の選択)の配列 (0~n個) の入れ子に際し、どのような並びで入れ子となる構造体が登場するか、並びの制約を記述できる。
マッピング定義の例：階層テーブルへの制約

```
/:= ( attribute | enumAttribute | stateAttribute | operation | alert )*  
operation := constant ? ( block | operationAnalogParameter |  
    enumParameter | attributeRefParameter | reserve )*  
block := ( operationAnalogParameter | enumParameter |  
    attributeRefParameter | reserve )*  
alert := ( alertAnalogParameter | enumParameter | attributeRefParameter |  
    reserve )*
```

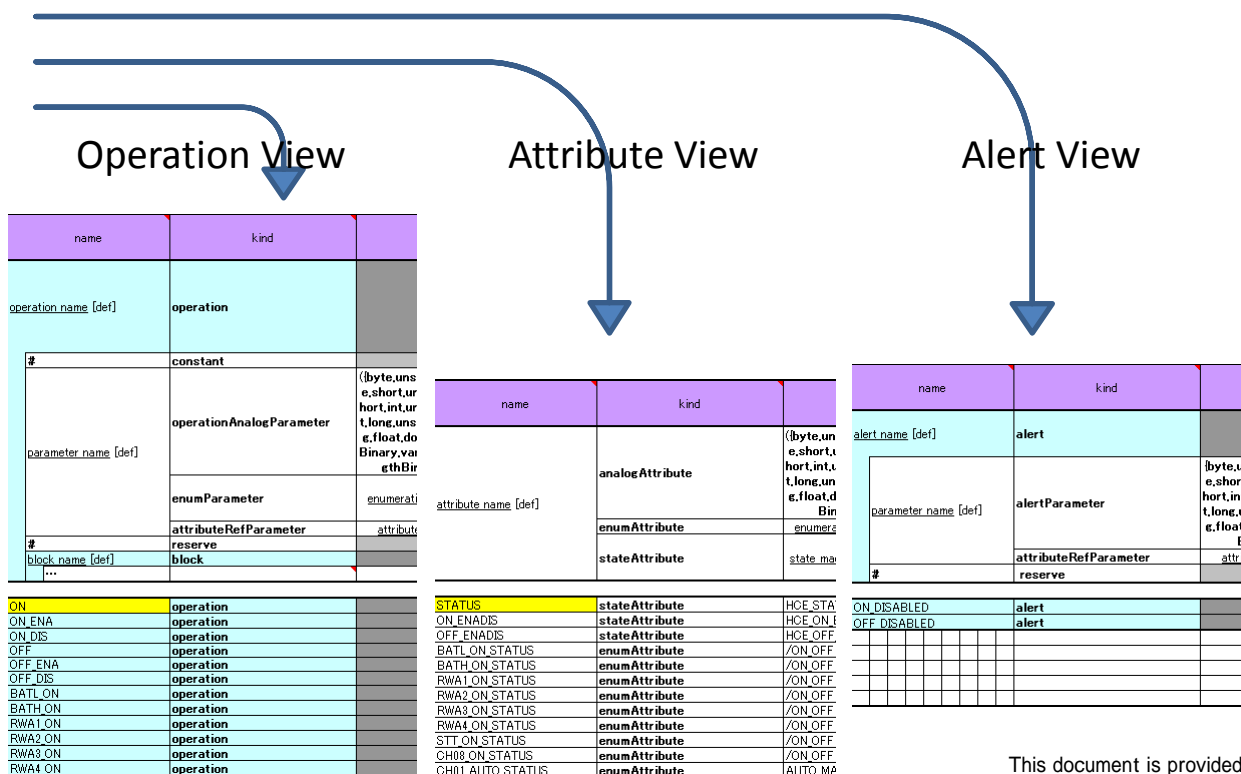
... 許される構造体の並びを正規表現で表現
- この記法により、xmlExcelHyper が表現可能なデータ構造は、XMLにて
 - 要素の中身が取りうるパターンを、要素の登場箇所に応じ変更しない
 - Mixed content (要素が、要素とテキストノードの双方持つ) は扱わない
 - 値が空の要素・属性は存在しない
- という前提において、RELAX NG (XML のスキーマの一つ) に比べ、以下の表現が不能なことのみが違いとなる。
 - 任意の要素名、属性名
 - Interleave
- **しかし、(XML構造そのままの) 複雑な階層テーブルは分かりづらい！**

複雑な階層テーブルをシンプルに見せる

- 選択された一部の構造体以外を行方向、列方向に非表示にすることで見える情報を厳選 (表のまだら具合を軽減) することができる。(一般的な機能と考えられるが xmlExcelHyper には未実装、SIB2UIに実装) 2

| name | kind | |
|------------------------------|--------------------------|---|
| attribute_name [def] | analogAttribute | (byte,unsignedByte,short,unsignedShort,int,unsignedInt,long,unsignedLong,float,double,hexBinary) |
| | enumAttribute | enumeration name |
| | stateAttribute | state_machine name |
| operation_name [def] | operation | |
| #
parameter_name [def] | constant | (byte,unsignedByte,short,unsignedShort,int,unsignedInt,long,unsignedLong,float,double,hexBinary,variableLengthBinary) |
| | operationAnalogParameter | |
| | enumParameter | enumeration name |
| | attributeRefParameter | attribute name |
| | reserve | |
| #
block_name [def]
... | block | |
| alert_name [def] | alert | |
| #
parameter_name [def] | alertParameter | (byte,unsignedByte,short,unsignedShort,int,unsignedInt,long,unsignedLong,float,double,hexBinary) |
| | attributeRefParameter | attribute name |
| | reserve | |

SIB2UI の例: Functional Object シート



| | | |
|--------------|----------------|----------------|
| ON | operation | |
| ON_ENA | operation | |
| ON_DIS | operation | |
| OFF | operation | |
| OFF_ENA | operation | |
| OFF_DIS | operation | |
| STATUS | stateAttribute | HCE STATUS |
| ON_ENADIS | stateAttribute | HCE ON ENADIS |
| OFF_ENADIS | stateAttribute | HCE OFF ENADIS |
| ON_DISABLED | alert | |
| OFF_DISABLED | alert | |
| BATL_ON | operation | |

| | | |
|---------|-----------|--|
| ON | operation | |
| ON_ENA | operation | |
| ON_DIS | operation | |
| OFF | operation | |
| OFF_ENA | operation | |
| OFF_DIS | operation | |
| BATL_ON | operation | |
| BATH_ON | operation | |
| RWA1_ON | operation | |
| RWA2_ON | operation | |
| RWA3_ON | operation | |
| RWA4_ON | operation | |

| | | |
|------------------|----------------|----------------|
| STATUS | stateAttribute | HCE STATUS |
| ON_ENADIS | stateAttribute | HCE ON ENADIS |
| OFF_ENADIS | stateAttribute | HCE OFF ENADIS |
| BATL_ON_STATUS | enumAttribute | /ON/OFF |
| BATH_ON_STATUS | enumAttribute | /ON/OFF |
| RWA1_ON_STATUS | enumAttribute | /ON/OFF |
| RWA2_ON_STATUS | enumAttribute | /ON/OFF |
| RWA3_ON_STATUS | enumAttribute | /ON/OFF |
| RWA4_ON_STATUS | enumAttribute | /ON/OFF |
| STT_ON_STATUS | enumAttribute | /ON/OFF |
| CH03_ON_STATUS | enumAttribute | /ON/OFF |
| CH01_AUTO_STATUS | enumAttribute | AUTO/MA |

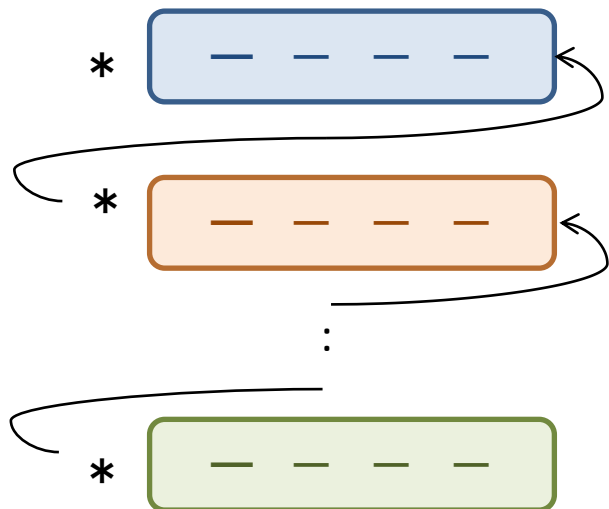
| name | kind | |
|------------------|-----------------------|--|
| alert_name [def] | alert | |
| # | parameter_name [def] | (byte,unsignedShort,int,unsignedLong,float,double,hexBinary) |
| | alertParameter | |
| | attributeRefParameter | attribute name |
| | reserve | |

親子関係のより分かり易い表現

- 構造体と構造体の固定的な入れ子関係 (より厳密には、単純・選択テーブルへの入れ子) に対し、スプレッドシート上でより分かり易い表現を用意。

ユーザインタフェースの例:

| state machine property
Function Definition | state property
Function Definition | | property of transition | | | | |
|---|---------------------------------------|-----------|------------------------|------------------|---------------------|--------|------------|
| | state name
[def] | (initial) | trigger kind | trigger | transitionTime[sec] | target | |
| state machine name
[def] | state name
[def] | (initial) | command | operation name | (min) | (max) | state name |
| | | | event | event name | | | |
| | | | spontaneous | | | | |
| 以下は例 | | | | | | | |
| HCE_STATUS | OFF | initial | command | HCE_ON | | | ON |
| | ON | | command | HCE_OFF | | | OFF |
| HCE_ON_ENADIS | DIS | initial | command | HCE_ON_ENA | | | ENA |
| | ENA | | command | HCE_ON_DIS | 1 | 10 | DIS |
| | | | event | HCE_ON_DISABLED | | | DIS |
| | | | spontaneous | | 10 | 20 | DIS |
| HCE_OFF_ENADIS | DIS | initial | command | HCE_OFF_ENA | | | ENA |
| | ENA | | command | HCE_OFF_DIS | 1 | 10 | DIS |
| | | | event | HCE_OFF_DISABLED | | | DIS |
| | | | spontaneous | | 10 | 20 | DIS |

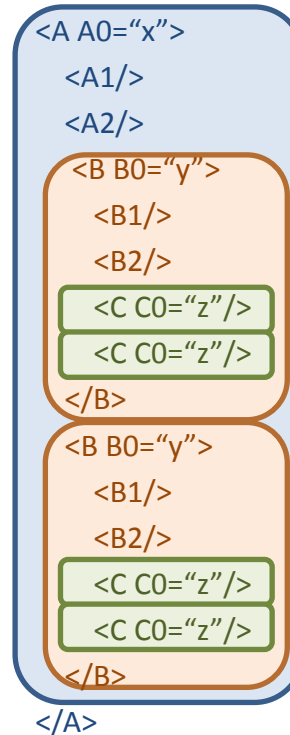
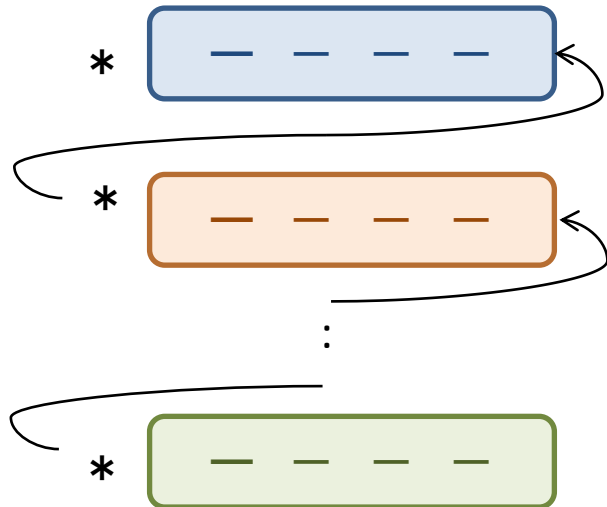


- 入れ子関係を左右に並べることで示す
- xmlExcelHyper は階層関係が分かり易いように罫線を描画し、階層関係の可視化を行う。

3

親子関係のより分かり易い表現

- 構造体と構造体の固定的な入れ子関係 (より厳密には、単純・選択テーブルへの入れ子) に対し、スプレッドシート上でより分かり易い表現を用意。



- 構造体はXML上で一つの要素に対応する。入れ子となる構造体に対応する要素は、親の要素の直下かつ、構造体の要素の末尾に並べられる。
- 既存のXML定義に対して、xmlExcelHyperを適用する場合、この制約が非適合となる可能性がある。この場合、外付けのXSLを実施しない限り取り扱えない。

xmlExcelHyper 非対応の表現

スプレッドシート上で複雑なデータを表現するにはその他の方法もありうる。

- セル内で構造体や繰り返しを表現すること

- 空白やカンマ区切りなどで繰り返しを表現
- 項目を区切り字(:など)で区切り、構造体のメンバを表現

| |
|---|
| 00:BLACK 01:RED 02:GREEN 04:BLUE 07:WHITE |
|---|

- 構造体を選択するための文法を定義

| |
|-----------------------|
| command HCE_ON |
| command HCE_OFF |
| command HCE_ON_ENA |
| command HCE_ON_DIS |
| event HCE_ON_DISABLED |
| spontaneous |
| command HCE_OFF_ENA |
| command HCE_OFF_DIS |
| HCE_OFF_DISABLED |
| spontaneous |

- この構造をXMLの要素や属性を用いて表現するには、外付けプログラムで対応が必要。(SIB2UIでも対応)
- データ型ライブラリの文字列型に正規表現の制約をかければ validationのみは実現可能



- 複数のセルを列方向 (さらに行方向) に並べ繰り返しを表現すること

| | | | | |
|-----------|----------|----------|----------|--|
| 00:BLACK | 01:RED | 02:GREEN | 04:BLUE | |
| 03:YELLOW | 05:MAGEN | 06:CYANT | 07:WHITE | |

シートの分割

- データ構造においては 1:n の関係を ID を用いた参照関係により二つのデータ構造に分割できる。
- 参照関係には、以下がある。
 - A) 親側(1)にIDを定義し、子側(n)のデータ構造から参照
 - B) 子側(n)にIDを定義し、親側(1)のデータ構造から参照SIB2UI は 8 シート (enumeration, functionalObject, attributeSequence, stateMachine, conditionAndEvent, conversion, attributeLimit, attributeChangeRule) に分割され、
A) の参照関係が7個、B) の参照関係が7個張られている。functionalObject シートがメインのシート。attributeSequence を除くシートは B) の関係で参照されている。そこで、Attribute Sequence 以外のシートにおいては、シートを分割する以外の解は無い。
- xmlExcelHyper は複数のシートを扱えるが、シート間の参照関係をサポートする機能を持たない。定義と参照の整合性確認には外付けのプログラムが必要。
- (SIB2のXMLは、シート間の情報のマージは行っていない)
- ツールの利用においては、定義元・参照先に移動する機能、参照において記載可能な候補から入力する機能があると有用である。
- ユースケースに応じ、シート間の情報を集約した view で見せる機能は有用。
SIB2UIにおいては、上記2つの項目を、外付けの別プログラム SIB2Viewer にて、この機能を実現している。



4. まとめ

- 複雑なデータ構造を持つデータベース (衛星設計ベース; 設計結果は、構造体40, 選択構造10うち階層は5, シート数8, 参照関係 7+7) に対し、スプレッドシート (Excel) から値を入力、XMLファイルへ入出力するインタフェース SIB2UI の開発を実施した。
- ユーザインタフェースは、ユーザの分かり易さを考慮し、表現するデータ構造に対応した規則に従うものとした。
- スプレッドシートとのXMLファイルの対応は、スプレッドシート上にマッピング定義を記述することで、XMLファイルの仕様とスプレッドシートの仕様、両者の対応がいつに視認でき、項目の追加、削除、並び替えにおいて整合性が担保されるようにした。
- マッピング定義に基づき、スプレッドシート上のデータの妥当性を確認、データの階層関係を可視化、XMLファイルとの入出力を行うライブラリを作成することで、アプリケーション側の開発の省力化を図った。
- xmlExcelHyper の階層テーブルを用い「原理的に扱える」XMLは、「実用上あまり問題とならない？」前提の元では、RELAX NG スキーマと同じであるが、そのままでは可視性に難がある。データ構造の入れ子関係を改善する表現を用いると、適用可能なXMLは限定的となる。結果として、現実的には、xmlExcelHyper は、XML の仕様とスプレッドシートの仕様を同時に決められるユースケースにのみ適用可能と思われる。
- SIB2UIの開発において、複雑なデータ構造を扱うために必要と判断した幾つかの処理 (1: 選択項目のスプレッドシートとXMLの表現の対応、2: 階層テーブルをシンプルに見せるためのViewの作成、3: セルでの繰り返し・構造体の表現、4: 定義・参照関係のチェック、5: 定義・参照関係の編集をサポートする機能) は、現状では xmlExcelHyper には実装されておらず、アプリケーション (SIB2UI) 側で自作となっている。これらの機能も xmlExcelHyper にてサポートされることが期待される。