xmlExcelHyper[1] を用いた SIB2UI[2] の開発

[1] Excel 上で複雑な XML を扱う汎用ライブラリ
[2] 衛星情報ベース２作成ツール

松崎恵一 [*1]　山下美和子 [*2]　馬場肇 [*3]　西村佳代子 [*4]　宮野喜和 [*4]

Development of SIB2UI[1] with xmlExcelHyper[2]

[1] Spacecraft Information Base Version 2 Users' Interface
[2] General library on Excel which handles complicated XML data structure

Keiichi MATSUZAKI[*1], Miwako YAMASHITA[*2], Hajime BABA[*3], Kayoko NISHIMURA[*4] and Yoshikazu MIYANO[*4]

Abstract

Design of spacecraft and onboard components of ISAS satellites started after the BepiColombo/MMO project are stored in a database called Satellite Information Base Version 2 (SIB2). This database has a complicated data structure and its model itself is under development. In parallel activities, the spacecraft and onboard components must be developed with this model. In the developments of the spacecraft and onboard component, tool which support design with the model, SIB2UI, is needed and should also be developed. On the other hand, the model should be determined with feedback from design of spacecraft and onboard component. In the development of the model of SIB2, shortening of span of the feedback was the most important subject. To resolve the subject, we introduced a technique to implement design tool. This technique consists of a general library xmlExcelHyper which handles a complicated data structure on a spreadsheet program, Microsoft Excel. With this library, developer of design tool can easily implement Excel-based tool which visualizes, imports and exports tree-base data structure which has degree of freedom of selection and repeat.

Key Words: Development process, Modeling, Design Tool

概　要

ISAS の衛星では，BepiColombo/MMO 衛星以降の搭載機器・衛星において，衛星情報ベース２(Satellite Information Base Version 2; SIB2) というデータベースに設計を記述する．このデータベースは，複雑なデータ構造を持ち，どのようにモデル化するか自身が開発対象である．他方で，このモデルを用いて搭載機器・衛星の開発も行う必要があり，モデルに従って設計を記述するツール (SIB2UI) が必要である．どのようにモデル化するかを決めるには，搭載機器・衛星の設計の実例を配慮する必要があり，このフィードバックを如何に早く収束させるかが課題であった．我々は，複雑なデータ構造を扱うモデルに従って設計するツールを効率よく開発する手法を考案し，適用することで，フィードバックの収束を加速した．その核となるのが，スプレッドシート (Microsoft Excel) 上で複雑なデータ構造を扱うための汎用ライブラリ xmlExcelHyper である．この，ライブラリを用いるとスプレッドシート上で選択構造や繰り返し構造からなる XML のツリー構造をそのデータ構造に応じ可視化・入出力するツールの開発を容易に実現できる．

---

*1 宇宙科学研究所 学際科学研究系
(Department of Interdisciplinary Space Science, Institute for Aerospace and Aeronautical Science (ISAS))
*2 宇宙科学研究所 Bepi Colombo プロジェクトチーム (Bepi Colombo Project Team, ISAS)
*3 宇宙科学研究所 宇宙科学プログラム・システムズエンジニアリング室
(Space Science Program Systems Engineering Office, ISAS)
*4 宇宙科学研究所 科学衛星運用・データ利用センター 衛星運用グループ
(Satellite Operations Group, Center for Science Satellite Operation and Data Archive, ISAS)

1 はじめに

ISAS (Institute for Space and Aeronautical Science) の衛星開発では，BepiColombo/MMO, SPRINT-A, ASTRO-H 以降，衛星情報ベース 2 (Spacecraft Information Base Version 2; SIB2) というモデルに基づいたデータベースに設計を記述する．このモデルはツリー状のデータ構造をもち，XML に格納される．他方，衛星設計の現場では，これをスプレッドシート (Microsoft Excel，以下単に Excel) 上のユーザインタフェースから扱うことが求められている．そこで，我々 SIB2/GSTOS-1 プロジェクト [1] では，Excel 上で，SIB2 を扱うツール，SIB2 作成ツール (SIB2UI[a]) の開発を行っている．

モデルに基づく開発の実現においては，モデル自体を開発しながら個々の開発を実施することがある．この場合，モデル自身の開発と，モデルを扱うツールの開発，モデルを用いた個々の開発が並行で進むこととなり，リスクが高い．この状況を打開するには，モデルを扱うツールを効率よく開発することで，モデル自身の開発と個々の開発の議論を加速し，モデルの仕様を早く収束させることが重要となる．そこで，我々は，Excel 上で複雑なデータ構造を扱うための汎用ライブラリ (xmlExcelHyper[b]) を開発し，これを SIB2UI に適用することで，効率的な開発を実現した．

本論文では，xmlExcelHyper を適用したツールにおいて取扱い可能なデータ構造 (Excel での見え方，XML で可能な表現範囲) とその記述方法，xmlExcelHyper を用いたモデル・ツール開発の流れや効能について，SIB2UI の実例を用いて述べる．まず，2 章では，開発ツールの開発手法を実証の題材としたモデル SIB とこれを扱うユーザインタフェース SIB2UI について紹介する．3 章では，このようなモデルに基づく，設計ツールを開発する上での課題について示す．4 章では，我々が考案した汎用ライブラリ xmlExcelHyper に基づく設計ツールの構築について示す．5 章では，xmlExcelHyper が扱うデータ構造・XML 構造とこれに関わる機能について示し，6 章でまとめを示す．

2 Spacecraft Information Base (SIB)

SIB は，衛星の設計情報を記述するデータベースである．ISAS では，「のぞみ」衛星以降の「れいめい」を除く全衛星において，テレメトリとコマンドを 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，衛星システムの試験・運用を効率的に進めてきた．

SIB は搭載機器の設計を記述するものである．そこで, その入力は, 設計のタイミングでなされるべきである．しかし，従来は，搭載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ため，SIB への入力が搭載機器の衛星システムへの統合のタイミングとなってしまっていた．また，後追いで設計結果の入力作業を実施するため，衛星レベルの開発期間に，搭載機器単体の設計あるいは実装とデータベース入力の間の整合性を確認する作業が必要となってしまっていた．

この状況を打破し，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効率化を図るため，山田，松崎らは，搭載機器の開発において，搭載機器単体の試験から適用可能な共通ソフトウェア GSTOS (Generic Spacecraft Test and Operations Software) と，新世代の SIB，SIB2 の検討を進めてきた．SIB2 においては，搭載機器・衛星を，山田 [2][3] が提唱する「衛星の機能モデル (Functional Model of Spacecraft; FMS[4])」と「衛星監視制御プロトコル (Spacecraft Monitoring and Control Protocol; SMCP [5])」に基づき設計する．ここで，FMS は，およそ，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である．また，SMCP は，衛星の監視制御のためのプロトコルであり，FMS に従って設計された搭載機器・衛星において，パケットのユーザデータ部に詰めるメッセージを規定するものである．

SIB2 のモデルの定義 (Definition of SIB2; DSIB2[6]) の抜粋を表 1 に示す．この表に登場する「機能オブジェクト」は，搭載機器や搭載機器内の機能の塊に対応する．また,「オペレーション」はコマンドにより駆動される動作,「アトリビュート」はテレメトリにより取得可能な値,「アラート」は，異常など機能オブジェクトから出力される間欠的なテレメトリを示し，これらが FMS の核となる．

SIB2UI の Excel ファイルは，1 つのファイルがサブシステムや搭載機器など衛星設計の一部に対応し，複数の Excel ファイルを取りまとめ衛星全体の設計を表現する．それぞれのファイルは表 2 に示すようにデータ構造及び衛星開発のステップを考慮して決められた 9 つのシートから構成される．これらのうち，最も基本的なのは functionalObject 及び attributeSequence シートである．functionalObject シートには, 機能オブジェクト, オペレーション, アトリビュート, アラートを定義する．また，attributeSequence シートには，アトリビュートをテレメトリとして取得する際の並び順を定義する．

SIB2UI は，4 つの機能 Edit Support，Import，Export，Validate から構成される．SIB2UI のシートは functionalObject シー

a) 本稿執筆の時点での最新版は 3.1.2.0 版
b) 本稿執筆の時点での最新版は V11L01

トや attributeSequence シートなど，それぞれが複雑なデータ構造を担っている．そこで SIB2UI には，どのようなデータ構造かわかり易いよう可視化の機能 (Edit Support) を設け，罫線描画，セルの結合や背景色の設定を行っている．また，編集中の任意のタイミングでシート内・シート間のデータ構造の整合性が取れているかチェックする機能 (Validate) も持たせている．さらに，XML ファイルからの入力 (Import)　と出力 (Export) の機能も持たせている．なお，SIB2UI では，実装を単純にするため，シート 1 枚を XML ファイル 1 つに対応させ，シート間の情報のマージは行っていない．

表 1　SIB2 のモデルの定義（抜粋）

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値 (上限 0~1 個，下限 0~1 個)

表 2　SIB2UI を構成するシート

シート名	概要
0.management	衛星名を定義する．
1.enumeration	数値と文字列の対応を定義する．
2.functionalObject	機能オブジェクト（アトリビュート，オペレーション，アラート）を定義する．
3.attributeSequence	テレメトリとして受信するアトリビュートの並び順を定義する．
4.stateMachine	状態を表す数値，文字列を定義すると共に，遷移の条件をオペレーションやイベントを参照することで定義する．
5.conditionAndEvent	アトリビュートの評価式の組み合わせ (論理式) により条件，イベントを定義する．
6.conversion	アナログ値の変換を定義する．
7.attributeLimit	アトリビュートのリミット値の危険値 (fatal limit)，警告値 (caution limit) を定義する．
8.attributeChangeRule	オペレーションに対するアトリビュートの変化則を定義する．

SIB2 作成ツール (SIB2UI) 入力リファレンス 第 1.5 版 2012 年 3 月 16 日から抜粋

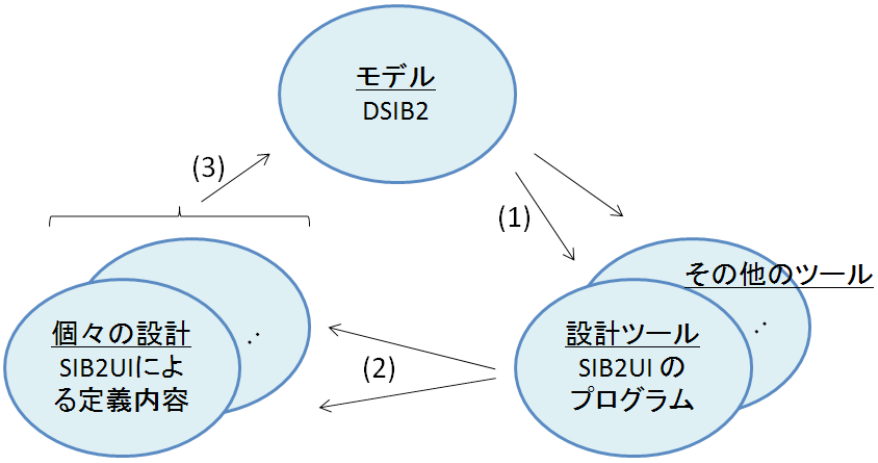
SIB2 及び SIB2UI は，ISAS の衛星に適用することを目的に開発を行っている．そこで，ISAS の衛星プロジェクト向けには web ページ [c] 経由での配布を行っている．他方，SIB2・SIB2UI 自身は，任意な衛星に対応する汎用な技術である．そこで ISAS 以外の衛星プロジェクトに対しても，個別に条件を設定し配布を行っている．

SIB2 は表 1 に示されるように，ツリー状のデータ構造を柱に定義されている．ツリー状のデータ構造を扱う記法として，XML が普及している．そこで，SIB2 においても，データベースの源泉の表記として XML を採用した．RELAX NG [7] など，XML のスキーマ言語を用いると，XML 構造における多重度を定義できる．そこで，DSIB2 に定義される多重度も，スキーマ言語の記述に焼き直すことができる．

c) http://www.c-soda.isas.jaxa.jp/software/project-tools.html

3 設計ツール開発における課題

一般に，SIB2 のようなモデルの策定においては，議論がデッドロックに陥り，仕様が収束しないというリスクが存在する (図 1 参照). モデルが定まらないと，これをサポートする設計ツールが作れない. 他方で，実際の設計で使うことができるツールが存在しないと, ユーザ (設計者) からは親身なレビューを受けられない. また, ツールのイメージを作り, ユーザとの議論を始めたのちも，ユーザとの議論はモデルをツールでどう表現するか (View) の話題に時間が費やされることになる. View はユースケースに対する依存度や個人の好みにも影響されるため，議論に多くの時間が費やされる. さらに，実例に基づかないと最適なモデルは作れない. 実際，SIB2 の開発も，SIB2UI の BepiColombo/MMO, SPRINT-A, ASTRO-H 向け版の開発の初期段階 (2010 年 4 月頃) において，この種のデッドロックに陥っていた.



(1) ツールの開発者は，モデルに基づき，設計ツールなど各種のツールを開発する.
(2) 個々の機器の設計者は，設計ツールを用いて個々の機器の設計を行う.
(3) モデルの制定者は，個々の機器の設計において有用なようモデル化を行う.

図 1 モデルに基づく開発実現におけるスパイラル

モデルに基づき設計を行うツールを作成する手法は多様である. その一つとして，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. これは，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，また，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と考えられる.

実際，SIB2 に基づく設計ツールにおいても，以下の 4 つの実装方法が試みられてきた. 1) 山田による，汎用の UML ツールを利用する方法，2) 馬場による，Web 用のフレームワーク (Ruby on Rails) を用い専用のツールを構築する方法，3) 山田による，スタンドアロンの専用プログラムを構築する方法，4) 松崎らによる Excel 上で構築する方法. これらのうち, 1) の方法は, ユーザ側において汎用で多機能な UML ツールの機能を学習する労力が高いため非採用となった. また, 2) 及び 3) の方法は，専用のユーザインタフェースを作りこむコストが高く SIB2 のモデル全体の実装には至らなかった. 結局，SIB2 に基づく設計ツールとして，実現に至ったのは SIB2UI に適用した 4) の Excel を用いる方法のみであった.

Excel は，モデルに基づく設計ツールで利用されるものの，表現されるデータ構造のわかり易さには限界がある. スプレッドシートが素直に表現できるデータ構造は，表，つまり，１種類の構造体の繰り返し構造のみである. これを超えるデータ構造を扱う場合には，常に，どのように表現するのかという View の問題に突き当たる. 図 2 に，SIB2UI の開発段階で登場したユーザインタフェースの案を示す. これは，２種類の構造体が親子となった繰り返し構造を表すものであり，２案が併記されている. この種の表現の自由度は，ツールが出来上がった後も，ユーザに不利益を与える. つまり，どのようなデータ構造を表すのかマス目を眺めても俄かにはデータ構造を把握しづらく，その理解に要する学習時間が個々の開発の設計のコストを押し上げることとなる.

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						
/ENA_DIS	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						
/POD:ERROR_CODE	enumeration	エラーコード								
#		error1	0x01:Error1							
#		error2	0x02:Error2							
#		error3	0x03:Error3							
#		error4	0x04:Error4							
/TOIU:XRX_OOH	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
#			OTR:ERR							
/POD	functionalObjectRef	P:xxxx D:xxxx D:xxxx								
ERROR_CODE	enumeration	エラーコード								
#		error1	0x01:Error1							
#		error2	0x02:Error2							
#		error3	0x03:Error3							
#		error4	0x04:Error4							

図 2　複雑なデータ構造をスプレッドシート上に表した例

Excel を用いることで，Excel 自身が持つ GUI の機能を流用し，ある程度設計ツールの開発費を抑えることができる．しかし，その場合にも一定量の開発コスト・開発期間が必要である．表 3 に，SIB2UI の実際の開発規模感を示す．開発規模は，ほぼ表現しているデータ構造の大きさに比例 (1 カラムあたり ~0.5) しており，その低減が望まれる．

表 3　SIB2UI のこれまでのリリースと開発規模感

(単位は任意の相対値)

FY2005: 表 2 個 , ~8 カラム , 開発規模 5 (*1)
FY2006: 表 2 個 , ~13 カラム , 開発規模 +10 (*2)
FY2007: 表 4 個 , ~30 カラム , 開発規模 +6 (*3)
FY2008: 表 4 個 , ~30 カラム , 開発規模 +3 (*4)
FY2009-2010: 表 9 個 , ~100 カラム , 開発規模 +26 (*5)

*1: SIB1.5 技術検討
*2: SIB2 技術検討
*3: BepiColombo/MMO EM ミッション総合試験向け
*4: BepiColombo/MMO EM システム総合試験向け
*5: BepiColombo/MMO, SPRINT-A, ASTRO-H 向け ; 従来の開発手法の場合の見積もり
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と見込まれる．

Excel から XML データを扱う手法は，幾つか存在する．図 3 の (a) は，単純な表形式の XML を扱うものである．この仕組みは Excel 2003 以降 Excel の標準機能として実装されている．一般にデータ構造は，キーを設定し，リレーションを張られた複数のデータ構造，最終的には単純な表にまで分割できるため，この方法のみでデータを扱うことは可能である．しかし，この方法では，情報が複数のテーブルに分かれることとなるため，設計ツールとしては使いづらい．図 3 の (b) は, 逆に, Excel 上に XML の要素や属性名をそのまま見せることで任意の XML を扱うものである．この種のツールはいくつか存在するが，要素名や属性名は必ずしもわかり易いものではないし，全てをツリーで表現されたデータ構造はユーザにわかり易いものではないため, やはり, 設計ツールにはなじまない．設計ツールの開発コストを下げるには，ある程度複雑なデータ構造を よりコンパクトに Excel 上に表現し，その操作をサポートする汎用ライブラリが求められる．そこで，我々は，2010 年 5 月に Excel ツールの開発方針の抜本的見直しを実施することとした．次章では，見直した開発方針と，その核となる汎用ライブラリ xmlExcelHyper を用いた設計ツールの構築について示す．

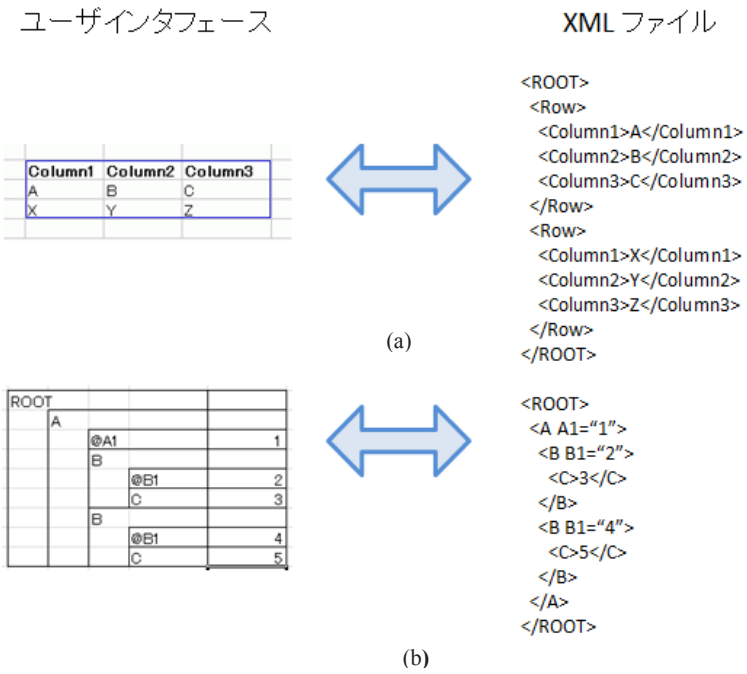


図 3 XML データを Excel 上で扱う既存の手法

4 xmlExcelHyper による設計ツールの構築

ユーザインタフェースを無秩序に設計すると，ツールを作る側に労力を要するし，使う側のユーザビリティも悪いものとなる．そこで，我々が設定した開発方針のひとつめは，ユーザインタフェースにおけるデータの表現方法の自由度を減らすというものである．図 4 に，我々が選定したデータ構造の表現方法のイメージを示す．この表現方法では，ひとつとして，図 4 の (a) に示されるように，種別が固定的な階層関係は左右に並べることで示す (5.4 参照)．例えば，SIB2 においては「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」と言う関係は，SIB2UI において，この方法で表現した．また, 図 4 の (b) に示されるように, 種別が可変的な階層関係は入れ子構造で示す (5.3 参照)．例えば，SIB2 においては「オペレーションはブロックもしくはパラメータを持つ．ブロックはパラメータを持つ」と言う関係は，SIB2UI において，この方法で表現した．この方法を用いると，行ごとに表す対象が変わることとなり，表がまだらになる．いずれの場合においても，図 4 の (c) に示されるように，データ構造と同じ記法で表のラベルを設けることとした．

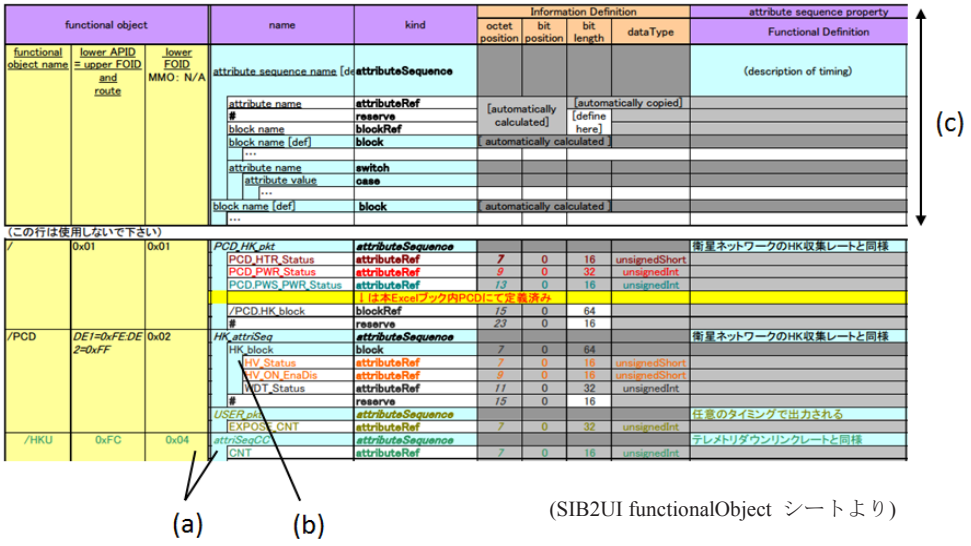


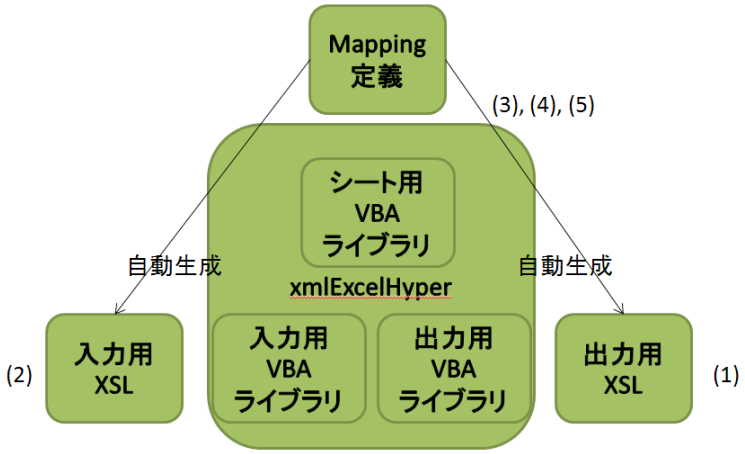
図 4 選定したデータ構造の表現方法のイメージ

我々が設定した開発方針のふたつめは，複雑な表と XML の対応の記述を把握しやすく，整合をとり管理しやすいものとすることである. この方法では, 表の中身が XML の何に対応するかの対応 (マッピング定義) を, ユーザインタフェース上の表のラベルと上下にならべて表記する．なお，マッピング定義の記法については 5 章で，図 7，図 9，図 11 など例を用い示す．この記法を採用することで，項目の並び順の変更や，追加，削除に関し Excel ユーザインタフェースと XML ファイルの仕様策定を両者の対応の整合性をとりつつ実施でき，モデルとツールの開発が効率化される．SIB2・SIB2UI の開発では，この記法を投入することで，SIB2 のモデル決定におけるデッドロックを解消し，モデルが定まらないことによる開発コストの増大を抑制した．

マッピング定義は，XML の記法 (XML のスキーマ)，スプレッドシート上の記法 (スプレッドシートのスキーマ)，スプレッドシートと XML の対応関係の情報をすべて含んでいる．そこで，マッピング定義を参照すれば，原理的に，Excel から XML ファイルを読み書きすることが可能である．我々は，マッピング定義を参照し，Excel から XML ファイルを読み書きする汎用ライブラリ xmlExcelHyper を開発し，これを SIB2UI の開発に適用することとした．図 5 に xmlExcelHyper の機能・実装イメージを示す．xmlExcelHyper を用いるとサポートされるデータ構造の範囲内であれば，SIB2UI の 4 つの機能に対応する図中の (1)(2)(4)(5) の機能をツール毎のプログラミングなしで実現できる．

xmlExcelHyper は, Excel 上で動作するため, 処理ロジックは基本的に VBA (Visual Basic for Application) で実現している．他方，スプレッドシートと XML 上の相互変換のロジックは，Excel に限らず一般的なスプレッドシートプログラムに適用可能なものである．そこで，将来，xmlExcelHyper を，他のスプレッドシートプログラムへも転用できるよう，Excel からのファイルの入出力は表形式の XML ファイルとした．また，表形式の XML ファイルとマッピング定義で記される XML との相互変換のロジックについては，汎用な XML 変換技術である XSLT [8] のスタイルシート (以下単に XSL) を用いて実施することとした．xmlExcelHyper は，あらかじめ，マッピング定義に対応する XSL を自動生成し，ファイルの入出力時にこれを VBA から呼び出す.

スプレッドシート上のマッピング定義はツールの製作時にのみ必要なものであり，原理的には，Excel 上から消去してもツールとしては動作させるようにできる．しかし，まず開発した SIB2UI においては，ユーザに対し非表示とするようツール側で対処することとした．逆に, このようにしておくことで, ユーザであっても, xmlExcelHyper の仕組みを知っていれば，設計ツールに，任意の設計情報を追加できるものとなる.



(1) マッピング定義から生成した出力用 XSL を用い，Excel から出力した表形式の XML をマッピング定義で規定する XML に変換する．(Export)
(2) マッピング定義から生成した入力用 XSL を用い，マッピング定義で規定する XML を表形式の XML に変換し，Excel へ入力する．(Import)
(3) マッピング定義を出力する．また，マッピング定義から XML 及びスプレッドシートのスキーマを抽出する．(スキーマ出力)
(4) マッピング定義に基づき，シート上の入力値をチェックする．(Validate)
(5) マッピング定義に基づき，階層を表す罫線描画を支援する．(Edit Support)

図 5　xmlExcelHyper の機能・実装イメージ

xmlExcelHyper の初版の開発は 2010 年度に終了し，SIB2UI も 2011 年 11 月のリリースにおいてこれを利用する実装に置き換えられた．開発した xmlExcelHyper は ISAS の衛星開発以外にも適用可能であり，web ページ [d] から一般に対し配布されている．SIB2UI への xmlExcelHyper の組み込みにおいては，特に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．ただし，構想段階で企画したものと出来上がった xmlExcelHyper の差異として，出力する XML スキーマは独自な記法のものとなったことがあげられる．これは，xmlExcelHyper の開発にかけた時間的な制約によるものであり，独自な記法は，原理的に RELAX NG に変換可能である．

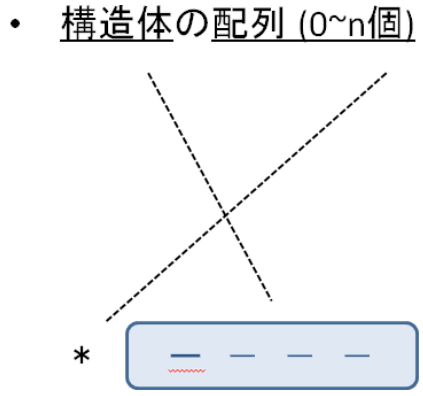
xmlExcelHyper は SIB2UI の 4 つの機能 (Edit Support, Import, Export, Validate) のいずれにおいても，その実現を担う．ただし，xmlExcelHyper が担うのは複雑なデータ構造の中においても，SIB2UI に頻繁に登場したパターンの処理に限られる．これを超える場合は，SIB2UI において外付けのプログラムを作成し，SIB2UI 全体の機能を実現している．SIB2UI・xmlExcelHyper の開発では，今回 xmlExcelHyper のサポート範囲に含めなかった範囲においても，今後，ライブラリ化の対象となりうる箇所も洗い出された．次章では，xmlExcelHyper が取り扱うデータ構造と XML 構造や，これらの間の対応を記述するマッピング定義の記法，SIB2UI を実現する上で洗い出された課題と複雑なデータを扱う機能のうち xmlExcelHyper で実現されたもの，されていないものについて述べることとする．

5 xmlExcelHyper が扱うデータ構造・XML 構造

xmlExcelHyper は，3 種類のデータ構造 : 単純テーブル (5.1)，選択テーブル (5.2)，階層テーブル (5.3) を基本とする．また，これらのテーブルに親子関係を規定し連結 (5.4) することができる．本章では，これらを順に示し，最後 (5.5) に xmlExcelHyper が対応しない表現方法について示す．

5.1 単純テーブル

xmlExcelHyper を用いて表現可能なデータ構造として最もシンプルな，単純テーブルと呼ばれるもののデータ構造とユーザインタフェース例をそれぞれ図 6 の (a) と (b) に示す．このデータ構造は複数のメンバを持つ構造体の 0~n 個の配列 (つまり表) からなるものである．SIB2UI は，この単純テーブルを 15 個有している．また，以降に示す選択テーブル，階層テーブルも含め，約 40 種の構造体を扱っている．



(a)

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 attributeLimitt シートより)

(b)

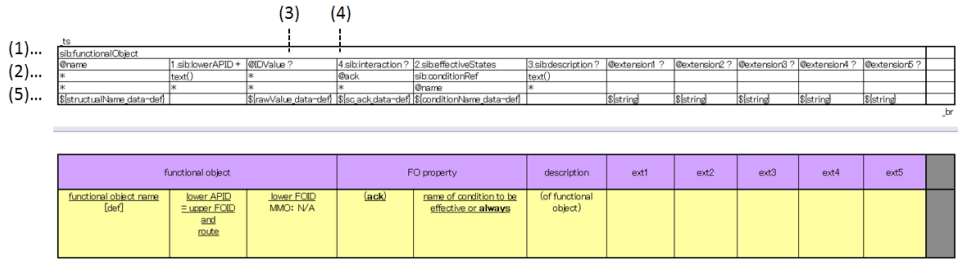
図 6 単純テーブルのデータ構造，ユーザインタフェース例

図 7 に，単純テーブルにおける，マッピング定義と対応する表のラベルの例を示す．単純テーブルの定義は，構造体と XML との対応の記述を一つ含む．ここで，構造体と XML との対応 (以下，構造体定義) は‘_ts’と記述したセルの下のセルを左上，‘_br’と記述したセルの上のセルを右下とする範囲に記述する．図 7 の例では，見やすさのためマッピング定義に罫線を描画してあり，構造体定義の範囲は太線の囲みで示される．ただし，罫線は xmlExcelHyper の処理には用いられない．

図 7 の構造体定義は 12 列からなり，右端の列はセルが空欄のみである．空欄のみの列は，データの表現には用いない．

d) http://www.c-soda.isas.jaxa.jp/software/

残りの 11 個の列が，それぞれ，構造体のメンバの定義となる．それぞれのメンバはテーブル定義の下に並べられた表のラベルで示される内容を保持する．テーブル定義の最下行 (5) には，構造体メンバの値に制約がある場合，その名称を ${ } で囲み記述する．それぞれの名称に対応する制約は RELAX NG のデータ型定義を記述したファイルを別途作成し，xmlExcelHyper に読み込ませる (図 15 参照)．なお，制約が無い場合は空欄とする．



(SIB2UI functionalObject シートより)

図 7　単純テーブルのマッピング定義 (上段) と対応する表のラベル (下段) の例

それぞれの構造体メンバをどの XML 要素・属性に対応させるか，また要素・属性の親子関係は，(1) ~ (2) に示される最下行より上の行に記述する．この記法は図 4 の (a) と同様 (ただし，左右ではなく上下) に，要素・属性の親子関係を可視化したものである．それぞれのセルには，XML のノードを指し示す汎用な技術である xpath[9] の記法を用いる．つまり，要素の場合は要素名を記載する，属性の場合は属性名に @ を冠し記載する．また，テキストノードの場合 text() と記載する．図 7 の例は，最上位行 (1) に示す sib:functionalObject 要素が，中間行 (2) に示す name 属性，sib:lowerAPID 要素，IDValue 属性，sib:interaction 要素，sib:effectiveStates 要素，sib:description 要素，extension1 から extension5 属性を有すこと，また，sib:lowerAPID 要素及び sib:description 要素がいずれもテキストノードを有すること，さらに，sib:effectiveStates 要素が sib:conditionRef 要素を有し，sib:conditionRef 要素が name 属性を有することを表している．なお，空欄のセル，‘*’ のみが記述されたセルは，それぞれの列の定義において無視され，XML 上の構造としての意味はもたない．また，最下行 (5) は，XML 上の属性もしくは末端の要素の値に対応する．

要素もしくは属性の多重度は (3) に示すよう, 正規表現の記号, すなわち, 0 か 1 (optional) の場合 ‘?’, 0 以上 (zeroOrMore) の場合 ‘*’，1 以上 (oneOrMore) の場合 ‘+’ を xpath の記法の右に付加することで記述する．なお，属性については XML のデータモデルに合わせ ‘?’ のみが指定可能である．一方，要素では zeroOrMore もしくは oneOrMore の指定も可能である．zeroOrMore もしくは oneOrMore を指定した場合，スプレッドシート上のセルの値を空白区切りで分解し，分解されたもの一つ一つを要素の値に対応させることを意図している．しかし，このような繰り返しの指定は SIB2UI において 2 個しか登場しないため，現状は不完全な実装 e) に留めている．この実装において xmlExcelHyper は一つのセルを一つの要素に対応させる．そこで，SIB2UI では，要素を分割・集約する処理を個別実装している (後述の個別対応項目 3 に含まれる).

スプレッドシート上の並び順と XML の要素の並び順が一致するとは限らない．スプレッドシート上の並び順と異なる順序で XML の要素を並べる場合, (4) で示すよう, xpath の記法 (つまり要素名) の左に N. の形式で並び順を表記する．図 7 の例では，sib:functionalObject 要素の子として，順に sib:lowerAPID 要素，sib:effectiveStates 要素，sib:description 要素，sib:interaction 要素が並ぶこととなる．

データ型定義を用いると値がとりうる制約として，「N 個の値から選択する」という表記も可能である．この場合，Excel 上で，その選択肢をプルダウンメニューから選べるようにすることが望ましい．しかし，現状の xmlExcelHyper は対応しておらず，SIB2UI での個別実装となっている (個別対応項目 1) .

xmlExcelHyper では，XML の属性・要素が optional もしくは zeroOrMore の多重度が指定されていて，これに対応セルが空欄の場合，これらの属性・要素は XML に出力しない．さらに，この多重度を XML 上の親子関係において複数回指定した場合, スプレッドシート上のデータ構造と XML との関係が不定となることがある. 例えば, 表 4 のケースにおいて, XML の構造として親子関係にある fatalCondition 要素，minParam 要素のいずれも optional に指定するとする．このケースで，値 1 が存在しない場合 (ア) として扱うべきなのか (イ) として扱うべきなのか仕様の設定に判断が必要となる．

e) 現状の実装では ‘*’ の指定は ‘?’ の指定として扱われる．‘+’ の指定は無指定として扱われる．

この判断が不要となるように xmlExcelHyper がサポートするマッピング定義では属性と末端の要素のみに多重度の指定が可能という制約を設けた．つまり，表 4 の例において (ア) のみを扱うこととした．

表 4 Optional の扱い

値がある場合

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

値が無い場合

(ア)

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

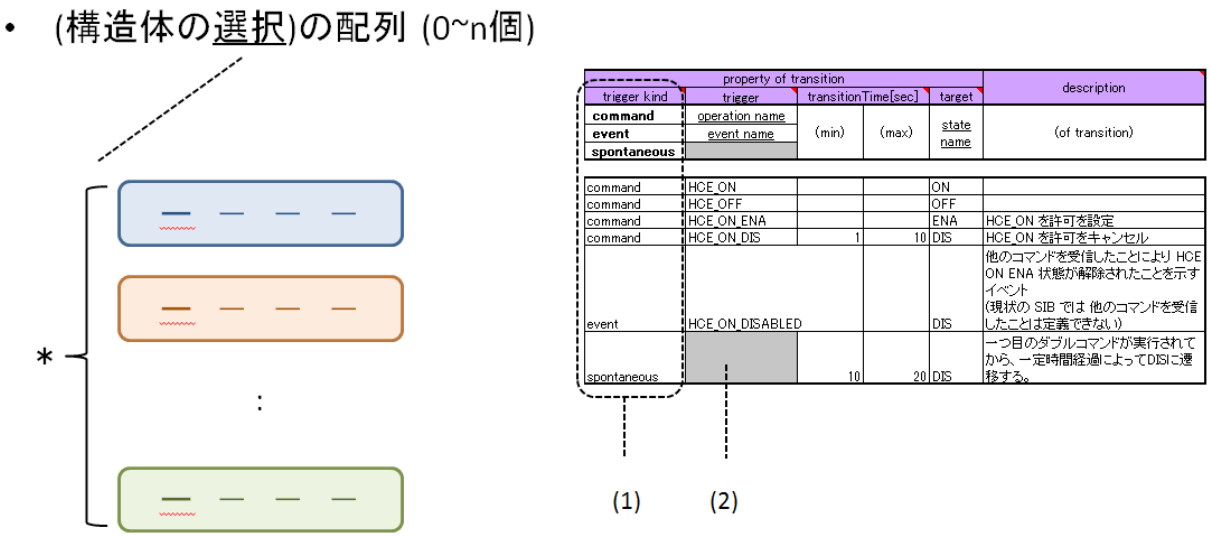
(イ)

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

5.2 選択テーブル

xmlExcelHyper を用いて表現可能なデータ構造として 1 段複雑な，選択テーブルと呼ばれるもののデータ構造とユーザインタフェース例をそれぞれ図 8 の (a) と (b) に示す．このデータ構造では，複数の構造体を任意の順で並べ表を作ることができる．いずれの構造体を選んだかを示すため，表には構造体の種別を示す列 (kind 列) を設けることとする．この方法を用いると，行ごとに表す対象が変わることとなり，行ごとに入力可能な列が異なることとなる．

選択テーブルのユーザインタフェースにおいて，xmlExcelHyper はマッピング定義に従い，kind 列入力用のプルダウンメニューを生成する．また，この列の入力時に，それぞれの構造体に応じ入力不能なセルに入力が不能であることを示すなどのため色付けの処理を行う．SIB2UI はこの選択テーブルを 7 個 (遷移のトリガ，条件・イベント種別，条件式の演算子 (4 重)，アトリビュートの変化則) 有している．



(1) kind 列 : 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
(2) 構造体によりフィールドの数や意味が異なるため，行ごとにどの列が入力可能か異なる

図 8 選択テーブルのデータ構造，ユーザインタフェース例

図 9 に選択テーブルのマッピング定義と対応する表のラベルの例を示す．一つのテーブル定義は‘_switch_tl’と記述したセルの下のセルを左上，‘_switch_br’と記述したセルの上のセルを右下とする範囲である．選択テーブルの定義には 5.1 で述べた構造体定義を上下に複数積み重ねたものを含む．図 9 の例には，太線の囲みで示す 3 つの構造体定義が含まれている．これに加え，kind 列を‘_case’の記述により指定する．kind 列に含まれ，値の右端に‘:’と記したセルには，スプレッドシート上で構造体の種別を表す文字列を指定する．このセルは XML 上のデータ構造としては意味を持たない．構造体定義では最下段のセルに背景色を指定する．この背景色はユーザが kind 列を設定した際に，その行のセルの背景色として設定される．

_switch_tl					
_case	_ts				
	sib:transition[descendant::sib:operationRef]				
	1.sib:Trigger	2.sib:transitionTimeCondition		@target	3.sib:description ?
	sib:operationRef	sib:minParam ?	sib:maxParam ?	*	text()
	@name	text()	text()	*	*
command:	${structualName_data-def}	${timeValue_data-def}	${timeValue_data-def}	${transitionTarget_data-def}	
	_ts				_br
	sib:transition[descendant::sib:eventRef]				
	1.sib:Trigger	2.sib:transitionTimeCondition		@target	3.sib:description ?
	sib:eventRef	sib:minParam ?	sib:maxParam ?	*	text()
	@name	text()	text()	*	*
event:	${structualName_data-def}	${timeValue_data-def}	${timeValue_data-def}	${transitionTarget_data-def}	
	_ts				_br
	sib:transition[not(descendant::sib:eventRef) and not(descendant::sib:operationRef)]				
	1.sib:Trigger	2.sib:transitionTimeCondition		@target	3.sib:description ?
		1.sib:minParam ?	2.sib:maxParam ?	*	text()
		text()	text()	*	*
spontaneous:		${timeValue_data-def}	${timeValue_data-def}	${transitionTarget_data-def}	
					_br
					_switch_br

property of transition					description
trigger kind	trigger	transitionTime[sec]		target	
command	operation name	(min)	(max)	state name	(of transition)
event	event name				
spontaneous					

(SIB2UI stateMachine シートより)

図 9　選択テーブルのマッピング定義 (上段) と対応する表のラベル (下段) の例

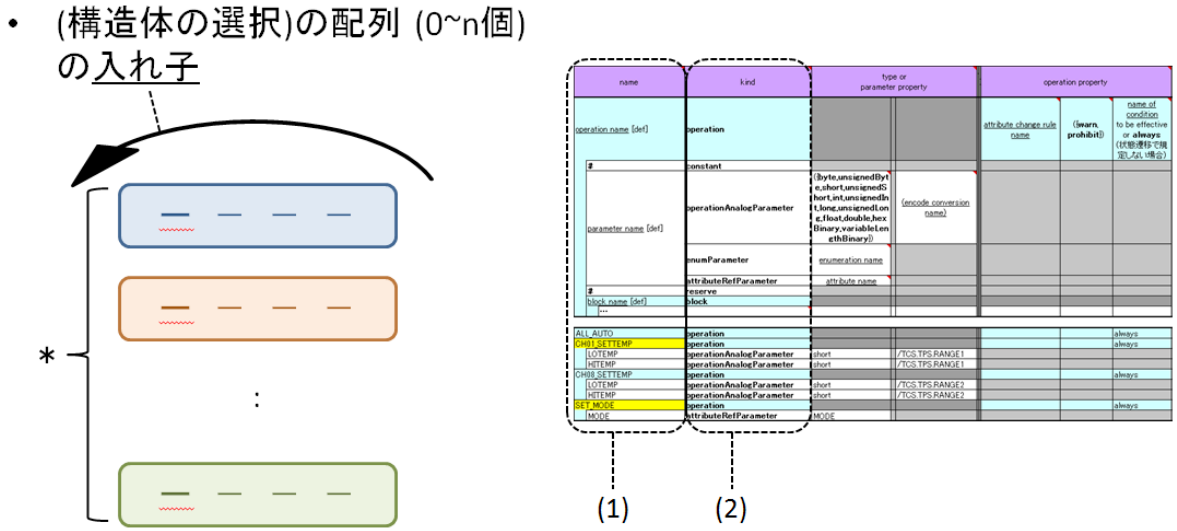
xmlExcelHyper のデータモデルでは，構造体の中身が構造体の登場箇所によらず，同じものである．他方で，xmlExcelHyper では，異なる構造体を同一の XML 要素に対応させることができることとした．表 5 に示す例では，異なる構造体を同一の XML 要素 <A/> に対応させている．この場合，<A/> 要素がいずれの構造体に対応するかは，子要素の違いを見なければ判断できない．そこで，XML の読み込みにおいて注意が必要となる．この種の判断を一般的に行うロジックは複雑なものとなるため，xmlExcelHyper では，冗長な情報ではあるものの，ツールの設計者において，それぞれ，A[B1] あるいは A[C1] などと，XML 読み込みの判定の際のヒントを xpath の述語の記法を用い，マッピング定義の要素名の指定の箇所へ記載することとした．この記述法は，図 9 の例においても，sib:transition 要素の指定で用いられている．

表 5　異なる構造体の同じ要素名へのマッピング

(a) <A> 　<B1> 値 </B1> 　<B2> 値 </B2> </A>
(b) <A> 　<C1> 値 </C1> 　<C2> 値 </C2> </A>

5.3 階層テーブル

xmlExcelHyper を用いて表現可能なデータ構造としてさらに複雑な，階層テーブルと呼ばれるもののデータ構造とユーザインタフェース例をそれぞれ図 10 の (a) と (b) に示す．このデータ構造では，複数の構造体で任意の入れ子を組むと共に並べて表を作ることができる．いずれの構造体を選んだかを示すため，表に kind 列を設けることは選択テーブルと同様である．階層テーブルにおいては，親子関係を，階層的な段差により示し，同一種別の構造体の入れ子も可能となっている．なお，この段差を用い階層を示す列を階層化部と呼ぶ．xmlExcelHyper はこの階層関係が分かり易いよう可視化するため，罫線を描画する機能を持つ．SIB2UI は，この階層テーブルを 3 個 (機能オブジェクトの子，アトリビュートシーケンス，データ変換) 有している．



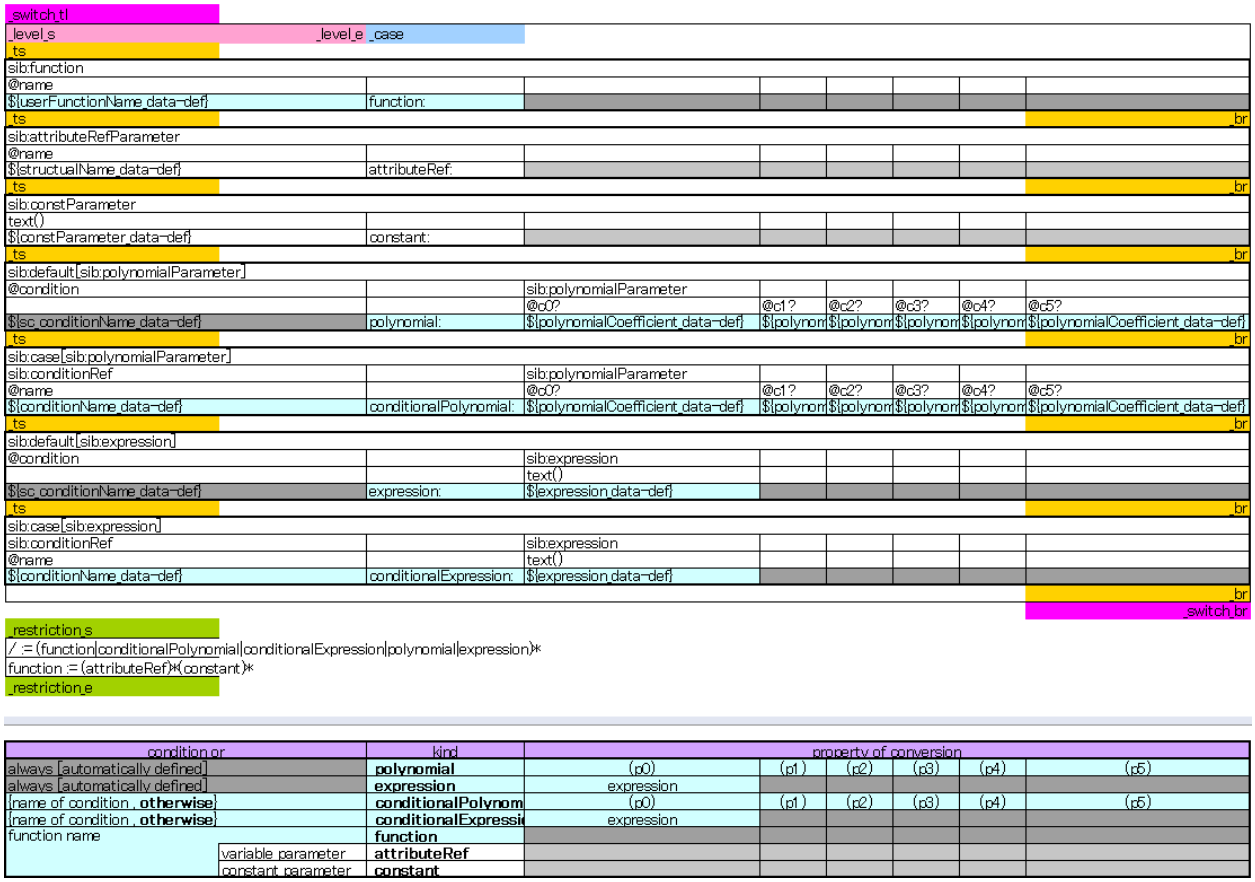
(SIB2UI functionalObject シートより)

(a) (b)

(1) 階層化部 : 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)
(2) kind 列 : 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

図 10 階層テーブルのデータ構造，ユーザインタフェース例

図 11 に階層テーブルのマッピング定義と対応する表のラベルの例を示す．階層テーブルの定義は，選択テーブルの定義に対し，階層化部の指定を加えたものである．階層テーブルの定義は，選択テーブルと同様に，複数の構造体定義と kind 列の指定を含んでいる．例では，7 個の構造体定義を含み，左から 3 列目に kind 列が指定されている．これに加え，階層テーブルでは，階層化部を‘_level_s’を記述した列から‘_level_e’を記述した列の範囲に記述する．例では，左端の 2 列が階層化部に指定されている．階層化部のマッピング定義のうち，構造体メンバと XML の対応は左端の列にのみ記述し，他の列は空欄にしておく．



_switch_tl								
_level_s		_level_e _case						
_ts								
sib:function								
@name								
${userFunctionName_data-def}	function:							
_ts								_br
sib:attributeRefParameter								
@name								
${structualName_data-def}	attributeRef:							
_ts								_br
sib:constParameter								
text()								
${constParameter_data-def}	constant:							
_ts								_br
sib:default[sib:polynomialParameter]								
@condition		sib:polynomialParameter						
		@c0?	@c1?	@c2?	@c3?	@c4?	@c5?	
${sc_conditionName_data-def}	polynomial:	${polynomialCoefficient_data-def}	${polynom	${polynom	${polynom	${polynom	${polynomialCoefficient_data-def}	
_ts								_br
sib:case[sib:polynomialParameter]								
sib:conditionRef		sib:polynomialParameter						
@name		@c0?	@c1?	@c2?	@c3?	@c4?	@c5?	
${conditionName_data-def}	conditionalPolynomial:	${polynomialCoefficient_data-def}	${polynom	${polynom	${polynom	${polynom	${polynomialCoefficient_data-def}	
_ts								_br
sib:default[sib:expression]								
@condition		sib:expression						
		text()						
${sc_conditionName_data-def}	expression:	${expression_data-def}						
_ts								_br
sib:case[sib:expression]								
sib:conditionRef		sib:expression						
@name		text()						
${conditionName_data-def}	conditionalExpression:	${expression_data-def}						
								_br
								_switch_br

_restriction_s
/ := (function|conditionalPolynomial|conditionalExpression|polynomial|expression)*
function := (attributeRef)*(constant)*
_restriction_e

condition or		kind	property of conversion					
always [automatically defined]		**polynomial**	(p0)	(p1)	(p2)	(p3)	(p4)	(p5)
always [automatically defined]		**expression**	expression					
{name of condition , **otherwise**}		**conditionalPolynom**	(p0)	(p1)	(p2)	(p3)	(p4)	(p5)
{name of condition , **otherwise**}		**conditionalExpressi**	expression					
function name		**function**						
	variable parameter	**attributeRef**						
	constant parameter	**constant**						

(SIB2UI conversion シートより)

図 11　階層テーブルのマッピング定義 (上段) と対応する表のラベル (下段) の例

XML 自身は，要素の無限な入れ子が可能である．他方で，階層テーブルにおいては，スプレッドシート上で視覚的に階層を表現するため, 階層数に上限を設ける必要がある. xmlExcelHyper は, 任意の上限値を許容する. SIB2UI においては, 機能オブジェクトとアトリビュートシーケンスに対しては 9 段，データ変換に対しては 2 段と上限を設定した．ここで，機能オブジェクトとアトリビュートシーケンスにおいては，任意の入れ子があり得るので実用的に十分な段数を設定した．また，データ変換に対しては，原理的な最大値を設定した．なお，データ変換に関しては 5.4 に示すテーブルの連結を用いても表現可能であるが，表の幅がコンパクトになるよう階層テーブルを用いた表現を採用した．

各種のデータ構造においては，構造体の親子関係，並び順に，制約がありうる．そこで，マッピング定義には，表 6 の例に示すように，ある構造体に対し，子の構造体として許される並び順を正規表現にて示すことで，制約を記述することとした．この記述は，図 11 の例に示されるようにマッピング定義の‘_restriction_s’と書かれたセルの下のセル，‘_restriction_e’と書かれたセルの上のセル，これらセルの間のセルに記述する．

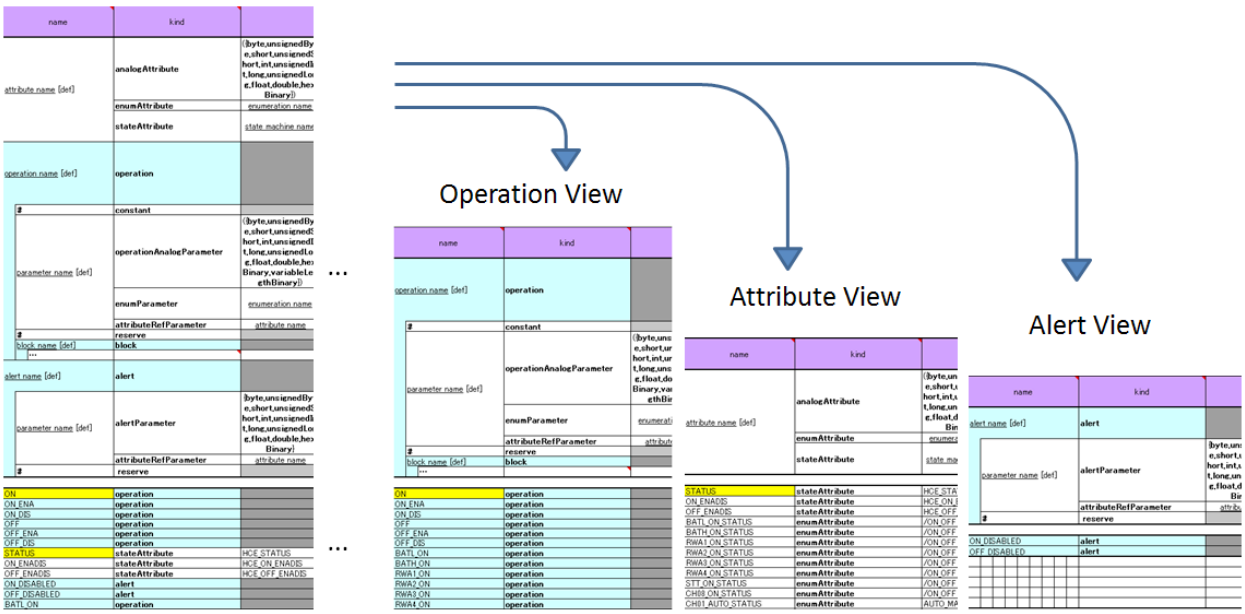
表 6　制約の記述例

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

この記法により，xmlExcelHyper が表現可能な XML 構造は，XML のスキーマの一つである，RELAX NG と比べ
- Mixed content (要素が，要素とテキストノードの双方持つ) は対象としない
- オプショナルと指定した要素・属性は値が空とならない
- 任意の要素名，属性名の登場を許容しない
- Interleave を許容しない
- 要素の入れ子の段数に上限がある

サブセットとなる．この種の表現が不能であっても，XML をデータの記述に用いる上で大きな支障とはならない．一方で，大きな階層化テーブルは必ずしも，把握しやすいものではないので，よりわかり易く表現する方法が必要である．

複雑な階層テーブルをシンプルに見せる手法のひとつとしてビューを設ける方法があげられる．SIB2UI においては，図 12 に示すように，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し，設計をより入力・把握しやすくする機能を設けている．この種の要求は，一般的な機能と考えられるが今のところ xmlExcelHyper には実装されておらず，SIB2UI での個別実装となっている (個別対応項目 2) ．次節では，階層テーブルを用いずに構造体の親子関係を示す方法を示す．

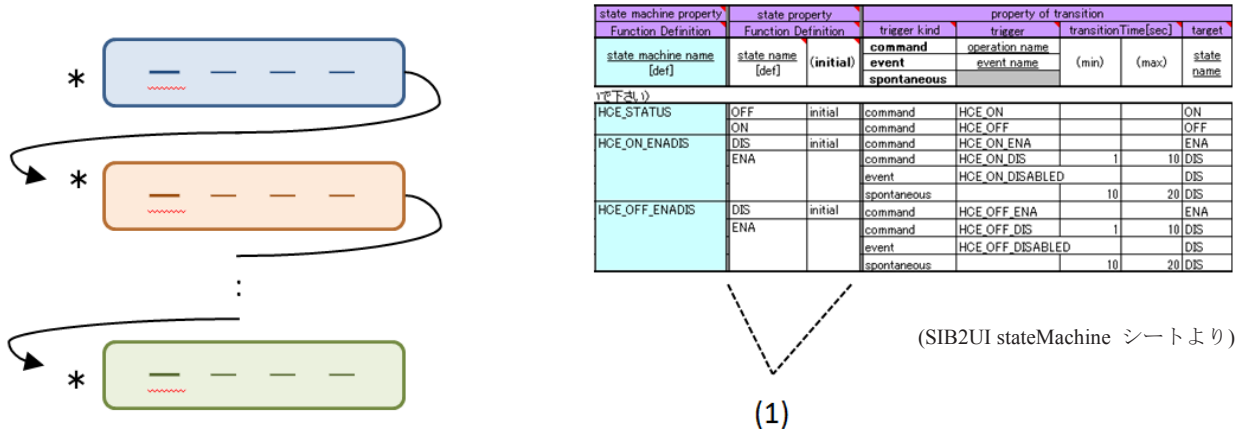


(SIB2UI functionalObject シートより)

図 12 ビューによる表示情報の制限

5.4 テーブルの連結

親子関係の可視性を高めるために，xmlExcelHyper では，単純テーブル，選択テーブル，階層テーブルを左右に連結することでも親子関係を表現することとした．図 13 の (a) に連結したテーブルのデータ構造の例，(b) に単純テーブル二つと選択テーブル一つを連結させたユーザインタフェース例を示す．xmlExcelHyper は，この親子関係がわかり易いように罫線を描画し，またセルを結合する機能を持つ．



state machine property		state property		property of transition				
Function Definition		Function Definition		trigger kind	trigger	transitionTime[sec]		target
state machine name [def]		state name [def]	(initial)	command event spontaneous	operation name event name	(min)	(max)	state name
いで下さい)								
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

(SIB2UI stateMachine シートより)

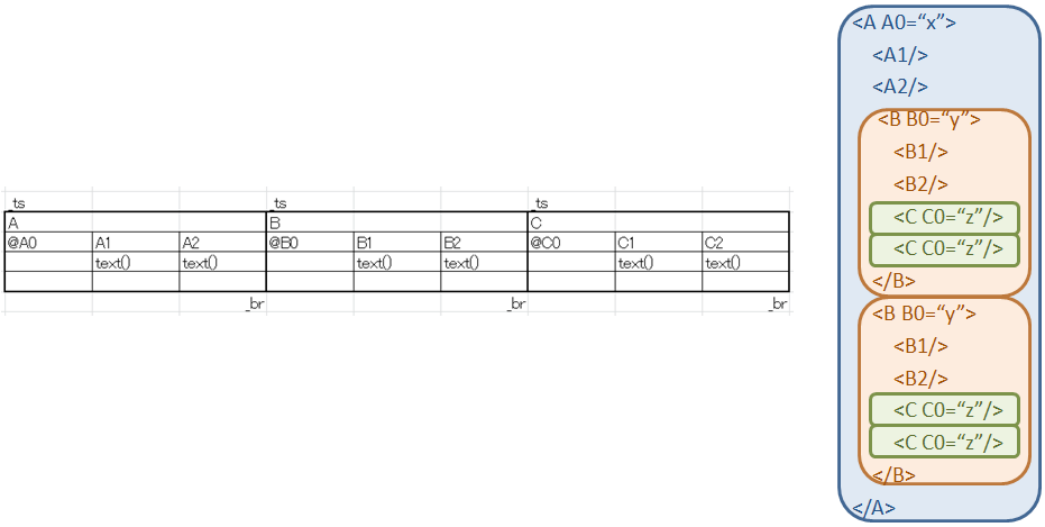
(1)

(a)　(b)

(1)　左が親，右が子となるように構造体の定義を並べる

図 13　連結したテーブルのデータ構造，ユーザインタフェース例

図 14 にこの連結された表を xmlExcelHyper が如何に XML に対応させるかの例を示す．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 を付加する必要がある．



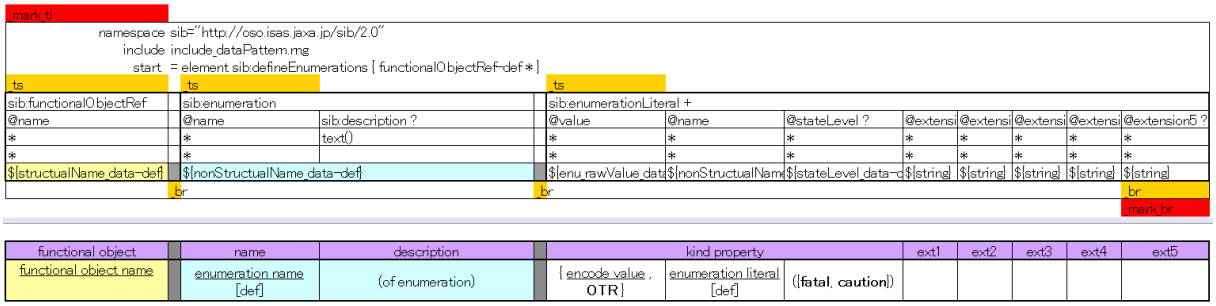
ts			ts			ts		
A			B			C		
@A0	A1	A2	@B0	B1	B2	@C0	C1	C2
	text()	text()		text()	text()		text()	text()
		_br			_br			_br

```
<A A0="x">
  <A1/>
  <A2/>
  <B B0="y">
    <B1/>
    <B2/>
    <C C0="z"/>
    <C C0="z"/>
  </B>
  <B B0="y">
    <B1/>
    <B2/>
    <C C0="z"/>
    <C C0="z"/>
  </B>
</A>
```

(a)　(b)

図 14　連結された表のマッピング定義と XML での表現の例

図 15 にテーブルの連結のマッピング定義と対応する表のラベルの例を示す．この例は，本論文において，マッピング定義の全体を示す唯一の例である．マッピング定義には 5.1~5.3 で述べた，複数のテーブル定義を左右に並べ，含める．図 15 の例には 3 つの単純テーブルの定義が含まれている．マッピング定義全体は‘_mark_tl’と書かれたセルの下のセルを左上，‘_mark_br’と書かれたセルの上のセルを右下とする範囲に記述される．マッピング定義全体では，左上に XML 名前空間の定義 (‘namespace’)，読み込むデータ型の定義 (‘include’)，XML のルート要素の指定 (‘start’) を行う．



(SIB2UI enumeration シートより)

図 15 テーブルの連結のマッピング定義 (上段) と対応する表のラベル (下段) の例

原理的に，階層テーブルは連結の任意の個所に登場して構わない．しかし，親子関係が簡単に把握できるよう SIB2UI では，階層テーブルは最も右にのみ登場することとし，xmlExcelHyper にも同様の制約を課すこととした．

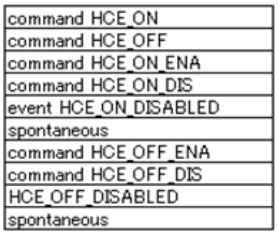
5.5 xmlExcelHyper では対応できない表現方法

本節では, SIB2UI で用いている, あるいは, 設計の際に議論されたデータの表現方法のうち, xmlExcelHyper ではカバーされていないものを示す．まず, 5.5.1 でセル内のデータ構造の表現について述べる．次に, 5.5.2 で参照関係を用いたデータ構造の分割について述べる．

5.5.1 セル内のデータ構造の表現

図 16 に，セル内のデータ構造の表現方法のうち，xmlExcelHyper の機能ではカバーされていないものの例を示す．(a) では，セル内で構造体や繰り返しを表現する．この例では，空白やカンマ区切りなどで繰り返しを表現するとともに，項目を区切り字 (‘:’ など) で区切り，構造体のメンバを表現している．(b) では，セル内で，表現する構造体を選択するための文法を導入している．(c) では，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している．これらのうち, (a) (b) は SIB2UI でも利用されており, SIB2UI 側で外付けの XSL を作成し, 処理が行われている (個別対応項目 3) ．ただし，これらの表現方法も，セルの値が妥当かのみならば，データ型ライブラリの文字列型に正規表現の制約を適用することで，xmlExcelHyper のみで判定が可能となっている．

(a)



(b)

図 16 対応していないデータ構造の表現 (セル内)

5.5.2　参照関係を用いたデータ構造の分割

一般にデータ構造においては 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．親子のあるデータ構造においては，以下の参照関係がありうる．

A) 親 (1) に ID を定義し，子 (n) から参照する．

B) 子 (n) に ID を定義し，親 (1) から参照する．

親子のデータ構造を スプレッドシート上で，ID を用いた参照関係で表現するか，階層テーブルあるいはテーブルの連結で表現するかはシート内外の表現方法として多様性がある．また，XML 上で，ID を用いた参照関係で表現するか，要素の入れ子で表現するかにも，XML ファイル内外の表現方法として多様性がある．また，これら，スプレッドシート上と XML 上の表現方法は，各々，独立に決めることができる．xmlExcelHyper はこれらのデータ構造のいずれをサポートする機能ももたない．

SIB2UI は，複数のシートの各々を XML ファイルに対応させる．単純化のため，SIB2UI では基本的に，シート間の ID を用いた参照関係を XML ファイル間の参照関係に対応させている．xmlExcelHyper はこのデータ構造を扱わないため，SIB2UI では，参照整合性の確認を外付けのプログラムで実施している (個別対応項目 4) ．また，シート間にデータ構造が分割されている場合，ユーザが設計を入力する際には，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 (個別対応項目 5) が有用である．さらに，ユースケースに応じ，シート間の情報を集約したビューで見せる機能 (個別対応項目 6) も有用である．SIB2UI においては，これら個別対応項目 5 と 6 を，外付けの別プログラム SIB2Viewer にて実現している．SIB2UI においては management シートを除く 8 つのシートの間に，A) の参照関係が 7 個，B) の参照関係が 7 個，計 14 個と多くの関係が張られており，これらが個別対応項目 4 と 5 の個別プログラミングの対象となっている．そこで，これらの項目は，今後，xmlExcelHyper におけるライブラリ化の対象として優先度が高いと考えられる．

SIB2UI においても，例外的に，スプレッドシート上と XML 上で異なるデータ構造の表現を採用した個所がある．ステートマシンの XML ファイルにおいては，UML の XML 表現である XMI と構造が近くなるよう表現方法を設定したため，遷移と状態の間で ID を用いた参照関係を張ることとした．他方で，Excel 上での表現においては，設計のしやすさを考慮し，遷移の開始状態を元に，状態と親子関係を設定しテーブルの連結を用いて表現することとした．これらのデータ構造の変換は，外付けの XSL で行なっている (個別対応項目 7) .

6 まとめ

複雑なデータ構造を持つデータベース (SIB2) に対し，Excel 上で内容を記述し，XML に入出力するツール SIB2UI を開発した．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，両者の対応がいっぺんに視認でき，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．また, 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汎用ライブラリ xmlExcelHyper 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 が扱うデータ構造として，SIB2UI が扱うデータ構造を分析し，特に頻繁に登場するパターン，構造体を扱う 3 種類のテーブル (単純テーブル，選択テーブル，階層テーブル) を含めた．SIB2UI においては，この構造体が約 40 個登場する．また，単純テーブルが 15 個，選択テーブルが 7 個，階層テーブルが 3 個登場する．これらのうち最も表現範囲が広い階層テーブルを用いると，扱える XML は，RELAX NG スキーマよりは狭いものの，有限な階層という制約の下において，正規文法に沿った任意の XML を規定できる．ただし，階層テーブルは，そのままでは可視性に難がある．そこで，データ構造の入れ子関係の可視性を改善するためテーブルを連結できることとした．しかし，この方法を用いると，適用可能な XML は限定的なものとなる．そこで，現状の xmlExcelHyper が威力を発揮するのは，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ケースと考えられる．

SIB2UI では，xmlExcelHyper には含めなかった幾つかの処理 (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，6: 見やすいビューを表示する機能，7: 参照を用いたデータ構造の表現方法の変換) を，現状ではアプリケーション (SIB2UI) 側で個別対応としている．これらのうち，項目 4 と 5 の対象であるデータ構造，すなわちシート間の参照関係については SIB2UI においては多数 (8 つのシートの間に，14 の参照関係) 張られており，今後の xmlExcelHyper サポート範囲の拡充として期待される．

謝辞

xmlExcelHyper の設計・製作は, 富士通株式会社, 富士通エフ・アイ・ピー・システムズ株式会社によって実施されました．山地尋之さん，上妻巧哉さん，川田耕一郎さん，中野華奈さん，境野淑介さん，田中孝明，広瀬智紀さんほかに感謝の意を表します．

参考文献

1) 馬場肇，松崎恵一，福田盛介，山田隆弘，飯塚祐介，山地尋之，大石克己 : SIB2/GSTOS1 の開発，第 11 回宇宙科学シンポジウム，2011 年 1 月
2) Yamada, T.: Standardization of Spacecraft and Ground Systems Based on a Spacecraft Functional Model, SpaceOps 2008, Heidelberg, Germany, May 2008.
3) Yamada, T. and Matsuzaki, K.: Model-Based Development of Spacecraft Onboard Functions, International Council on Systems Engineering (INCOSE) International Symposium 2010, July 2010, Chicago USA.
4) Takahiro Yamada: Functional Model of Spacecraft (FMS) DRAFT, Issue 0.9, March 2011
5) Takahiro Yamada: Spacecraft Monitor and Control Protocol (SMCP) DRAFT, Issue 0.13, 10 August 2011
6) 山田隆弘，松崎恵一：衛星情報ベース２定義（案) 第 0.9 版，2009 年 11 月 11 日
7) RELAX NG Specification, Committee Specification 3 December 2001, Ed. James Clark and MURATA Makoto, The Organization for the Advancement of Structured Information Standards 2001 (https://www.oasis-open.org/committees/relax-ng/spec-20011203.html)
8) XSL Transformations (XSLT) Version 2.0, W3C Recommendation 23 January 2007 (http://www.w3.org/TR/xslt20/)
9) XML Path Language (XPath) 2.0 (Second Edition), W3C Recommendation 14 December 2010 (Link errors corrected 3 January 2011) (http://www.w3.org/TR/xpath20/)