

**TTC標準**  
Standard

J T - M 3 0 3 1

tML スキーマの実装適合性宣言ひな型の  
ためのガイドライン

Guidelines for Implementation Conformance  
Statement proformas for tML Schemas

第 1 版

2006 年 6 月 1 日制定

社団法人  
情報通信技術委員会

THE TELECOMMUNICATION TECHNOLOGY COMMITTEE



本書は、(社)情報通信技術委員会が著作権を保有しています。

内容の一部又は全部を(社)情報通信技術委員会の許諾を得ることなく複製、転載、改変、転用及びネットワーク上での送信、配布を行うことを禁止します。

## 目次

<参考> .....	3
1. 範囲 .....	5
2. 参照 .....	5
3. 定義 .....	5
3.1 実装適合性宣言の定義 .....	5
4. 略称と略語 .....	5
5. tMLスキーマのICSひな型 .....	5
5.1 スキーマ概要 .....	6
5.2 SICSひな型の仕様のためのガイドライン .....	6
5.2.1 SICSひな型仕様のための一般的注意事項 .....	7
5.2.2 要素支援ひな型 (Element support proforma) .....	8
5.2.3 型支援ひな型 (Type support proforma) .....	8
5.2.4 属性支援ひな型 (Attribute support proforma) .....	9
5.2.5 識別制約支援ひな型 (Identity constraint support proforma) .....	10
5.2.6 複合型置換支援ひな型 (Complex type replacement support proforma) .....	10
5.2.7 代替グループ支援ひな型 (Substitution group support proforma) .....	11
6. SICS ひな型を埋める指定 .....	11
6.1 “Element Support Table” .....	11
6.2 “Type support Table” .....	11
6.3 “Status” カラム .....	12
6.4 “Support” カラム .....	13
6.5 “Index” カラム .....	13
6.6 “Subindex” カラム .....	14
6.7 “Identifier” カラム .....	14
6.8 “Content model” カラム .....	14
6.9 “Type/Content model” カラム .....	15
6.10 “Type” カラム .....	15
6.11 “Category” カラム .....	15
6.12 “Constraints” カラム .....	15
6.13 “Values” カラム .....	15
6.14 “Additional information” カラム .....	16
6.15 “Use” カラム .....	16
6.16 “Order” カラム .....	16
6.17 “Nillable” カラム .....	16
6.18 “Abstract” カラム .....	16
6.19 “Selector” カラム .....	16
6.20 “Filed” カラム .....	17
付属資料A .....	18

<参考>

1. 国際勧告等との関連

本標準は、ITU-T 勧告 M.3031 (05/2004), Corrigendum 1 (11/2005) に準拠したものである。

2. 上記国際勧告等に対する追加項目等

2.1 オプション選択項目

なし。

2.2 ナショナルマター項目

なし。

2.4 原勧告と章立ての構成の相違

なし。

3. 改版の履歴

版 数	制 定 日	改 版 内 容
第1版	2006年6月1日	制 定

4. 工業所有権

本標準に関わる「工業所有権等の実施の権利に係る確認書」の提出状況は、TTCホームページでご覧になります。

5. その他

(1) 参照している勧告・標準等

ITU-T 勧告 : M.3013(2000)

ISO 標準 : ISO/IEC10646-1:2000, ISO/IEC 11179-3:1994, ISO/IEC 11179-5:1995

W3C 勧告 : eXtensible Markup Language (XML) 1.0 (Second Edition) (2000),

XML Schema Part 1: Structures (2001), XML Schema Part 2: Datatypes (2001),

Namespaces in XML (1999)

IETF 仕様 : RFC2141 (1997), RFC2396 (1998)

OMG 仕様 : Unified Modeling Language Specification, Version 1.4 (2001)

Committee T1 標準 : T1.227-2000

TTC 標準 : JT-M3010 (2001)

(2) ITU-T 勧告 M.3031 (05/2004) の誤記について

ITU-T 勧告 M.3031 (05/2004) 6.7 節 f)

For the “Subscription group support table”, this field shall be entered the name of the element that is declared as a substitution group of the specified element

この文章に対して本標準では、“Subscription group support table”を“Substitution group support table”の誤記として扱っている。

ITU-T 勧告 M. 3031 (05/2004) 6.10 節

The completing instructions follow the same rules as described in item b of Section 6.9 for simple type.

この文章に対して本標準では、item b of Section 6.9 を item b of Section 6.7 の誤記として扱っている。

## 1. 範囲

この勧告は、tML ベースのインタフェース情報仕様のためのひな型である、スキーマ実装適合性宣言 (SICS) を定義するためのガイドラインを提供する。スキーマ実装適合性宣言 (SICS) は、tML スキーマ定義のためのひな型に実装内容がどの様に適合しているかを定義する。

## 2. 参照

以下の ITU-T 勧告と他の参考文献の中で参照しているテキストは、この勧告の一部となる。発行時点では、これらの版は有効であるが、全ての勧告とその他の参考文献は改版の可能性がある。このことから、この勧告を利用する者は、下記の勧告とその他の参考文献の最新版が適用可能かどうか確認することを勧める。現在有効な ITU-T 勧告のリストは定期的に公表されている。

[1] ITU-T X.291 (1995), *OSI conformance testing methodology and framework for protocol Recommendations for ITU-T applications – Abstract test suite specification*

[2] ITU-T X.296 (1995), *OSI conformance testing methodology and framework for protocol Recommendations for ITU-T applications – Implementation conformance statements.*

[3] ITU-T X.724 (1996), *Information technology – Open Systems Interconnection – Structure of management information: Requirements and guidelines for implementation conformance statement proformas associated with OSI management.*

[4] TTC JT-M3030, テレコミュニケーションマークアップ言語 (tML) フレームワーク

[5] W3C, *eXtensible Markup Language (XML) 1.0 (Second Edition)*, 6 October 2000.

[6] W3C, *XML Schema Part 0: Primer*, 2 May 2001.

[7] W3C, *XML Schema Part 1: Structures*, 2 May 2001.

[8] W3C, *XML Schema Part 2: Datatypes*, 2 May 2001.

[9] W3C, *Namespaces in XML*, 14 January 1999.

[10] W3C, *XML Path Language (XPath) Version 1.0*, 16 November 1999.

## 3. 定義

### 3.1 実装適合性宣言の定義

この勧告は、ITU-T の勧告 X. 296[2]にて定義した下記の用語を使用している。

- a) (ICS) 項目;
- b) (ICS) 課題;
- c) 状況 (価値);
- d) (支援) 回答

## 4. 略称と略語

この勧告では、下記の略語を用いる：

ICS	実装適合性宣言
ITU-T	国際電気通信連合・電気通信標準化セクター
SICS	スキーマ実装適合性宣言
tML	電気通信マークアップ言語
W3C	ワールド・ワイド・ウェブ・コンソーシアム
XML	拡張マークアップ言語

## 5. tML スキーマの ICS ひな型

tML スキーマは、tML ベースのインタフェース情報を定義するためのテンプレートとして使われる。SICS ひな型は、スキーマ要素と属性といった tML スキーマ機能を表さなくてはならない。この節では、これらの

言語機能に基づく、tML スキーマの機能を導入し SICS ひな型を定義する。

### 5.1 スキーマ概要

tML スキーマ[4]は、人によって作成されたルールをコンピュータで実行できるための、共通の語彙を表現している。これらは、tML ドキュメントの構造、中身、意味を定義する手段を提供し、ある程度の正確性をもって、ある要素の出現回数（例えば、その頻度）を定義する。この要素は属性、子テキスト、部分要素を持つ場合がある。部分要素の並べ方の順番は、もし要素が部分要素を含む場合には考慮されなければならない。目標とする名前空間[9]と既定のオプションの定義が含まれる事もある。tML のスキーマはオブジェクトの複製と同等である要素や属性の参照を利用する。

スキーマの文法仕様[6]、[7]、[8]に従い、これらの構成の包含関係を図 5-1 に示す。

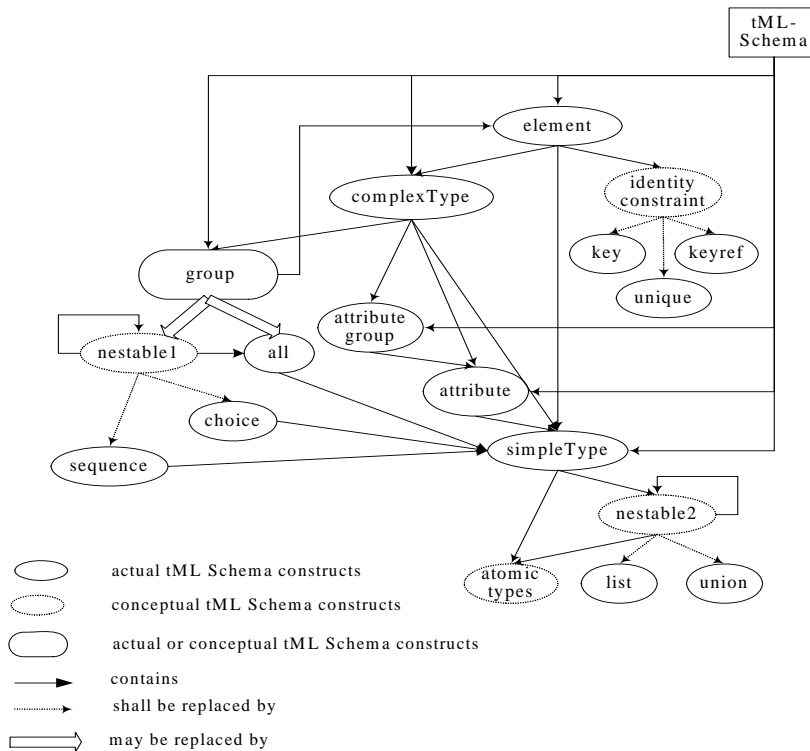


図 5-1/JT-M3031 tML スキーマ仕様の包含関係  
(ITU-T M. 3031)

注記 1—上記の図は、適合宣言に関連するスキーマ構成を説明している。スキーマ適合性とは何の関係もない、例えば注釈の様な構成は、この図で示していない。概念的な tML スキーマ構成は、現実の tML スキーマ構成に常時置き換えることができる。“group”は、具体的スキーマ定義に従い、現実的か概念的のどちらかの tML スキーマ構成と考えられることができる。すなわち、“group”は、個々の構成としての面と、他の現実的なスキーマ構成を置き換えられるか、どちらかになる。

注記 2—全ての tML 組込の単純型は W3C XML スキーマ第 0 部：入門書[6]の 2.3 節に示す通りであるが、図 5-1 に示す“atomic types”に置き換える事ができる。ICS ひな型は、上記に示す関係を表さなくてはならない。

### 5.2 SICS ひな型の仕様のためのガイドライン

ひな型仕様は、以下の節で記述するスタイルに従わなければならない。ひな型仕様は、この勧告によって

必要とされた情報を提供しなければならない。必要な場合は他の情報のために表が追加される。

SICS に関して 3 つのレベルのドキュメンテーションがある。すなわち：

- a) SICS ひな型の実装のための道具である、ガイドラインあるいは勧告；
- b) tML ベースのネットワーク管理と連携する SICS ひな型。これは、実装の供給者によって入力される。完成された SICS ひな型は、SICS である；
- c) tML ベースのネットワーク管理に関する標準に適合要求の一部として、実装の供給者によって用意した SICS。

この勧告の範囲では、項目 b のためのガイドラインだけが提供される。

#### 5.2.1 SICS ひな型仕様のための一般的注意事項

この勧告は、tML ベースの管理インタフェース ICS ひな型仕様を構成するための指示を提供する。SICS ひな型は、六つのサブのひな型を含む：要素支援ひな型、型支援ひな型、属性支援ひな型、識別制約支援ひな型、複合型置換支援ひな型、代替グループ支援ひな型である。これらの全てのひな型は、ITU-T 勧告 X.724[3] で指定されたひな型と同様に表形式となっている。

5.2.2 節は、要素の SICS ひな型について述べる。5.2.3 節は、型の SICS ひな型について述べる。5.2.4 節は、属性の SICS ひな型について述べる。5.2.5 節は、識別制約の SICS ひな型について述べる。5.2.6 節は、SICS ひな型を複合型の置き換えを説明する。5.2.7 節は、代替グループの SICS ひな型について述べる。付属資料 A は、実装の供給者によって入力される SICS ひな型仕様の例を示す。

ITU-T 勧告 X.291[1] と X.296[2] で定義された以下の共通の記法を、この勧告の状態値カラムにも適用している：

- m 必須
- o オプション
- c 条件付き
- x 除外（“x” は “excluded” の略）
- 適用範囲外

注記 1—記法 ‘c’、‘m’、‘o’ と ‘x’ は、その表の条件付き項目或いは選択項目と関連が有る場合は、“c:” の文字が前に付く。

注記 2—条件付きを意味する記法 ‘c’ は、特定の条件を示すために数字が後に付く（例えば、c1、c3）。異なる条件に対応して異なる番号が割り振られ、これらの条件は ICS ひな型ドキュメンテーションの中で明確に記述されている。

注記 3—記法 ‘o’ には、状態値の中で互いに排他的、或いは選択的なオプションを示すために、“.n” が付けられる（‘n’ は重複しない番号）。この番号付けされたセットに関する要求は、該当の表の脚注やその他の場所で明確に記述されている。

下記の ITU-T の X.291[1] と ITU-T の X.296[2] で定義されている共通の表記方法は、回答カラムに使われる：

- Y : 実装された
- N : 実装されなかった
- : 答え不要
- Ig : 項目を無視（統一的処理を行うためであり、意味は持たない）

SICS ひな型仕様は、インタフェース仕様の要求を満たすために、5.2.2、5.2.3、5.2.4、5.2.5、5.2.6 と 5.2.7 の各節を写し、“Support” と “Additional information” カラムを除いた表を完成し、残りの表を拡張する事により形成される。SICS ひな型には、tML ベースのインタフェース情報モデルで定義されている、全てのインスタンス化可能な要素と、それらがサポートする型と属性のための表を提供しなければならない。これらの型は、W3C XML スキーマ第 0 部：入門書[6]の 4.5 節に示す様な、多様な方法を使って定義されるこ



とができる。

SICS ひな型から SICS を形成するために、実装の供給者は SICS ひな型の全ての表の “Support” と、必要であれば “Additional information” カラムを記入しなければならない。

### 5.2.2 要素支援ひな型 (Element support proforma)

要素のひな型の目的は、実装の供給者が標準の形で適合性情報を提供するためにスキーマ要素定義が適合していると主張した場合のメカニズムを提供することである。

要素支援ひな型を表 5-1 に示す。

表 5-1/JT-M3031 Element support table  
(ITU-T M. 3031)

Index	Identifier	Content model	Values	Nilable	Status	Support	Additional information

ここで、

- “Index” カラムは、tMLベースのインタフェース情報モデルで定義された各インスタンス化可能なルート要素に対して、一意の数字が割り当てられる。このインデックスは、他の表でも参照される。
- “Identifier” カラムは、要素の名前を明らかにするために使われる。
- “Content model” カラムは、要素の内容モデルを示し、“empty” か “any” か “mixed” か “children” のどれかをとる。詳細は6.8節を参照。“Content model” はそれ以降の支援表に展開される。
- “Values” カラムは、もしtMLスキーマで定義されていれば、この要素の固定またはデフォルト値の識別に用いられる。詳細に関しては6.13節を参照。
- “Nilable” カラムは、この要素の値が空か、そうでないかを識別するために用いる。このカラムは、“true” か “false” のどちらかを取ることもある。
- “Status” カラムは、この要素に対する適合要求を明らかにするために用いる。それぞれインスタンス化できる非抽象要素について、もしそれが必須であれば、“Status” カラムは ‘m’ となる；もしそれがオプションであれば、このカラムは ‘o’ となる（特定の条件の下では、“.n” が追加される。詳細は、5.2.1節を参照）；もしインスタンスtMLドキュメントにおいて、それが特定の条件で発生するのであれば、このカラムには、‘c’ とそれに続く一意な数字が入力される；もしそれがインスタンス化可能でなければ、そのための表は無くなる。
- “Additional information” カラムは、実装者のこの要素の実装に関わる情報に使われる。

### 5.2.3 型支援ひな型 (Type support proforma)

型のひな型の目的は、標準形式で適合情報を提供するために要素のスキーマ型定義に適合していると主張している実装の供給者にメカニズムを提供することである。この表は、スキーマの要素が “complexType” または “simpleType” の場合での要素支援表の拡張形態である。この表は、その表中で入れ子になって登録されている複合型とサブ要素/サブ型の型情報に従って拡張される。

この型支援ひな型を表 5-2 に示す。

表 5-2/JT-M3031 Type support table  
(ITU-T M.3031)

Index	Identifier	Type / Content model	Order	Nilable	Abstract	Constraints	Values	Status	Support	Additional information

ここで、

- “Index” は、要素の中の型に関する一意的な参照となるか、複合型の中のサブ要素の一意的な参照である。
- “Identifier” フィールドは、複合型に含まれている型名かサブ要素を識別するために使われる。

- “Type/Content model” フィールドは、要素の型名または、サブ要素の単純型のカテゴリを識別するために用いる。このカラムが単純型の場合は、この部分には単純型の基本となる型が記入され、もしこの単純型が基本型の拡張であるならば “union” または “list” が記入される。この場合は、単純型の制限に関する詳細記述を行うために更にカラムが拡張される。もし、この複合型を示す場合は、この部分には “complexType” か、グループを参照する場合は “group” が入る。もしこのカラムがサブ要素を示す場合は、この部分には、要素型の型名が入る。詳細は4.9節を参照。
- “Order” フィールドは、“sequence”、“choice”、“all”のどれかをとる複合型の内容モデルの中に現れる可能性のあるサブ要素を指定する事を目的として使用する。この部分はカラムが複合型である場合のみ有効となる。詳細は6.16節を参照。
- “Nillable” フィールドは、このカラムがサブ要素を示す場合に、サブ要素の値が空か、そうでないかを指定するために使用する。
- “Abstract” フィールドは、複合型か或いはサブ要素が抽象か否か、を指定するために使用する。
- “Constraints” フィールドは、対応する表の項目に関する可能性のある制約を指定するために使用される。制約は、3つの種類に分けられ、複合型の制約、サブ要素の制約、そして単純型の制約である。複合型の制約には、“block” または “-” が適用され、サブ要素の制約には “minOccurs” と “maxOccurs” が適用され、単純型を持つ要素には全ての想定可能な様相が適用される。tMLスキーマで複数の様相が定義された場合は、それら全てはこの部分に適当な接頭語を伴って記入されなければならない。詳細に関しては6.12節を参照。
- “Values” フィールドは、（もしtMLスキーマに定義されていれば）列がサブ要素を示しているとき、固定のあるいはデフォルトの値を指定するために使用される。詳細は6.13節を参照。
- “Additional Information” フィールドは、実装者に対して型やサブ要素について実装の仕様情報を示すために使用される。

#### 5.2.4 属性支援ひな型 (Attribute support proforma)

属性のためのひな型の目的は、複合型のスキーマの属性の定義に対するコンフォーマンスを要求する実装側に対して、標準的な形式でコンフォーマンスの情報を提供するメカニズムを提供することである。このテーブルは、複合型かそれが含まれるサブ要素の何らかの複合型が定義された属性を持つ場合に、“Type support table” に対する拡張になる。このテーブルは複合型あるいはそれが入れ子になって含まれるサブ要素の型で定義される属性情報によって拡張される。

属性支援ひな型を表 5-3 に示す。これは属性を含む複合型ごとに適用される。

表 5-3/JT-M3031 Attribute support table  
(ITU-T M.3031)

Index	Subindex	Identifier	Type	Use	Constraints	Values	Status	Support	Additional information

ここで、

- “Index” は属性定義や宣言を持つ複合型に対して一意な参照である。
- “Subindex” フィールドは複合型のそれぞれの属性の参照を記述するものであり、対応する複合型のインデックスとデリミタ ‘.’ を持つ一意な番号をつないで作られる。
- “Identifier” フィールドは属性の名前を特定するために使用される。
- “Type” フィールドは、型名称と属性のベース型を特定するために使用される。属性は単純型であるため、このフィールドは5.2.3節の同じカラムで記述したように単純型に対して同じ意味を持つ。詳細は6.10節を参照。
- “Use” フィールドはインスタンスドキュメントの中の属性出現の制約を示すのに使われ、適用可能ならこの属性に対するtMLスキーマに特有の値を入れる。

- “Constraints” フィールドは属性の単純型に対して適用可能な様相を特定するために使われる。このフィールドは5.2.3節の同じカラムで記述された単純型に対して同じ意味を持つ。詳細は6.12節を参照。
- “Values” フィールドはこの属性がtMLスキーマで定義されていた場合は、属性の固定あるいはデフォルトの値を特定するために使用される。
- “Additional information” フィールドはこの属性について何らかの実装特有の情報を実装者が追加するためのスペースである。

#### 5.2.5 識別制約支援ひな型 (Identity constraint support proforma)

識別制約のひな型の目的は、スキーマの識別制約の定義に対する適合性を要求する実装側に対して、標準的な形式で適合性の情報を提供するメカニズムを提供することである。このテーブルは、要素と（あるいは）それが包含しているサブ要素が tML スキーマの中で定義された識別制約を持つ場合に、要素支援テーブルに対する拡張になる。このテーブルは要素と（あるいは）入れ子にされたサブ要素の識別情報によって拡張される。

識別制約支援ひな型を表 5-4 に示す。これは識別制約を含む要素ごとに適用される。

**表 5-4/JT-M3031 Identity constraint support table**  
(ITU-T M.3031)

Index	Identifier	Category	Selector	Field	Status	Support	Additional information

ここで、

- “index” は要素の中でのこの識別制約の一意的な参照である。
- ” Identifier” フィールドは識別制約の名前を特定するために使われる。
- “Category” フィールドは一意的な識別の種類を特定するために使われ、” unique”、” key”、” keyref” のひとつをとる。詳細は6.11節を参照。
- “Selector” フィールドは一意的な識別に対して選択されたようその範囲を特定するために使われ、XPath表記<sup>1</sup>である。
- “Field” フィールドは選択されたようその組の範囲内で一意でなければならないそれぞれの選択された要素に関連する部分（属性または要素）を特定する。
- “Additional information” は識別制約について何らかの実装特有の情報を実装者が追加するためのスペースである。

#### 5.2.6 複合型置換支援ひな型 (Complex type replacement support proforma)

“complex type replacement” に対するひな型の目的は、スキーマの複合型置き換えの定義に対する適合性を要求する実装側に対して、標準的な形式で適合性の情報を提供するメカニズムを提供することである。要素の複合型がインスタンスドキュメントで導出される型によって置き換えられる場合に、このテーブルは型支援テーブルに対する規範となる付録である。このテーブルはこの複合型から導出されるすべての適切な型の情報によって拡張される。そして、このテーブルには、それぞれの項目の詳細を記述するために他の” Type support table” が続く。

複合型置き換え支援ひな型を表 5-5 に示す。

**表 5-5/JT-M3031 Complex type replacement support table**  
(ITU-T M.3031)

<sup>1</sup> XPathの概念は、XML Path Language1.0[10]で参照できる。

Index	Identifier	Derivation	Status	Support	Additional information

ここで、

- “Index” は normative annex の中で導出された型に対する一意な参照である。
- “Identifier” フィールドは、導かれた複合型の名称を示すために使われる。
- “Derivation” フィールドは、この型が元の複合型を拡張した方法を示すのに使われ、“extension” が “restriction” になる。
- “Additional information” は、複合型の置き換えについての何らかの実装特有の情報を実装者が追加するためのスペースを提供する。

#### 5.2.7 代替グループ支援ひな型 (Substitution group support proforma)

“Substitution group” に対するひな型の目的は、スキーマの代替グループの定義に対する適合性を要求する実装側に対して、標準的な形式で適合性の情報を提供するメカニズムを提供することである。インスタンスドキュメントの中で一つまたは複数の複合型のサブ要素がそれらの代替要素によって置き換えられる場合に、このテーブルは型支援テーブルに対する規範となる付録である。このテーブルはこの要素に対して代替グループとして定義されたすべてのありうる要素の情報によって拡張される。このテーブルには、それぞれの要素やその型に対する詳細を記述するために他の “Type support table” が続く。

代替グループ支援ひな型を表 5-6 に示す。

表 5-6/JT-M3031 Substitution group support table  
(ITU-T M.3031)

Index	Substitutable element	Status	Support	Additional information

ここで、

- “Index” はこの規範となる付録の中で導出される型に対する一意な参照である。
- “Substitutable element” フィールドはインスタンスドキュメントの中の要素に対しての代替であるようその名称を示すために使われる。
- “Additional information” は、実装者が代替グループについて何らかの実装特有の情報を追加するためのスペースを与える。

### 6. SICS ひな型を埋める指定

この章では 5.2 節やその中の説明で定義したそれぞれのカラムを埋めるための指定を示す。これらの指定は二つの節に分割できる；一つは ICS ひな型仕様を開発するための指定であり、もう一つはひな型を完成する実装のためである。

#### 6.1 “Element Support Table”

完全な tML ベースのインタフェース情報モデルでは、ひとつまたは複数のルート要素が存在する。それぞれのルート要素に対応して、ICS のひな型には “Element support table” がある。それぞれのテーブルには ICS のひな型ドキュメントの範囲内に一意なインデックスが割りあてられる。

#### 6.2 “Type support Table”

要素や型が直接あるいは間接的に子構造を入れ子にした定義を含んでいる場合、子構造の拡張は一度以上繰り返されることはない。代わりに、一度拡張した後は、参照は定義が拡張されたインデックスに対する “Additional Information” カラムに作られる。

### 6.3 “Status” カラム

このカラムは適合要件を特定し、特定の tML スキーマに対する適合性のために要求されるサポートレベルを示す。値は 5.2.1 節に定義されたものである。

図 6-1 は異なる適合要件に応じた包含関係と共に、tML スキーマの構成の状態を示している。

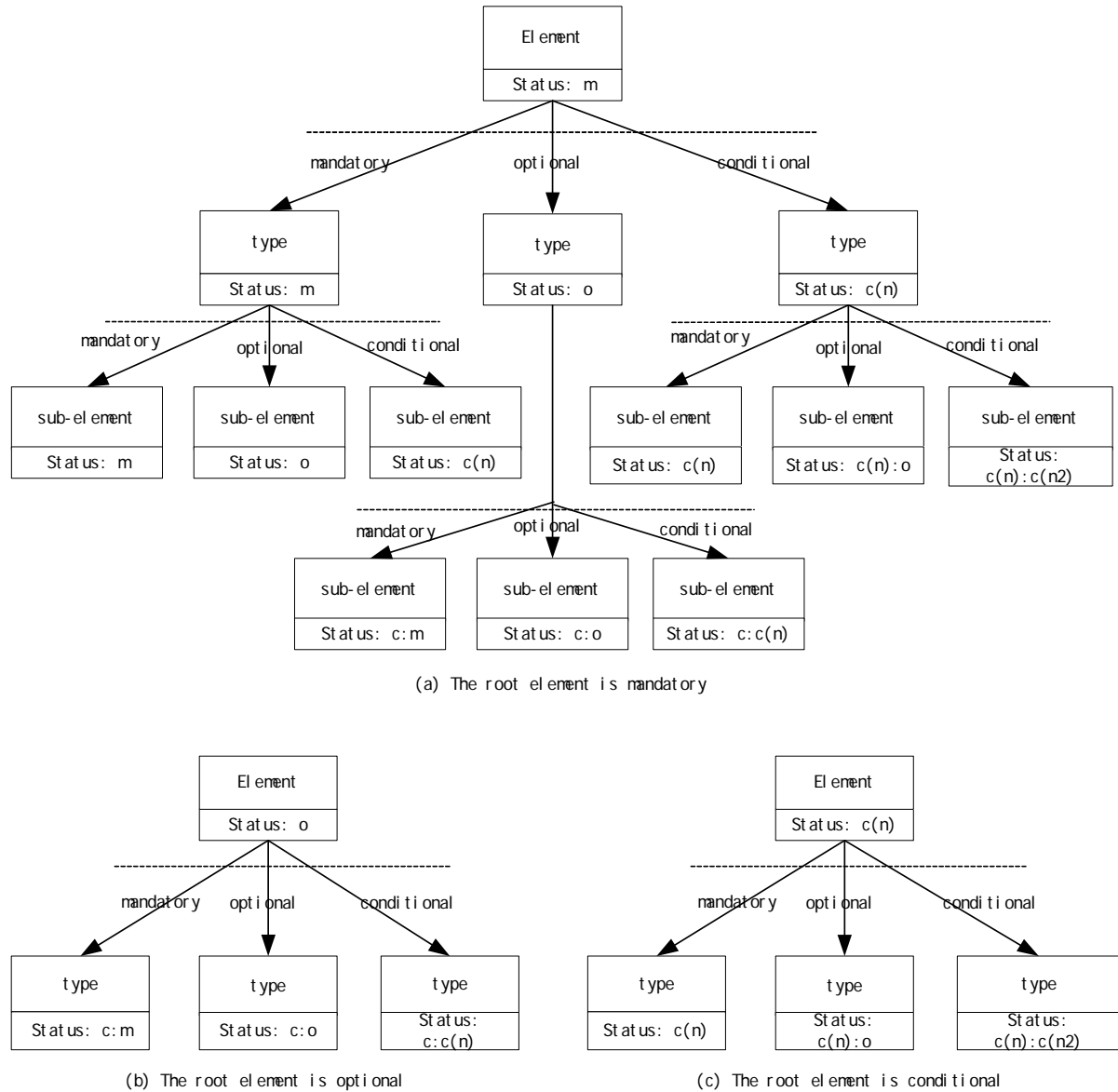


図 6-1/JT-M3031 包含構成の状態図  
(ITU-T M.3031)

このフィールドを埋めるガイドラインを以下に示す。

- a) “Element support table” に対しては、もしそれが必須なら、“Status” フィールドには ‘m’ が入る。それがオプションならこのフィールドには ‘o’ が入る。(ある特定な状況では “.n” という添え字が付く。詳細は 5.2.1 節を参照) もし、特定の状況の下でインスタンスドキュメントに現れるならば、このフィールドには ‘c’ に一意な番号を付けたものが入る。
- b) ” Type support table” に対しては、もしその列が型を表すなら、その項目の状態は図 6-1 に示される。このフィールドには、この型を持つ要素がオプションである場合を除いて” Element support

table”にある値と同じ値が入り、このフィールドには添え字として” c:” が付いた型の状態が入る。同じ規則がそれらの間の包含関係を持つ構造に対しても適用される。つまり、包含される構造の状態はその包含する構造の状態に依存する。もし、列がサブ要素を示すなら、その状態は図 6-1 に示した上に述べたものと同じ規則に従う。サブ要素が抽象的ならこのフィールドには’ x’ が入る。そうでなければ、包含する型の状態が’ m’ ならば、このフィールドは包含する型の値で始まる数より構成され、それ自身の適合要件を含む。包含する型の状態が’ m’ ならば、このフィールドにはそれ自身の適合要件が入る。複合型が置き換え可能なら、項目 f に書かれたものと同じ規則が適用される。

- c) ” Attribute support table” に対しては、規則を完全にすることはサブ要素に対して項目 b で述べたものと同じものに従う。その要件に対するデフォルトポリシーは、この属性に対する” Use” フィールドに” required” と入ったときに、その状態は必須になる。もし、” Use” フィールドに” optional” と入ると、その状態もまたオプションになる。” Use” フィールドに” prohibited” と入ると、その状態は禁止される。
- d) ” Identity constraint support table” に対しては、規則を完全にすることはサブ要素に対して項目 b で述べたものと同じものに従う。
- e) ” Complex type replacement support table” と” Substitution group support table” に対しては、もし、（” Element support table” や” Type support table” の中で）型や置き換えたり代入されたりする要素に対して特定の制約条件が何もないならこれらふたつのテーブルの” Status” フィールドに対するデフォルト値はオプションである。（ある特定の状況では、添え字に” .n” が付くかもしれない。詳細は 5.2.1 節を参照）置き換えられたり代入されたりする型や要素（サブ要素）が抽象的である場合、これら二つのテーブルの” Status” フィールドのデフォルト値は必須である。デフォルト値は tML ベースのインタフェース仕様の中に記述された実際の許可ないし禁止された使用法によって変わるかもしれない。
- f) ルート要素、サブ要素または複合型が置き換え可能または代入可能であるならその項目の” Status” カラムには’ o’ が入る。（特定の状況では、” .n” の添え字がつくかもしれない。詳細は 5.2.1 節を参照）

#### 6.4 “Support” カラム

このカラムは供給者や実装者がそれぞれの項目の実装のサポートレベルを示すために埋められる。使用可能なフィールドの選択は 5.2.1 節にリストされている。

もし、項目が必須なら、” Support” カラムには適合要件として’ Y’ が入る。そうでなければこのカラムは’ Y’ または’ N’ が入る。

#### 6.5 “Index” カラム

SICS ひな型の中のそれぞれの行は、行の左側の端で番号付けされている。この番号付けは、SICS ひな型のすべてのありえる実装詳細を一意に特定する手段として含まれている。

個々のレスポンスに対して参照する方法は次の順序で行われる。

- a) 項目の上位節(super-clause)に対する参照
- b) デリミタ文字 ‘.’
- c) 一意な番号

この表記の使用例は付属資料 A に示されている。

“Index” カラムは、ルート要素の範囲内の構成間の包含関係を示している。包含関係は 5.1 節：スキーマ概要に示されている。

## 6.6 “Subindex” カラム

このカラムは“Index”カラムと同じフォーマットである。これは”Index”フィールドの値で始まる数字で構成され、一つまたは二つのレベルが下がった階層を含んでいる。表 6-1 に示した例を見ると、”Index”フィールドの値は、”2”であり、”SubIndex”フィールドにはそれぞれ”2.2”と”2.2.1”が入っている。

表 6-2/JT-M3031 “SubIndex” カラムの使用例  
(ITU-T M.3031)

Index	Subindex	Identifier	Type	Use	Constraints	Values	Status	Support	Additional information
2	2.2	postcode	simpleType	Optional			o		
	2.2.1		string		length=6		c:m		

## 6.7 “Identifier” カラム

このカラムは固有のテーブル内の特定項目の名前や識別子を指定する。

- “Element support table” においては、このフィールドには要素の名前<sup>2</sup>が入らなければならない。
- “Type support table” においては、もし列が複合型や単純型を示すならば、このフィールドには型の名前が入らなければならない。もし列がサブ要素を示すならば、このフィールドにはサブ要素の名前が入らなければならない。
- “Attribute support table” においては、このフィールドには属性の名前が入らなければならない。
- “Identity constraint support table” においては、このフィールドには制約の名前が入らなければならない。制約の名前がないならば、空白のままとなる。
- “Complex type replacement support table” においては、このフィールドにはオリジナルの型に代わる複合型の名前が入らなければならない。
- “Substitution group support table” においては、このフィールドには特定の要素の代替グループとして宣言された要素の名前が入らなければならない。

## 6.8 “Content model” カラム

“Element support table” 中の “Content model” フィールドには、要素の内容モデルが入らなければならない。内容モデルの適切な値は、“empty”、“any”、“mixed”または“children”である。もし要素が空ならば、すなわち、要素は全く内容がないことになり、このフィールドには“empty”が入らなければならない。もし、要素の内容に何の制約もないならば、このフィールドには“any”が入らなければならない。もし文字データがサブ要素に並んで現れ、かつ、最も深いサブ要素に限定されないならば、このフィールドには“mixed”が入らなければならない。そうではなく、サブ要素が入っていれば、このフィールドには“children”が入らなければならない。

<sup>2</sup> 要素の名前は、一意性のために名前空間の情報を含むべきである。もし可能ならば、この規則は型やサブ要素の名前にも適用する。

## 6.9 “Type/Content model” カラム

“Type support table”（もし可能ならば、ひな型仕様に入れられるべき）の“Type/Content model”カラムは、特定項目の型や内容モデル情報を含んでいる。もし列が単純型を示していれば、このフィールドには単純型の基本型が入らなければならない。組込み型に加えて、このフィールドには、もしこの単純型が組込み基本型の一つの拡張であれば、括弧の中に base atomic 型を並んで持つ“union”や“list”も入る。もし列が複合型を示していれば、このフィールドには“complexType”が入らなければならない。もし列がグループを示していれば、このフィールドには“group”が入らなければならない。もし列がサブ要素（取りうる値は6.8節を参照）を示していれば、サブ要素の型が複合型の時は、このフィールドにはサブ要素の内容モデルが入り、サブ要素の型が単純型の時は、単純型の基本型が入らなければならない。

## 6.10 “Type” カラム

“Attribute support table”の“Type”カラムは、属性の型情報を含んでいる。このフィールドには、単純型の名前と属性の基本型が入り、単純型が名前付けされていない場合は、属性の基本型が入らなければならない。終了の指定は、6.7節の項目bで単純型のために説明されるのと同じ規則に従う。

## 6.11 “Category” カラム

“Identity constraint support table”の“Category”カラムは、特定項目の一意性の識別型情報のためのカテゴリを含む。この一意性の識別の値は、“unique”、“key”または“keyref”のどれか一つでなければならない。属性が、識別規則により“key”で示されている時は、その値は一意であり、nilに設定されることはできない。

## 6.12 “Constraints” カラム

テーブル（もし可能ならば、ひな型仕様に入れられるべき）の“Constraints”カラムは、特定項目の制約を含んでいる。

- a) “Attribute support table”においては、このフィールドには単純型のための適用可能な様相<sup>3</sup>の情報が入る。tMLスキーマに複数の様相が指定されているときは、全ての値が接頭語として関連する様相とともにこのフィールドの中にリストされるべきである。例えば、“maxInclusive=20”、“totalDigits=4”である。型が列挙型のときは、すべてのとりうる値がこのフィールドに、接頭語と共に括弧のなかにリストされなければならない。すなわち、“enum={one, two, three}”である。
- b) “Type support table”においては、制約は3種類、すなわち複合型に対する制約、サブ要素に対する制約、単純型に対する制約に区別される。列が単純型を示しているときは、終了の指定は項目aに記述された同じ規則に従う。列が複合型を示し、tMLスキーマの中で“block”属性がこの型に指定されていれば、“restriction”、“expansion”または“all”のいずれかである“block”属性の値は、接頭語“block=”とともにこのフィールドに入らなければならない。それ以外では、“-”が適用される。列がサブ要素を示しており、“minOccurs”かつ/または“maxOccurs”の値が定義されているときは、このフィールドには関連する接頭語とともにこれらの値が入らなければならない。もしサブ要素に単純型があれば、その型に対応可能な様相の制約は、項目aに指定された同じ規則に従って、サブユニットのこのフィールドにリストされる。付属資料Aでこの使用方法の例がある。

## 6.13 “Values” カラム

テーブル（もし可能ならばひな型仕様に入れるべき）の“Values”カラムは、特定の項目のために事前指定された値を含んでいる。このフィールドには、もしtMLスキーマに値が定義されているならば、単純型の要素または属性のために、“fixed”値または“default”値が入る。その値は、“fixed=”または“default=”として事前に決定される。この規則は“Element support table”、“Type support table”

---

<sup>3</sup> 全ての様相は、W3C XMLスキーマ第0部：入門書[6]の付属資料Bで参照できる。



そして “Attribute support table” に適用される。

#### 6.14 “Additional information” カラム

このカラムは、他のどのカラム含まれなかった供給者により提供された追加情報を含む。以下の場合は、tML スキーマに従って、その他の情報が入力されるものとする。

“Type Support table” において、もし列が、インスタンスドキュメントの中で導出される型によって置き換えることが可能な複合型を示している場合は、関連する “Complex type replacement support table” に対する参照をこのフィールドに入れられる。もし列が、インスタンスドキュメントの中で代替可能な要素によって置き換えることが可能な要素を示している場合は、関連する “Substitution group support table” に対する参照をこのフィールドに入れられなければならない。すなわち、“See Annex A x.x” である。もし列が、直接または間接的な子として自身を扱っている型または要素を示している場合は、テーブルはこの枝の拡張を止める。しかしながら、直接または間接的な親のインデックスを示して、情報はこのフィールドに入力されなければならない。

#### 6.15 “Use” カラム

“Attribute support table” 中の “Use” カラムの適切な値は、“required”、“optional”、そして “prohibited” である。もし属性が一度現れるのならば、このフィールドには “required” が入る。もし属性が一度あるいは全く現れないのならば、このフィールドには “optional” が入る。もし属性が現れないのならば、“prohibited” が入らなければならない。

#### 6.16 “Order” カラム

“Type support table” 中の “Order” カラムでの適切な値は、列が複合型を示している時は “sequence”、“choice”、そして “all” である。もし複合型で定義されたサブ要素の一つ以上が事前に定義されたオーダーで現れなければならない場合だと、このフィールドには “sequence” が入らなければならない。もし複合型の中にサブ要素が一つだけ、または一つのグループとしてインスタンスドキュメントに現れるならば、このフィールドには “choice” が入らなければならない。もし子が一度、あるいは全く現れない、または順不同で現れるかもしれない場合は、このフィールドには “all” が入らなければならない。

#### 6.17 “Nillable” カラム

“Nillable” カラムは、ある要素またはサブ要素の値が nil に設定できるか否かを指定するために使われる。列がサブ要素を示している “Element support table” または “Type support table” において、もし要素に nil 値がセットできるならば、“Nillable” フィールドには “true” が入らなければならない。そうでない場合は、このフィールドは常に “false” が入らなければならない。

#### 6.18 “Abstract” カラム

“Type support table” 中の “Abstract” カラムは、対応する複合型またはサブ要素が抽象であるかどうかを示す。もし複合型またはサブ要素が抽象であれば、“true” が入り、もしそうでなければ、“false” が入る。その他のケースでは、“-” が適用される。

#### 6.19 “Selector” カラム

“Identity constraint support table” 中の “Selector” カラムには、一意性を特徴づける要素の範囲のための XPath 表現が入る。XPath は、親を所有していないツリーにおける最初のサブ要素を除いた、親の直接のサブ要素のシーケンスで構成される。付属資料 A の使用例を参照。

## 6.20 “Filed” カラム

“Identity constraint support table”の中の”Field”カラムには、それぞれ選択された要素に関連する部品（属性または要素）のためのXPath表現が入る。属性または要素は、選択された要素の集まりの中で一意でなければならない。

## 付属資料 A

### SICS ひな型 仕様の例

#### A.1 tML スキーマ定義

この付属資料では、“Element support”、“Type support”、“Attribute support”、“Identity constrain support”、“Complex type replacement support”そして“Substitution group support”の SICS ひな型を記述している。以下は tML スキーマ定義の簡単な例である。

```
<?xml version="1.0" encoding="UTF-16"?>
<xsd:schema targetNamespace="urn:int.itu/tML/SICSExample"
xmlns:tML="urn:int.itu/tML/SICSExample"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
version="1.0">
  <xsd:annotation>
    <xsd:documentation xml:lang="en-GB">
      <tML:SICSExample>
        <tML:OriginalAuthor>Dong Wenli - BUPT China</tML:OriginalAuthor>
        <tML:CreationDate>05-04-2004</tML:CreationDate>
        <tML:Description>This Schema contains some example Schema definitions.
          This example doesn't include the whole tML Schema metadata provided
          in Annex A/M.3030, only because this is just an example showing some
          Schema features in order to display the corresponding SICS proformas
          in A.2. It is not necessary to include the whole tml Schema metadata
          in M.3030 for this purpose. No more annotation is provided in
          this Schema example. </tML:Description>
        <tML:Source>M.tMLconf</tML:Source>
        <tML:SchemaHistory/>
      </tML:SICSExample>
    </xsd:documentation>
  </xsd:annotation>
  <xsd:element name="externalGsmCell">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="tML:userLabel" maxOccurs="unbounded"/>
        <xsd:element ref="tML:absElement"/>
      </xsd:sequence>
    </xsd:complexType>
    <xsd:key name="empid">
      <xsd:selector xpath="tML:userLabel"/>
      <xsd:field xpath="@id"/>
    </xsd:key>
  </xsd:element>
  <xsd:element name="userLabel" type="tML:Usertype"/>
  <xsd:element name="absElement" type="xsd:string" abstract="true"/>
  <xsd:complexType name="Usertype">
    <xsd:sequence>
      <xsd:element ref="tML:name" minOccurs="0" maxOccurs="3"/>
      <xsd:group ref="tML:ddTop"/>
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:string" use="required" fixed="bj88"/>
  </xsd:complexType>
</xsd:schema>
```

```

<xsd:attribute name="contr" use="optional" default="66666">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:minLength value="5"/>
      <xsd:maxLength value="8"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:attribute>
</xsd:complexType>
<xsd:element name="name" nillable="true">
  <xsd:complexType>
    <xsd:choice>
      <xsd:element ref="tML:metar"/>
      <xsd:element ref="tML:name"/>
    </xsd:choice>
  </xsd:complexType>
</xsd:element>
<xsd:element name="metar" type="xsd:decimal"/>
<xsd:simpleType name="ListOfMyIntType">
  <xsd:list>
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="one"/>
        <xsd:enumeration value="two"/>
        <xsd:enumeration value="three"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:list>
</xsd:simpleType>
<xsd:group name="ddTop">
  <xsd:choice>
    <xsd:element name="ddc1">
      <xsd:simpleType>
        <xsd:restriction base="tML:ListOfMyIntType"/>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="ddc2">
      <xsd:simpleType>
        <xsd:restriction base="xsd:date"/>
      </xsd:simpleType>
    </xsd:element>
  </xsd:choice>
</xsd:group>
<xsd:element name="absElementSub" type="xsd:integer" substitutionGroup=
"tML:absElement"/>
<xsd:element name="namesub" type="xsd:string" substitutionGroup="tML:name"/>

```

```

<xsd:complexType name="AuserType">
  <xsd:complexContent>
    <xsd:extension base="tML:UserType">
      <xsd:attribute name="postcode" use="optional">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:length value="6"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:attribute>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="Busertype" block="restriction">
  <xsd:complexContent>
    <xsd:restriction base="tML:UserType">
      <xsd:sequence>
        <xsd:element ref="tML:name" minOccurs="0" maxOccurs="3"/>
      </xsd:sequence>
      <xsd:attribute name="id" type="xsd:string" use="required" fixed="bj88"/>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>
</xsd:schema>

```

A.2 SICS

上記のスキーマ定義から推測されるように、externalGsmCell だけがルート要素であり、この簡単なスキーマベースの情報モデルの SICS ひな型は、以下のように記述する事ができる。

E 1 : tML:externalGsmCell

Table 1 : Element support

Index	Identifier	Content model	Values	Nilable	Status	Support	Additional information
1	tML:externalGsmCell	children		False	m		

Table 1.1 : Type support

Index	Identifier	Type/Content model	Order	Nilable	Abstract	Constraints	Values	Status	Support	Additional information
1.1		complexType	sequence	-	false		-	m		
1.1.1	tML:userLabel	children	-	false	false	maxOccurs=unbounded		m		
1.1.1.1	tML:UserType	complexType	sequence	-	false		-	o		see Annex [A 1.1.1.1]
1.1.1.1.1	tML:name	children	-	true	false	minOccurs=0 maxOccurs=3		c:o		see Annex [A 1.1.1.1.1]
1.1.1.1.1.1		complexType	choice	-	false		-	c:m		
1.1.1.1.1.1.1	tML:metar1	decimal	-	false	false			c:o		
1.1.1.1.1.1.2	tML:name2	children	-	true	false			c:o		see Table 1.1.1.1.1
1.1.1.1.2	tML:ddTop	group	choice	-	-		-	c:m		
1.1.1.1.2.1	tML:ddc1	simple	-	false	false			c:o		
1.1.1.1.2.1.1		tML:ListOfMyIntType	-	-	-		-	c:m		
1.1.1.1.2.1.1.1	tML:ListOfMyIntType	list	-	-	-		-	c:m		
1.1.1.1.2.1.1.1.1		string	-	-	-	enum {one two three }	-	c:m		
1.1.1.1.2.2	tML:ddc2	date	-	false	false			c:o		
1.1.2	tML:absElement	string	-	false	true			x		see Annex [A 1.1.2]

Attribute support

Index	Subindex	Identifier	Type	Use	Constraints	Values	Status	Support	Additional information
-------	----------	------------	------	-----	-------------	--------	--------	---------	------------------------

1.1.1.1	1.1.1.1.3	id	string	required		fixed=bj88	c:m		
1.1.1.1	1.1.1.1.4	contr	simpleType	optional		default=66666	c:o		
	1.1.1.1.4.1		string		minLength=5 maxLength=8	-	c:m		

Table 1.2 : Identity constraint support

Index	Identifier	Category	Selector	Field	Status	Support	Additional information
1.2.1	empid	key	tML:userLabel	id	M		

Annex [A 1.1.1.1]

Table A 1.1.1.1 Complex type replacement support for "tML:Usertype"

Index	Identifier	Derivation	Status	Support	Additional information
1	tML:Busertype	restriction	O		
2	tML:Ausertype	extension	O		

Index	Identifier	Type/Content model	Order	Nilable	Abstract	Constraints	Values	Status	Support	Additional information
1.1	tML:Usertype	complexType	sequence	-	false		-	c:m		
1.1.1	tML:name	children	-	true	false	MinOccurs=0 maxOccurs=3		c:o		
1.1.1.1		complexType	choice	-	false		-	c:m		
1.1.1.1.1	tML:metar	decimal	-	false	false			c:o		
1.1.1.1.2	tML:name	children	-	true	false			c:o		see Table 1.1.1

Attribute support

Index	Subindex	Identifier	Type	Use	Constraints	Values	Status	support	Additional information
1.1	1.1.2	id	string	required		fixed=bj88	c:m		

Index	Identifier	Type/Content model	Order	Nilable	Abstract	Constraints	Values	Status	Support	Additional information
2.1	tML:Usertype	complexType	sequence	-	false		-	c:m		
2.1.1	tML:name	children	-	true	false	minOccurs= 0 maxOccurs= 3		c:o		
2.1.1.1		complexType	choice	-	false		-	c:m		
2.1.1.1.1	tML:metar	decimal	-	false	false			c:o		
2.1.1.1.2	tML:name	children	-	true	false			c:o		see Table 2.1.1

2.1.2	tML:ddTop	group	choice	-	false		-	c:m		
2.1.2.1	tML:ddc1	simple	-	false	false			c:o		
2.1.2.1.1		tML:ListOfMyIntType	-	-	-		-	c:m		
2.1.2.1.1.1	tML:ListOfMyIntType	list	-	-	-		-	c:m		
2.1.2.1.1.1.1		string	-	-	-	enum {one two three }	-	c:m		
2.1.2.2	tML:ddc2	date	-	false	false			c:o		

Attribute support

Index	Subindex	Identifier	Type	Use	Constraints	Values	Status	Support	Additional information
2	2.2	postcode	simpleType	optional			c:o		
	2.2.1		string		length=6		c:m		
2.1	2.1.3	id	string	required		fixed=bj88	c:m		
2.1	2.1.4	contr	simpleType	optional		default=66666	c:o		
	2.1.4.1		string		minLength=5 maxLength=8	-	c:m		

Annex [A 1.1.1.1.1 ]

Table A 1.1.1.1.1 Substitution group support for element "tML:name"

Index	Substitutable element	Status	Support	Additional information
1	tML:namesub	O		

Index	Identifier	Type/Content model	Order	Nilable	Abstract	Constraints	Values	Status	Support	Additional information
1	tML:namesub	string	-	false	false			o		

Annex [A 1.1.2]

Table A 1.1.2 Substitution group support for element "tML:absElement"

Index	Substitutable element	Status	Support	Additional information
1	tML:absElementSub	M		

Index	Identifier	Type/Content model	Order	Nilable	Abstract	Constraints	Values	Status	Support	Additional information
1	tML:absElementSub	integer	-	false	false			m		