INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

M.3030

(08/2002)

SERIES M: TMN AND NETWORK MAINTENANCE: INTERNATIONAL TRANSMISSION SYSTEMS, TELEPHONE CIRCUITS, TELEGRAPHY, FACSIMILE AND LEASED CIRCUITS

Telecommunications management network

Telecommunications Markup Language (tML) framework

ITU-T Recommendation M.3030

ITU-T M-SERIES RECOMMENDATIONS

TMN AND NETWORK MAINTENANCE: INTERNATIONAL TRANSMISSION SYSTEMS, TELEPHONE CIRCUITS, TELEGRAPHY, FACSIMILE AND LEASED CIRCUITS

Introduction and general principles of maintenance and maintenance organization	M.10–M.299
International transmission systems	M.300–M.559
International telephone circuits	M.560–M.759
Common channel signalling systems	M.760–M.799
International telegraph systems and phototelegraph transmission	M.800–M.899
International leased group and supergroup links	M.900–M.999
International leased circuits	M.1000–M.1099
Mobile telecommunication systems and services	M.1100–M.1199
International public telephone network	M.1200–M.1299
International data transmission systems	M.1300–M.1399
Designations and information exchange	M.1400–M.1999
International transport network	M.2000–M.2999
Telecommunications management network	**M.3000–M.3599**
Integrated services digital networks	M.3600–M.3999
Common channel signalling systems	M.4000–M.4999

For further details, please refer to the list of ITU-T Recommendations.

ITU-T Recommendation M.3030

Telecommunications Markup Language (tML) framework

Summary

This Recommendation is a framework containing rules, guidelines, and objectives for developing telecommunications industry standard telecommunications Markup Language (tML) schemas for operations, administration, maintenance and provisioning (OAM&P) applications. The tML language is based on the World Wide Web Consortium (W3C) XML 1.0 Recommendation. tML schemas are based on the W3C XML schema Recommendation. Also included in this framework are rules, objectives, and guidelines on procedures to develop new telecommunications vocabularies and to use/reuse existing vocabularies.

Source

ITU-T Recommendation M.3030 was prepared by ITU-T Study Group 4 (2001-2004) and approved under the WTSA Resolution 1 procedure on 22 August 2002.

Keywords

telecommunications Markup Language, eXtensible Markup Language, tML Schema.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

© ITU 2003

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

CONTENTS

Page

ITU-T Recommendation M.3030

Telecommunications Markup Language (tML) framework

1 Scope

This Recommendation, referred to as the telecommunications Markup Language (tML) framework document, establishes the tML.

tML is used in the context of the telecommunications Operations, Administration, Maintenance and Provisioning (OAM&P) domain[1]. tML is an application of the World Wide Web Consortium (W3C) eXtensible Markup Language (XML).

tML is to be used as a message format between telecommunications OAM&P entities for functions such as ordering, billing, maintenance, and provisioning.

This Recommendation contains guidelines for developing tML message structures for use at the TMN x Reference Point (see ITU-T Recs M.3010 [1] and M.3013 [2]).

This version of the tML Framework focuses on the TMN X interface. Subsequent versions may extend the scope to include other TMN interfaces.

The scope of this version of the Recommendation (Figure 1) includes rules, objectives, and guidelines for:

– Specifying business document structure (i.e. tML Schemas) for X interface applications of tML;

– Use of common vocabulary structure;

– Use of namespaces;

– Mapping from existing standards to tML;

– Specification of metadata used.

The scope of this Recommendation does not specify the following items because trading partners specify these items through negotiation:

– Business Process Scenario;

– Implementation Infrastructure Profile-Specification of any particular communications protocol profile (including provisions for reliability, availability and survivability, or RAS), and provisions for security, privacy, and non-repudiation;

– Data and Vocabulary content.

[1] Other uses of the term t-m-l exist today, and shouldn't be confused with telecommunications Markup Language (tML). tML distinguishes itself from other uses of the term with a lower case "t". Two examples of other TMLs are:

1) Tutorial Markup Language (TML), an interchange format designed to separate the semantic content of a question from its screen layout or formatting (TML has been specified using SGML); and

2) Telephony Markup Language (TML), a proprietary framework for applying the Web for distributed Computer Telephony and messaging applications.

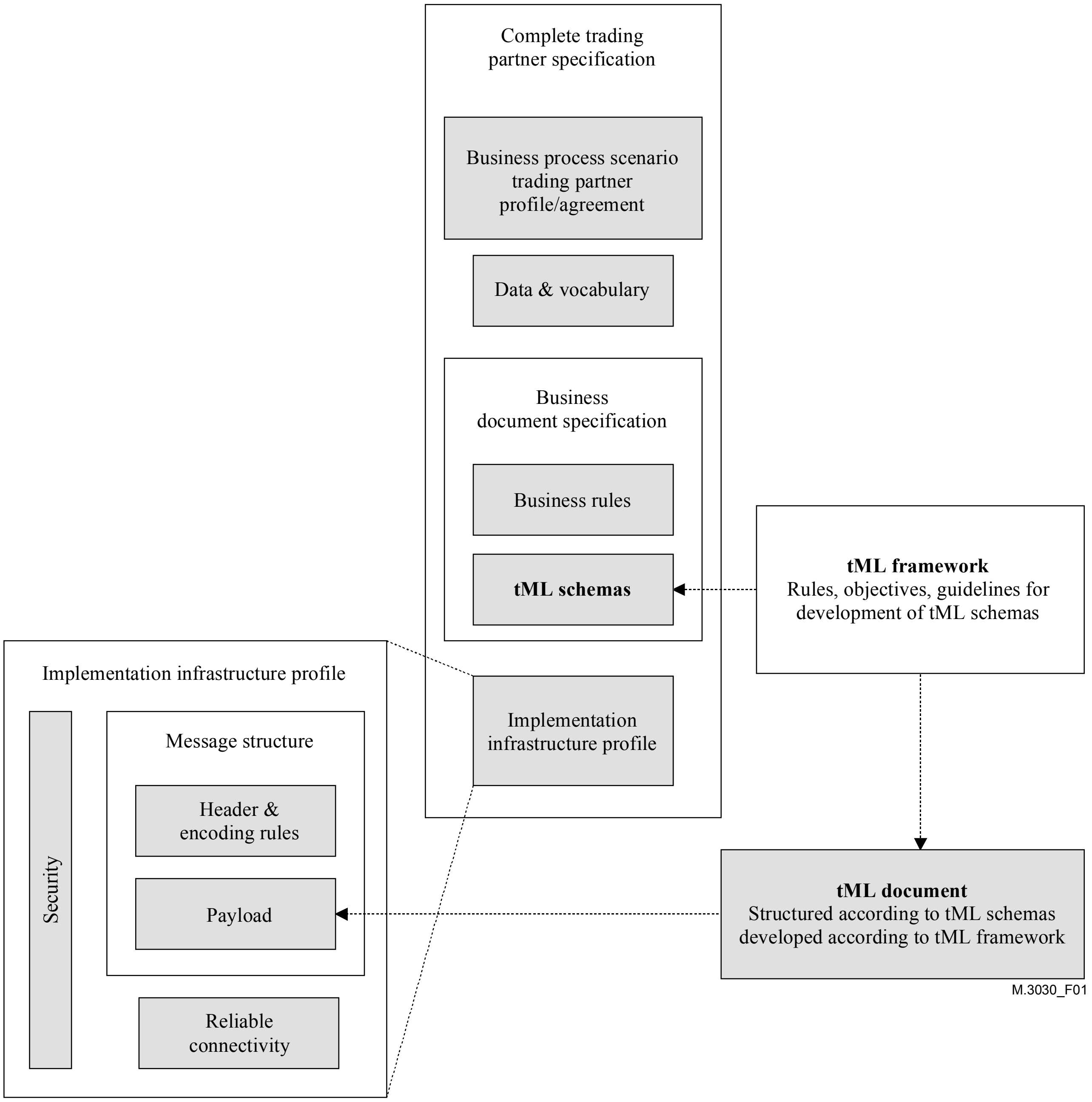
Complete trading partner specification
Business process scenario trading partner profile/agreement
Data & vocabulary
Business document specification
Business rules
tML schemas
tML framework
Rules, objectives, guidelines for development of tML schemas
Implementation infrastructure profile
Implementation infrastructure profile
Security
Message structure
Header & encoding rules
Payload
Reliable connectivity
tML document
Structured according to tML schemas developed according to tML framework
M.3030_F01

Figure 1/M.3030 – tML Framework Scope

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

2.1 ITU-T Recommendations

[1] ITU-T Recommendation M.3010 (2000), *Principles for a telecommunications management network.*

[2] ITU-T Recommendation M.3013 (2000), *Considerations for a telecommunications management network.*

2.2 ISO Standards

[3] ISO/IEC 10646-1:2000, *Information technology – Universal Multiple-Octet Coded Character Set (UCS) – Part 1: Architecture and Basic Multilingual Plane.*

[4a] ISO/IEC 11179-3:1994, *Information technology – Specification and standardization of data elements – Part 3: Basic attributes of data elements.*

[4b] ISO/IEC 11179-5:1995, *Information technology – Specification and standardization of data elements – Part 5: Naming and identification principles for data elements.*

2.3 World Wide Web Consortium (W3C) Recommendations

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

[6] XML Schema Part 1: Structures, 2 May 2001.

[7] XML Schema Part 2: Datatypes, 2 May 2001.

[8] Namespaces in XML, 14 January 1999.

2.4 IETF Specifications

[9] IETF RFC 2141 (1997), *URN Syntax.*

[10] IETF RFC 2396 (1998), *Uniform Resource Identifiers (URI): Generic Syntax.*

2.5 Object Management Group Specifications

[11] Unified Modeling Language Specification, Version 1.4, September 2001.

2.6 Committee T1 Standards (Sponsored by ATIS, Alliance for Telecommunications Industry Solutions)

[12] T1.227-2000 – *OAM&P – Extension to Generic Network Information Model for Interface Between Operations Systems Across jurisdictional Boundaries to Support Fault Management.*

3 Definitions

This Recommendation defines the following terms:

3.1 XML-based definitions

3.1.1 attribute: A characteristic of an object or entity (ISO/IEC 11179-3); a property of an element; attributes can be viewed as metadata for the element because they can pass information about an element. Composed structurally as *name=value* pair, where *name* is the name of the attribute and *value* is the value of the attribute.

3.1.2 character: An atomic unit of text as specified by ISO/IEC 10646; a single alpha, numeric, or punctuation mark, as defined by ISO/IEC 10646.

3.1.3 character data: All text characters that are not markup characters.

3.1.4 content model: All data between the start tag and end tag of an element that affects the structure of the element in the instance document. This could be attributes or other elements within an element. (This term is not used).

3.1.5 context: A designation or description of the application environment or discipline in which a name is applied or from which it originates (ISO/IEC 11179-3).

3.1.6 data dictionary: A collection of metadata about the data objects or items in a data model including such characteristics as name, semantic meaning, relationships, and type of data.

3.1.7 data element: A unit of data for which the definition, identification, representation, and permissible values are specified by means of a set of attributes (ISO/IEC 11179-3).

3.1.8 default namespace: A namespace with an empty prefix. A default namespace is considered to apply to the element where it is declared (if that element has no namespace prefix), and to all elements with no prefix within the content of that element. If the URI reference in a default namespace declaration is empty, then unprefixed elements in the scope of the declaration are not considered to be in any namespace.

3.1.9 delimiter: A special character that marks the beginning and end of a string or text field.

3.1.10 document: A class of data object; may be the text of a printed document, a set of database records.

3.1.11 document element: The root element of an instance document. There is exactly one element, called the root, or document element, no part of which appears in the content of any other element.

3.1.12 element: A logical data structure within an XML document; start tags and end tags define the beginning and end of an element. Each XML document contains one or more elements, the boundaries of which are either delimited by start-tags and end-tags, or, for empty elements, by an empty-element tag. Each element has a type, identified by a case-sensitive name, sometimes called its "generic identifier" (GI), and may have a set of attribute specifications. Each attribute specification has a name and a value.

3.1.13 element declaration: Associating a name with a type. (This term is only within the definition of local element declaration, which is not used).

3.1.14 element type: The name that appears in a start-, end- or empty-tag. In the following example there are three elements but only two element types:

```
<enduser>
     <firstname>Jo Anne</firstname>
     <firstname>Stephen</firstname>
</enduser>
```

An element type has element content when elements of that type must contain only child elements, optionally separated by white space.

3.1.15 empty element: Elements that do not have content. An empty-tag follows the syntax: <name></name> or <name/>.

3.1.16 end tag: The end of every element that begins with a start-tag must be marked by an end-tag containing a name that echoes the element's type as given in the start-tag. It is assumed that an XML processor is doing its work on behalf of another module, called the application.

3.1.17 entity: A virtual storage unit of no fixed value, identified by a name; often a separate file, but may be a string or even a database record.

3.1.18 eXtensible Markup Language (XML): By construction, XML documents are conforming SGML documents; a W3C Recommendation; an application profile or restricted form of the Standard Generalized Markup Language (SGML). By construction, XML documents are conforming SGML documents. The XML subset of SGML has been specifically designed to function on the Web. While HTML's tags are predefined, XML allows tags to be defined by the developer of the page. Thus, XML-defined Web pages can function like database records.

3.1.19 generic identifier (GI): The name of an element "type". Sometimes the term "tag name" is used to refer to a GI.

3.1.20 global element (and attribute) declarations: Global elements, and global attributes, are created by declarations that appear as the children of the schema element. Once declared, a global element or a global attribute can be referenced in one or more other declarations.

3.1.21 HyperText Markup Language (HTML): A system of coding information from a wide range of domains (e.g. text, graphics, database query results) for display by World Wide Web browsers. Certain special codes, called tags, are embedded in the document so that the browser can be told how to render the information.

3.1.22 import: Defines incorporation of type declarations from another schema. The import mechanism is used in an XML Schema to allow definitions and declarations contained in other schemas under different namespaces to be referenced in the document. If a developer references schema fragments or modules from another namespace, then the import mechanism must be used. If the developer references fragments or modules from the same namespace, then the include mechanism must be used.

3.1.23 include: Defines incorporation of another schema into the existing namespace. An entity is included when its replacement text is retrieved and processed, in place of the reference itself, as though it were part of the document at the location the reference was recognized. XML Schemas can be included in other XML Schemas. Such included schema documents must either

a) have the same target namespace as the <include>ing schema document; or

b) no target namespace at all, in which case the <include>d schema document is converted to the <include>ing schema document's target namespace.

3.1.24 local element declarations: Local element declarations are nested further inside a schema structure and are not direct children of the root schema element (This term is not used).

3.1.25 local name: A local name is the local part of a qualified name. This is called the local part in namespaces in XML.

3.1.26 lower camel case: Lower Camel Case is a capitalization pattern where the first letter is lowercase, but after that all words are capitalized and no separators are used. thisIsAnExample.

3.1.27 markup: Markup takes the form of start-tags, end-tags, empty-element tags, entity references, character references, comments, CDATA section delimiters, document type declarations, processing instructions, XML declarations, text declarations, and any white space that is at the top level of the document entity.

3.1.28 metadata: Data that describes other data.

3.1.29 meta-language: A language that describes other languages. SGML and XML are considered meta-languages because they define markup languages such as tML. (This term is not used).

3.1.30 name: A token beginning with a letter or one of a few punctuation characters, and continuing with letters, digits, hyphens, underscores, colons, or full stops, together known as name characters.

3.1.31 namespace: A conceptual collection of unique names identified by a URI or a URN reference [IETF RFC 2396]; used in XML documents as element types and attribute names. Namespaces are used in XML to qualify names in order to separate them from other names.

NOTE – A URI has a similar format to a URL. It is not always possible to resolve a URI to find an instance of an XML schema.

3.1.32 namespace prefix: A string that associates an element or attribute name with a namespace URI in XML.

3.1.33 namespace root: The standard part of any tML Namespace: e.g. `urn:int.itu/tML`.

3.1.34 namespace URI: A URI that identifies an XML namespace. Strictly speaking, this actually is a namespace URI reference. This is called the namespace name in W3C document Namespaces in XML.

3.1.35 object class term: A component of the name of a data element that represents the logical data grouping (in a logical data model) to which it belongs; e.g. "employee." (This term is not used).

3.1.36 parent element: An element containing other elements; the elements contained within the parent element are known as child elements. (This term is not used).

3.1.37 qualified name: The name of an element or attribute defined as the concatenation of a local name (as defined in this specification), optionally preceded by a namespace prefix and colon character.

3.1.38 registry: In the context of this Recommendation, the location of metadata about a repository. A registry is provided for users who want to locate schemas, owners of schemas, and other information stored in a repository.

3.1.39 repository: One or more globally distributed locations used to store schemas, names and locations of schema owners, UML models, and other data and constructs needed to facilitate interoperable exchanges of information between entities.

3.1.40 root element: One element that contains all other elements of the document; the root element may be the document element.

3.1.41 schema: A set of schema components. A collection (vocabulary) of type definitions and element declarations whose names belong to a particular namespace called a target namespace. A schema defines the allowable contents of a class of XML documents. The purpose of a schema is to define and describe a class of XML instance documents by using these constructs to constrain and document the meaning, usage and relationships of their constituent parts: datatypes, elements and their content, attributes and their values, entities and their contents, and notations. A class of documents refers to all possible permutations of structure in instance documents that will still conform to the rules of the schema.

3.1.42 schema components: The generic term for the building blocks that comprise the abstract data model of the schema. There are 13 kinds of schema components defined: named components (Simple type definitions, Complex type definitions, Attribute declarations, Element declarations, attribute group definitions, identity-constraint definitions, model group definitions, notation declarations, annotations), and un-named components (model groups, particles, wildcards, and attribute uses).

3.1.43 schema element: The root element of a schema document, i.e. `<schema>`.

3.1.44 semantics: The branch of linguistic science that deals with the meanings of words (Webster). Semantics concerns the meanings of name parts and possibly separators that delimit them (ISO 11179).

3.1.45 Standard Generalized Markup Language (ISO 8879): a meta-language used to construct other markup languages. XML is a simple dialect of SGML, sometimes referred to as SGML-lite.

3.1.46 start tag: The beginning of every non-empty XML element is marked by a start-tag.

3.1.47 syntax: The structure of expressions in a language, and the rules governing the structure of a language. The relationships among characters or groups of characters, independent of their meanings or the manner of their interpretation and use. (ISO/IEC 11179-1).

3.1.48 tags: Text structures (markup characters) that mark the beginning and end of elements within an XML-based document. Two types of tags exist, a start-tag and an end-tag. The name in the start- and end-tags gives the element's type. This is the element-type name. The element's type given in the start-tag must be matched in the end-tag. The text between the start-tag and end-tag is called the element's content.

3.1.49 tag name: Not really a "tag name" but properly referred to as a generic identifier. For example, in `<automobile>`, "automobile" is a generic identifier. A generic identifier is unique in its namespace.

3.1.50 target namespace: Target namespaces enable us to distinguish between definitions and declarations from different vocabularies. For example, target namespaces would enable us to distinguish between the declaration for element in the XML Schema language vocabulary, and a declaration for element in a hypothetical chemistry language vocabulary.

3.1.51 universal character set (UCS): A character set defined jointly by Unicode and ISO/IEC 10646; encodes most of the world's writing systems in a single character set, allowing users to mix languages and scripts within a document without needing any tricks for switching character sets.

3.1.52 unicode: A character coding system designed to support the interchange, processing, and display of the principal written languages of the Americas, Europe, the Middle East, Africa, India, Asia, and the Pacific Basin. Unicode provides a unique number for every character, regardless of the platform, the program, or the language. The most current version of the Unicode standard contains 49194 distinct coded characters. Unicode is a 16-bit superset of the ASCII character set. Unicode standards are synchronized with UCS-2 subset of ISO 10646. Although the character codes are synchronized between Unicode and ISO/IEC 10646, the Unicode Standard imposes additional constraints on implementations to ensure that they treat characters uniformly across platforms and applications. To this end, it supplies an extensive set of functional character specifications, character data, algorithms and substantial background material that is not in ISO/IEC 10646.

3.1.53 upper camel case: Upper Camel Case is a capitalization pattern where the initial letters of all words are capitalized and no separators are used. ThisIsAnExample.

3.1.54 UCS Transformation Format (UTF): A Unicode Standard. ISO/IEC 10646-1 defines a multi-octet character set called the Universal Character Set (UCS) which encompasses most of the world's writing systems. Multi-octet characters, however, are not compatible with many current applications and protocols, and this has led to the development of a few so-called UCS transformation formats (UTF), each with different characteristics: UTF-8, UTF-16, and UTF-32 [IETF RFC 2279]. Text data is said to be in a Unicode encoding form if it is encoded in UTF-8, UTF-16 or UTF-32. XML is entirely defined in terms of Unicode characters and mandates the UTF-8 and UTF-16 encodings while allowing any other encoding for parsed entities. The XML 1.0 specification requires all conforming XML processors to accept both UTF-16 and UTF-8.

3.1.55 UCS Transformation Format 8 (UTF-8): UTF-8 encoding is of variable-length, and characters are encoded with one, two, three, or four bytes. It is the only standard that currently defines the Euro and that can display all characters in all locales. UTF-8 uses all bits of an octet, but has the quality of preserving the full US-ASCII range: US-ASCII characters are encoded in one octet having the normal US-ASCII value and any octet with such a value can only stand for an US-ASCII character, and nothing else. [IETF RFC 2279].

3.1.56 UCS Transformation Format 16 (UTF-16): UTF-16 encoding is a variable-length 16-bit representation. Each character is made up of one or two 16-bit units. In terms of bytes, each character is made up of two or four bytes.

3.1.57 UTF-32: UTF-32 encoding is a fixed 32-bit (four-byte) representation.

3.1.58 valid: An XML document with a declared schema that follows all the rules of that definition.

3.1.59 version identifier: The identification of an issue of a schema specification in a series of evolving schema specifications within a registration authority.

3.1.60 vocabulary (telecommunications): A collection of words (nouns, verbs) and meanings within the construct of a given telecommunications context.

NOTE – Element type names are representations of (derived from) specific vocabulary words.

3.1.61 well-formed document: An XML document, or textual object, that follows all of the rules in the W3C XML Recommendation. A textual object is a well-formed XML document if:

1) Taken as a whole, it matches the production labeled document;
2) It meets all the well-formedness constraints given in the XML Recommendation;
3) Each of the parsed entities that is referenced directly or indirectly within the document is well-formed.

3.1.62 World Wide Web Consortium (W3C) Recommendation: Reflects consensus within W3C that the ideas or technology specified by a Recommendation are appropriate for widespread deployment and promote W3C's mission.

3.1.63 XML aware: Any software application that recognizes an XML-based data format and understands XML concepts. (This term is not used)

3.1.64 XML declaration: An optional declaration at the top of an XML-based document that specifies the version of XML, e.g. the following is a complete XML document, well-formed but not valid: `<?xml version="1.0"?> <greeting>Hello, world!</greeting>.`

3.1.65 XML derivative: Any application of the XML Recommendation that is designed to support a specialized purpose, or vertical market: examples are CML (Chemical Markup Language), MathML, FPML (financial products markup language), WML (wireless markup language), and telecommunications Markup Language (tML).

3.1.66 XML document: A data object is an XML document if it is well-formed, as defined in the XML Recommendation.

3.1.67 XML vocabulary: An XML-based tag set with a specific functionality. tML, WML, and MathML are examples of XML vocabularies.

3.2 Internet Engineering Task Force (IETF)-based definitions

3.2.1 IETF: A large open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet. It is open to any interested individual. The IETF publishes Requests for Comments and specifications.

3.2.2 Uniform Resource Identifier (URI): An IETF standard. The basic form of address on the Web. This includes URL (Uniform Resource Locators) and all future resource categories. URLs are created using URI schemes, e.g. http:// and ftp:// are specific subsets of a URI.

3.2.3 Uniform Resource Locator (URL): An IETF standard. Describes the location of Web resources; a hierarchical scheme consisting of a protocol (e.g. http), followed by a hostname (e.g. www), and then a datapath.

3.2.4 Uniform Resource Name (URN): A name that identifies a resource or unit of information independent of its location. A URL identifies the location of a container for an instance of a resource identified by a URN.

3.3 Object Management Group (OMG) definitions

3.3.1 Unified Modelling Language (UML): An OMG specification to provide system architects working on object analysis and design with one consistent language for specifying, visualizing, constructing, and documenting the artifacts of software systems and for modeling of business processes.

3.4 This Recommendation defines the following additional terms:

3.4.1 Global Telecommunications Data Dictionary (GTDD): A public database containing (at a minimum) the name, source, version, description, semantics, value types, and namespaces of terms used for telecommunications Operations, Administration, Maintenance and Provisioning (OAM&P) applications. tML is one of many client applications of the GTDD.

3.4.2 implementation framework: A set of specifications or architectural components applied to transport tML instance documents between entities.

3.4.3 telecommunications Markup Language (tML): A vocabulary derived from application of the W3C XML Recommendation; tML is used as a message format in telecommunications OAM&P interface applications.

3.4.4 tML base library: A grouping of reusable common data components. Within a region there may be any number of these libraries. They may not have a namespace of their own and are `included` into tML Schemas where appropriate. Each data component within a set of Regional Base Libraries shall be uniquely named.

3.4.5 tML framework: The ITU-T Recommendation that guides a set of rules, recommendations and guidelines that supports development and implementation of tML Schemas in the TMN context. This Recommendation is known as the "tML framework document".

3.4.6 tML namespace: A collection of names, identified by a URI reference [RFC 2396], that is used in tML documents as element types and attributes.

3.4.7 tML region: A participating Standards Development Organization that uses the tML Framework, e.g. ITU, ANSI T1, ETSI, and JSI.

3.4.8 tML schema: Constructs to constrain and document the meaning, usage and relationships of the constituent parts of tML instance documents. tML Schemas are based on the W3C XML Schema Recommendation.

3.4.9 tML tag (domain specific): Name for a domain specification element, attribute, or entity that is unambiguous within a specific tML domain.

3.4.10 vocabulary: In the context of this Recommendation, a particular set of terms used to convey information between trading partners. The particular set of terms used to form the vocabulary is dependent on the application. The terms used to form a vocabulary are found in one or more places, including the Global Telecommunications Data Dictionary (GTDD). tML vocabularies represent telecommunications specific terms, and these terms are located in the GTDD.

4 Abbreviations

This Recommendation uses the following abbreviations:

ANSI	American National Standards Institute
B2B	Business-to-Business
ETSI	European Telecommunications Standards Institute
GTDD	Global Telecommunications Data Dictionary
IETF	Internet Engineering Task Force
ISO	International Organization for Standardization

ITU-T	International Telecommunication Union – Telecommunication Standardization Sector
OAM&P	Operations, Administration, Maintenance and Provisioning
OO	Object-Oriented
OS	Operations System
OSS	Operational Support System
SDO	Standards Development Organization
TA	Trouble Administration
tML	telecommunications Markup Language
TMN	Telecommunication Management Network
UCS	Universal Multiple-Octet Coded Character Set
UID	Unique Identifier
URL	Uniform Resource Locator
URN	Uniform Resource Name
UTF	UCS Transformation Format
W3C	World Wide Web Consortium
XML	eXtensible Markup Language

5 Conventions

Conventions are followed in this Recommendation to make the reader aware of the categories of urgency in the text.

5.1 Requirements

Certain paragraphs contain mandatory Requirements that "must" or "shall" be met by a conforming entity. Requirements are preceded by boldface "**Rule**" enclosed in parentheses, and are followed by a short statement indicating the requirement.

5.2 Objectives

Certain paragraphs contain Objectives. These paragraphs contain a recommended course of action or application of best practice that "should" be implemented, if possible, at the discretion of the entity.

5.3 Guidelines

Certain paragraphs contain Guidelines. These paragraphs contain preferences that "may" be followed by an entity, at their choosing.

5.4 Font

Elements and attributes are shown in `Courier New` font.

5.5 The term "shall"

In order to indicate compliance with this Recommendation, the term "shall" is indicated, should one elect to use this Recommendation.

6 General

6.1 Goals for this Recommendation

6.1.1 Key goals for this Recommendation

a) To provide uniformity in the development of tML Schemas;

b) to facilitate interoperability between trading partners on the TMN X interface using XML technology.

6.1.2 Secondary goals

a) To enable the telecommunications industry to utilize emerging platform independent Business-to-Business (B2B) frameworks. This will be achieved in this Recommendation by encouraging contributors to place their work in a structure of tML Schemas that will enable reuse across domains. It will also show that creating tML Schemas is only one part of the process of enabling interchange of data between trading partners;

b) to lay the foundations for the Global Telecommunications Data Dictionary (GTDD). This will be achieved by encouraging the tML Schemas to contain as full a set of semantic information as possible and thereby allowing the GTDD to be populated from the tML Schemas. It should be noted that tML Schemas will not be the only source of information for the GTDD and that, in the future, tML Schemas may be generated from the GTDD.

6.1.3 Future additions to ITU-T Rec. M.3030 to be accomplished

tML use of the GTDD and registry/repository.

6.2 Derivation of term 'tML'

The term 'telecommunications Markup Language' (tML), in the context of this Recommendation, refers to a mark-up language derived from applying the W3C XML Recommendation.

The vocabulary (see clause 3) and semantics of the tML language is to be defined in the Global Telecommunications Data Dictionary (GTDD). A unique identifier (UID) will be assigned in the tML Schema to each element name. The UID will map the element name to its metadata, including its definition (semantics), in the GTDD.

The definition of each element and associated metadata shall be included with each element in a tML Schema submitted to the ITU-T. Refer to Annex A for more information about the metadata to be specified in a tML Schema. The definitions and metadata contained in tML schemas can be a source for GTDD entries subsequently.

tML is a hierarchical structure (Figure 2) of W3C XML-based schemas that build upon each other with increasing specialization in the following sequence:

1) a base (core) vocabulary consisting of common data elements for use in all domains;

2) a set of operation-specific application domain vocabularies, e.g. provisioning, trouble administration, and billing;

3) a set of service/technology-specific domain vocabularies, e.g. DSL, SDH, and ISDN.

Overlaying these domains are international and regional-specific variations.

6.3 tML Schema Structure

The tML Schema structure show in Figure 2 illustrates the method by which schemas will be included in tML. Namespaces will be assigned to these schemas according to this structure, i.e. regional/application/technology. Figure 2 shows the relationship between the Regional Base Library of common data elements and the application and technology domains. Regions may consist of ITU, ANSI T1, ETSI, or JSI SDOs or other domains of common interest. For each Region, (for which international is just a special case of an SDO Region) the complete structure enclosed by the 'Regional' box will exist for each tML Region. This will enable each Region to develop standard schemas without reference to other geographies, while not preventing adoption of those schemas by another Region. This will enable contributions that meet the local needs to progress without being bound by 'international' requirements, while at the same time allowing these regional contributions to be used more widely.

In Figure 2, the External Resource Libraries represent potential source material to be introduced into tML Schemas. tML Schemas shall not **`import`** from any External Resource Library or schema, but shall adopt the useful terms from them so that new tML data components are created. i.e. a tML Schema shall not refer to any schemas that are not tML compliant.

For the purposes of this Recommendation, 'International' is regarded as a peer to other regions. This means that the set of International tML Schemas has no precedence over a Regional tML Schema. However, tML Schemas that are common to more than one region may become an International tML Schema. For example, the North American ANSI Trouble Administration tML Schema (ANSI/TA) would not inherit properties from the International Trouble Administration tML Schema (ITU/TA). This does not precluded the ANSI standard from adopting features of the International Standard, where their use is appropriate. One major reason for this is to keep the tML Namespace specification as manageable as possible. To this end, the Regional Base Library should not have an independent namespace but should be **`included`** where appropriate. Data Components in the Regional Base Libraries within any region shall be uniquely named. This naming requirement shall be managed by the relevant SDO within the region.

It is intended that the application domains will take precedence over the technology domains to maximize reusability of tML Schema elements. The tML Framework does not preclude the use of libraries of common data elements from a technology domain. It does however restrict these to being **`included`** into the appropriate application domain namespace. Such libraries would not have a namespace of their own, but would become part of the application domain namespace into which they are **`included`**.

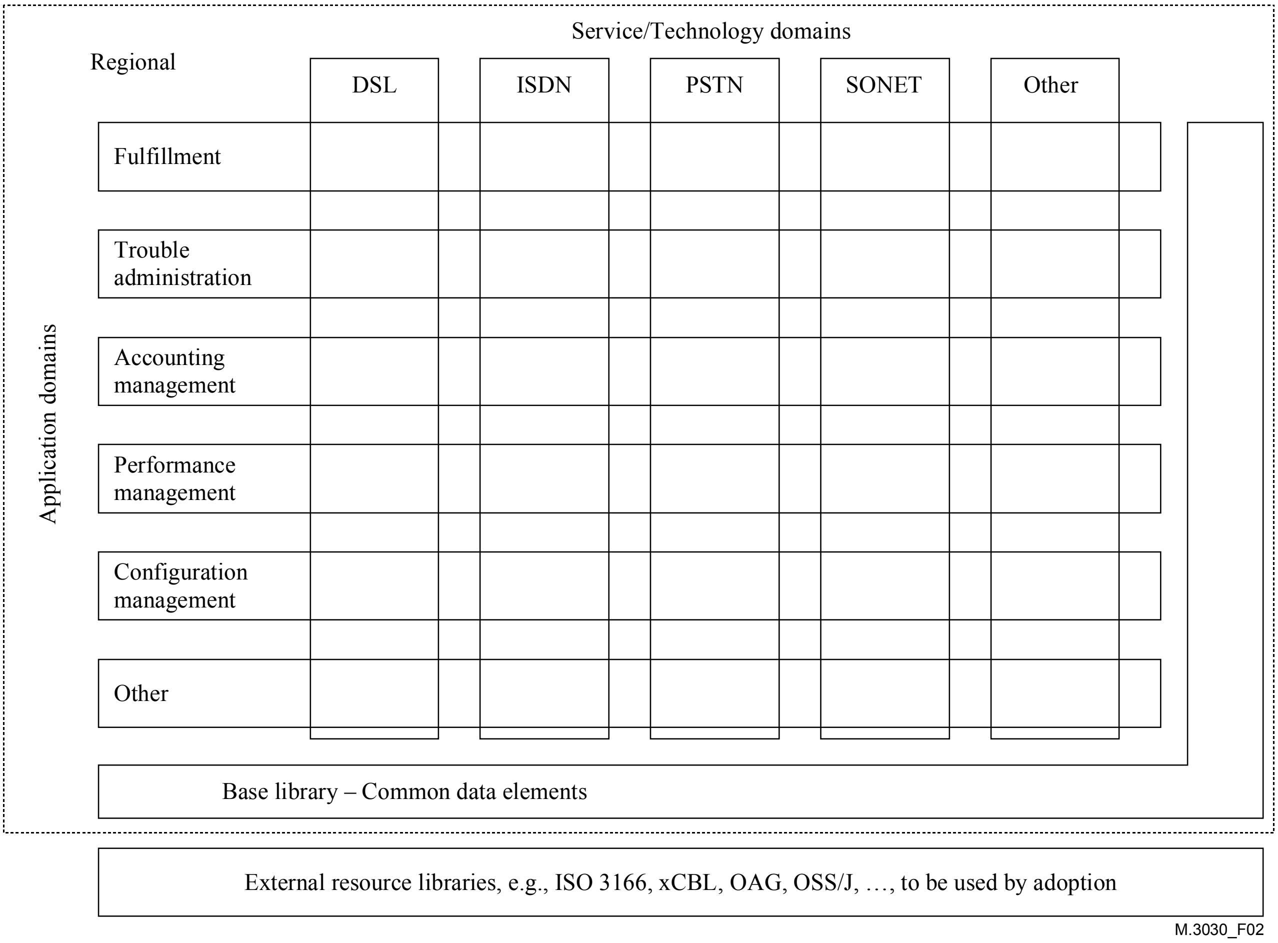

Figure 2/M.3030 – tML Schema Structure

An example of this (Figure 3) is a contribution based on the North American T1.227 standard for Trouble Administration[2]. This is in the main 'generic' standard for Trouble Administration without reference to any particular technology. Therefore, the general parts would be in the ANSI/TroubleAdministration domain. However, certain of the details within T1.227, that refer to specific types of fault only applicable to particular technologies (such as Special Services Circuits), would be in the ANSI/TroubleAdministration/SpecialServicesCircuits domain. Figure 3 shows how these would be related.

[2] T1.227-2000-OAM&P – Extension to Generic Network Information Model for Interface Between Operations Systems (OSs) Across Jurisdictional Boundaries to Support Fault Management. The scope of this standard is limited to Operations System to Operations System interfaces for OSs used for network management and located in different jurisdictions.

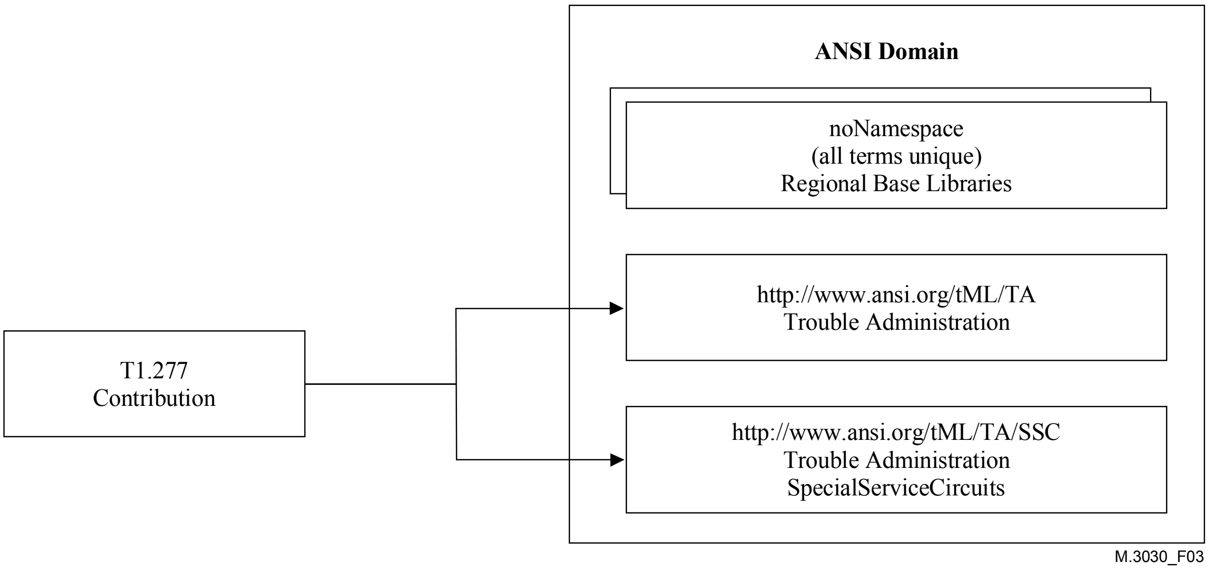

Figure 3/M.3030 – An example mapping of contribution-to-tML schema structure

6.4 tML use at TMN X Reference Points

tML may be applied initially at the TMN X Reference Point

6.4.1 tML at TMN X Reference Point

tML can be used on the X interface (at the TMN X reference point) as the universal language for describing data formats regardless of the hardware platform, software operating system, or programming language used by two or more Administrations exchanging information. Use of tML is important for Administrations that wish to integrate applications with their Web-based systems. tML documents exchanged over the TMN X interface can be made part of any strategic plans (that are based on the XML message format) for integrating Operational Support Systems (OSS) and enterprise information technology (IT) systems.

Example Application Domains of tML at the X interface are (see Figure 2):

– *Fulfillment*
  - Pre-Order;
  - Order;
  - Provisioning;
  - Etc.
– *Accounting Management*
  - Billing;
  - Account Information Maintenance;
  - Etc.
– *Performance Management*
– *Trouble Administration*
  - Testing;
  - Maintenance Management;
  - Etc.
– *Configuration Management*
– *Other*

A future source of application domain content for the X interface is ITU-T Rec. M.1400.

7 tML Technical Specifications

7.1 Conformance with the Recommendation

The following clauses provide Rules, Objectives, and Guidelines pertaining, directly or indirectly, to developing tML Schema standards.

Schemas submitted for standardization shall be deemed conformant with this Recommendation provided that they comply with all requirements (R items) in this Recommendation.

A **Rule** is a prescribed practice to be followed. To be conformant with this Recommendation, the indicated Rule shall be followed in practice. An implementation shall be strictly conformant to this Recommendation if the implementation always behaves as specified by the Rules in this Recommendation whenever it exchanges messages (either as initiator or responder) with another implementation.

An **Objective** is a voluntary practice. To be conformant with this Recommendation, it is not necessary to follow an Objective.

A **Guideline** is a choice of style when determining a course of action. To be conformant with this Recommendation, it is not necessary to follow a Guideline.

7.2 tML Schemas – General

The W3C Schema language is a W3C Recommendation that is used as a basis to develop tML Schema documents. A tML Schema is made available to a processor of a tML instance document so that it can validate the syntax of the instance document. tML Schemas are developed as industry standards. In the most basic application a particular tML Schema is applied to a class of tML documents as a validation mechanism for that class of documents. An instance of a class of tML documents is compared to the tML Schema to determine if the instance document conforms to the specific set of rules defined in the tML Schema related to instance document structure and syntax.

7.3 Validation of tML instance documents

When entities exchange tML instance documents they may not require that the documents be validated against a tML Schema. The only rule for tML instance document exchange is that tML documents are well-formed, meaning that they are constructed in accord with the W3C XML Recommendation. They do not need to be validated. However, in most cases, entities will require validation of the tML instance document against a tML Schema in order that the application that processes the instance document has a standard measure of structural quality. This measure enhances the prospect of interoperability between entities. Furthermore, the use of a tML Schema improves productivity because the Schema will have been pre-approved by a standards body and thus there will be no need for each entity to develop its own mechanism to ascertain the quality of the data structure in an instance document. To make schema validation easier to accomplish at little or no cost, there are XML Schema validation tools available on the public Internet.

7.4 Other Goals

tML Schema developers should strive for the following goals:

– Interoperability;

– Reuse;

– Extensibility;

– Reduced complexity;

– Enhanced readability by humans.

7.5 Rules, objectives and guidelines

7.5.1 Schema validation

Rule 1: tML Schemas shall be valid according to W3C Schema Recommendation May 2001, Parts 1 and 2.

Rule 2: tML Schemas shall conform to all the **Rules** in this Recommendation.

Guideline 1: tML Schemas should be encoded in UTF-16 character set. Because UTF-8 is a subset of UTF-16 it may be used optionally by pair-wise agreement.

Guideline 2: tML Schemas should be subjected to a rigorous review process to help prove the conformance to this Recommendation.

7.5.1.1 Local and global declarations

Rule 3: tML Schemas shall set elementFormDefault to qualified.

Rule 4: tML Schemas shall set attributeFormDefault to unqualified.

7.5.1.2 XML schema namespace declaration

Rule 5: The XML Schema namespace must be declared as http://www.w3.org/2001/XMLSchema.

Guideline 3: The prefix **xsd**: Should be used to indicate this namespace but in practice any prefix can be used.

Guideline 4: Every Domain Specific tML Schema should have at least two namespaces: the targetNamespace and the XMLSchema namespace (http://www.w3.org/2001/XMLSchema).

7.5.2 Use of 'External' and 'Base' libraries

Guideline 5: tML Schemas shall not `import` from any External Resource Library or Schema but shall adopt the useful terms from them so that new tML data components are created.

Guideline 6: a tML Schema shall not refer to any namespaces outside of the tML Framework.

Guideline 7: To this end, the Regional Base Libraries should not have an independent namespace but should be `included` where appropriate. This does not preclude regional SDO from using namespaces to guarantee naming uniqueness, but it is discouraged as it could lead to tML Documents referring to large numbers of namespaces.

Rule 6: Data Components in the Regional Base Libraries within any region shall be uniquely named.

Rule 7: Base library naming requirements shall be managed by the relevant SDO within the region.

7.5.3 Namespace naming

tML does not define a complete namespace but only one section that is designed to indicate that a schema is conformant to this Recommendation.

Rule 8: The namespace root shall be the responsibility of the regional SDO: example:

```
http://www.ansi.org/
urn:int.itu/
http://etsi.org/
```

Guideline 8: tML Schemas that are not intended to be accessible by using the Namespace URI as a URL should use a URN, e.g.:

`http://www.ansi.org/tML/` – could be thought to be a web location (URL), but is a name

`urn:int.itu/tML/` – is a unique name and will not be thought to be a Web address.

Rule 9: After the namespace a conformant tML Schema shall include tML/, therefore the namespace roots above would become:

```
http://www.ansi.org/tML/
urn:int.itu/tML/
http://etsi.org/tML/
```

Guideline 9: tML Schemas may additionally be named in the following manner:

```
ApplicationDomain/TechnologyDomain
```

Where;

`ApplicationDomain` is the name of the application domain in which the schema belongs such as `TroubleAdminisitration`

`TechnologyDomain` is the name the service or technology domain to which the schema belongs. This would not occur in the case of a schema that is generic to a complete Application Domain.

Examples of a complete schema name definition with including a Namespace Root are:

`urn:int.itu/tML/TroubleAdministration` – for the generic schema for International Trouble Administration;

`urn:int.itu/tML/TroubleAdministration/DSL` – for the element of International Trouble Administration relevant only to DSL products;

`http://www.ansi.org/tML/TroubleAdministration` – for the generic schema for American Trouble Administration;

`http://www.ansi.org/tML/TroubleAdministration/DSL` – for the element of Trouble Administration relevant only to DSL products;

`http://etsi.org/tML/FulFilment` – for a European Fulfilment schema; and

`http://etsi.org/tML/FulFilment/DSL` – for European DSL specific schema elements.

If there is more than one schema in any of the areas, either for Application or Technology it should be indicated by prefixing the application or technology followed by a 'dot' and then the schema name: e.g.:

```
urn:int.itu/tML/TroubleAdministration.ErrorCodes or
urn:int.itu/tML/TroubleAdministration/DSL.AcceptanceCodes
```

7.5.4 Naming conventions

Naming Rules, Guidelines and Objectives in this Recommendation are specified as a basis for naming of all named artefacts used in tML Schemas including element and attribute names, type definitions – complex and simple types.

7.5.4.1 Naming considerations

Naming is influenced by a number of factors.

Clarity of the data model is important for developers of applications. It is usually believed that a good data model will outlast the application that uses it.

Legacy data models exist and developers may (and probably will) be required to use names already in existence.

Guideline 10: Names should be descriptive so humans can read them.

Guideline 11: Industry-recognized terms may be desired for a name rather that a newly conceived name. Do not assume that because your part of the industry understands the term that other regions, or new entrants, will understand.

Guideline 12: Expand Abbreviations where any possible doubt as to its meaning might occur. An Abbreviated term may be good for application programs and for display on a screen in the consideration of time and space requirements, respectively.

Guideline 13: There may be a concatenation of multiple words that is desired to make up a name so that the name is descriptive, or because the concatenation represents an object class, or because one wishes to distinguish one name from another that is similar and has different meaning.

7.5.4.2 General conventions

Tag naming Rules, Objectives, and Guidelines in this Recommendation are based on a subset of ISO IEC 11179-5, *Naming and identification principles for data elements*.

Rule 10: tML tag name assignments shall be according to the conventions in this Recommendation unless business requirements mandate other naming conventions.

Objective 1: Names should be intuitive, even if it means that the name is long.

Objective 2: Names should be as short as possible and still be descriptive of the expected content.

Objective 3: Be aware of terms that are already defined; refer to an industry data dictionary before finalizing terminology.

Objective 4: Abbreviations should be avoided.

Objective 5: Names chosen should not be confused with other names in the same domain.

Objective 6: Be consistent in a type of name, using similar names with like items.

Objective 7: Distinguish between type, element and group names by using the suffixes `Type` and `Group`.

Objective 8: Avoid using the same name for an element, a notation, and an entity. This leads to confusion in the data model.

Guideline 14: Simple and Complex type definitions should normally have names.

Rule 11: Terms used in tML Schemas shall be derived from existing terminology agreed upon by industry participants.

Rule 12: Terms used in each domain shall be unique.

Rule 13: Terms shall not be used for ambiguous purposes within the same domain. For example if the tag *address* has been defined as a location identifier it can be used for a customer's address, a delivery address, etc., but not for an "address" that a speaker has given to an audience:

```
<speaker>Abraham Lincoln</speaker>
<address>The White House</address>
<address>Gettysburg Address</address> <!-- not allowed -->
```

Guideline 15: Accurate and complete definitions of tML Schemas and tML tags are critical. The same tML tag should never be used in different ways within a domain.

Rule 14: The same tML term shall never be used for an attribute of an element and a child of the same element.

7.5.4.3 Element and attribute naming

7.5.4.3.1 Element naming

Rule 15: Element names shall not contain the colon character ":".

The reason for this is because such use may cause confusion with a namespace specification. Thus a colon in an element type or attribute name such as `foo:bar` might be interpreted by a processor to mean a separator between an XML namespace prefix `foo` and a local name bar.

Rule 16: Elements names shall only contain Alpha-Numeric characters subject to regional language variations, i.e. Letters and Numbers only; No punctuation.

Guideline 16: Element names should be in Upper Camel Case, e.g. `PurchaseOrder`[3].

7.5.4.3.2 Attribute naming

Rule 17: Attribute names shall not contain the colon character ":".

The reason for this is because such use may cause confusion with a namespace specification. Thus a colon in an element type or attribute name such as `foo:bar` might be interpreted by a processor to mean a separator between an XML namespace prefix `foo` and a local name bar.

Rule 18: Attribute naming should adhere to the following Rules. Attribute names shall only contain Alpha-Numeric characters according to regional language variations, i.e. Letters and Numbers only; No punctuation.

Guideline 17: Attribute names should be in lower Camel Case, i.e. The first word is all in lower case, e.g. partyIdentifier[3].

7.5.5 tML Schema Annotations

It is an objective to maintain as much documentation as is practical within a tML Schema (or instance document) so that one does not have to hunt externally for the needed information. Documentation and comments are especially useful within reusable items such as named complex types, named model groups, and named attribute groups.

Rule 19: The `<documentation>` element shall be used wherever the Schema grammar allows, but preferably inline immediately preceding any element that requires information to clarify the use of the element where the text is inserted.

Rule 20: The `xml:lang` attribute shall be used with any `<documentation>` elements to indicate the language of the information.

While careful choice of tag names lends readability to the tML document, judicious use of comments within a schema definition can greatly extend the understanding of the schema.

Guideline 18: Every tML Schema should contain comments documenting the structure and content of the tML documents described by that schema.

Rule 21: Each term, whether used as a Type, Group, Element or Attribute shall be fully defined with the following information within a `<documentation>` element:

Unique IDentifier (UID);

Short Definition;

[3] There are existing names in common use that prevents this from being a Rule, and there are cases where new element names are to be developed in keeping with past camel case usage rules that may conflict with this Rule.

Full Definition;

Source of the tag including version;

Name;

Basic or Aggregate;

Business Term(s);

Status.

Rule 22: tML Schemas shall have the following metadata:

Full Version History;

List of changes including:

- Dates;
- Person who performed the change;
- Reason for the change;

Original Author;

Date of Creation;

Standards Development Organization responsible for the schema.

7.5.6 Schema Elements vs. Attributes

In most cases, the choice of element vs. attribute is based on the preference of the Schema author. However, the choice is mandatory in other cases.

7.5.6.1 Elements

7.5.6.1.1 When to use Elements

Guideline 19: If you are in doubt about how to use an item, create an element.

Guideline 20: Use an element to represent a piece of information that can be considered an independent object.

Guideline 21: Use an element when the information is related via a parent/child relationship to another piece of information. In this case, the element is also a sub-element of the element to which it is related.

Guideline 22: Use an element when the data changes frequently. XML-aware processors may find it easier to find and modify element data, compared with attributes.

Guideline 23: Use an element if there is a change, further refinement, or revision to the item

7.5.6.1.2 When Elements are mandatory

Rule 23: Elements must be used when there is a link between data items.

Rule 24: Elements must be used to represent reusable structures.

7.5.6.2 Attributes

7.5.6.2.1 Advantages of attributes

- Attributes are useful for qualifying other data items.
- Attributes originally were the only items covered that could be associated with Enumerated Lists of Values. This is not the case with XML Schemas.

7.5.6.2.2 Disadvantages of attributes

- Attributes cannot contain multiple values (child elements can);
- Attributes are not easily expandable (for future changes);
- Attributes cannot describe structures (child elements can).

7.5.6.2.3 When to use Attributes

Rule 25: Attributes must be used for identifiers of elements.

Guideline 24: Use attributes for simple information.

Guideline 25: Keep the number of attributes associated with an element to a minimum. Too many attributes make documents hard to read.

Guideline 26: Use an attribute for information that qualifies elements, such as language.

Guideline 27: Use attributes to control the contexts in which elements are processed.

Guideline 28: Use an attribute to store information that may not be human readable but may be used by computers to process the element properly.

Guideline 29: Use an attribute if the data is a simple string that rarely changes, i.e. fixed values.

7.5.6.2.4 When not to use Attributes

Guideline 30: Do not use an attribute to specify document structure.

Guideline 31: Do not use an attribute if there is any doubt about whether it is appropriate.

Guideline 32: Do not use attributes where future enhancements might be required to further qualify the data to be represented by an attribute.

7.5.7 Versioning

Rule 26: Versioning of tML Schemas shall be done through the use of the `versionID` attribute contained inside the `schema` element. This method of managing version numbers allows backward compatibility of tML Documents from one version to the next. Example:

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" versionID="1.0">
```

Objective 9: Old tML Schema versions should be available for validating documents against those versions, and applications should be able to rely on those old versions existing.

Rule 27: All minor change releases of tML Schemas shall be backward compatible or else shall have versioning capability to satisfy this rule.

7.5.8 Reusable content

Reuse of tML Schema constructs falls into two basic categories:

1) reusing components we have already defined in the target tML Schema document; and
2) global components defined elsewhere. Reuse of internal tML Schema constructs may be accomplished by a number of methods through the use of naming.

7.5.8.1 Named complex types

XML Schema type definitions are independent of instance declarations. Type definitions can be reused in different contexts to describe distinct nodes within the instance document. In this manner, you can build a content model in a consistent, typed manner.

Objective 10: Named Complex Types should be used only if it is important to represent object-oriented (OO) classes in document structures.

New types can be derived from a named complex type by restricting and extending the complex type. This derivation is useful if it is desired to make use of OO concepts in the content model such as inheritance. But the use of named complex types is useful only if the schema developer wishes to represent classes in a document structure and if OO concepts such as inheritance are needed in element content. Unless OO principles are needed in document structures, then named complex types may be difficult to work with, as explained in the clause on named model groups.

Objective 11: A type definition should be suffixed with the keyword "Type" in a tML Schema to improve the readability, e.g. <simpleType name="AlarmType">

Objective 12: Complex types and model groups that are to be named and reused should be stable and not subject to change, due to versioning issues.

7.5.8.2 Named model groups

Model Groups are content models that can be reused in element content models. The model group content model can be used by itself as a content model or it can be reused in another element content model. In certain cases, named model groups may be a preferred method of creating reusable building blocks for constructing schemas, as contrasted with named complex type definitions. Complex types can be extended only at the end of the type. Extending the content model can be done with a named model group by nesting it inside another anonymous complex type or model group.

Objective 13: Named model groups should be considered as the primary method of developing reusable building blocks in a tML Schema.

Objective 14: A group definition should be suffixed with the keyword "Group" in a tML Schema to improve the readability, e.g. <xsd:group name = "NameGroup">.

7.5.8.3 Named attribute groups

Guideline 33: A group of attributes may be reused in multiple elements provided they are given a name.

7.5.8.4 XML Schema constraints

Guideline 34: XML Schemas should not use constraints to enforce business rules within the schema, as the implementation of the constraint enforcement varies across schema tool kits.

7.5.8.5 Enumerations

Objective 15: The use of enumeration type in tML Schemas should be limited to a known and stable set of values. As a general rule of thumb, if a set of valid values changes frequently, they should not be listed in an enumeration.

7.5.8.6 Type vs. Element

Objective 16: The tML Schema developer should create a TYPE instead of ELEMENT when in doubt about whether a particular item will be reusable. An ELEMENT or TYPE can always be derived or extended from a base type.

7.5.9 Interoperability testing

Guideline 35: Any requirement for tML Schema interoperability testing, prior to approving a draft tML Schema, shall be at the discretion of the cognizant work group of the standards body. If a work group in a standards body deems interoperability testing is necessary, participants within the work group shall supply the instance documents.

7.5.10 tML Document-well-formed

Rule 28: A tML Document is inherently an XML-based document, and accordingly, it shall be well-formed according to the W3C XML Recommendation.

Rule 29: A tML Document will conform to a set of definitions defined by tML Schemas.

Annex A

tML Schema Metadata

This schema contains the definitions of the set of Metadata that all tML compliant Schemas must contain. The metadata is divided into certain portions associated with the complete Schema, i.e. this section and that which is associated with a component within a Schema. For the purposes of this Schema the term component refers to any Schema structural element that might benefit from semantic definition. This Schema is itself conformant to the tML Framework (M.3030).

```
<?xml version="1.0" encoding="UTF-16"?>
<xsd:schema targetNamespace="urn:int.itu/tML/tMLSchemaMetadata"
xmlns:tML="urn:int.itu/tML/tMLSchemaMetadata"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
version="1.0">
  <xsd:annotation>
    <xsd:documentation xml:lang="en-GB">
      <tML:SchemaMetadata>
        <tML:OriginalAuthor>Martin Roberts - BT plc</tML:OriginalAuthor>
        <tML:CreationDate>08-03-2002</tML:CreationDate>
        <tML:Description>This schema contains the definitions of the set of
                  Metadata that all tML compliant schemas must contain.
                  The metadata is divided into that which is associated with the
                  complete schema, i.e. this section and that which is associated
                  with a component within a schema.  For the purposes of this schema
                  the term component refers to any schema structural element that
                  might benefit from semantic definition.
                  This schema is itself comformant to the tML Framework
(M.tML)</tML:Description>
        <tML:Source>M.tML</tML:Source>
        <tML:SchemaHistory/>
      </tML:SchemaMetadata>
    </xsd:documentation>
  </xsd:annotation>
  <xsd:element name="SchemaMetadata">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="OriginalAuthor">
          <xsd:annotation>
            <xsd:documentation xml:lang="en-GB">
              <tML:ComponentMetadata>
                <tML:UID>tML000001</tML:UID>
                <tML:Definition>The creator of the original schema</tML:Definition>
                <tML:ComponentType>Basic</tML:ComponentType>
                <tML:Status>Active</tML:Status>
              </tML:ComponentMetadata>
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:element name="CreationDate" type="xsd:date">
          <xsd:annotation>
            <xsd:documentation xml:lang="en-GB">
              <tML:ComponentMetadata>
                <tML:UID>tML000002</tML:UID>
                <tML:Definition>The date the Schema was created</tML:Definition>
                <tML:ComponentType>Basic</tML:ComponentType>
                <tML:Status>Active</tML:Status>
              </tML:ComponentMetadata>
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:element name="Description">
          <xsd:annotation>
            <xsd:documentation xml:lang="en-GB">
              <tML:ComponentMetadata>
                <tML:UID>tML000003</tML:UID>
                <tML:Definition>A textual description of the intended purpose of the
schema</tML:Definition>
                <tML:ComponentType>Basic</tML:ComponentType>
                <tML:Status>Active</tML:Status>
              </tML:ComponentMetadata>
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:element name="Source" minOccurs="0">
          <xsd:annotation>
            <xsd:documentation xml:lang="en-GB">
              <tML:ComponentMetadata>
                <tML:UID>tML000004</tML:UID>
                <tML:Definition>The source of the schema e.g. an existing
standard</tML:Definition>
                <tML:ComponentType>Basic</tML:ComponentType>
                <tML:Status>Active</tML:Status>
              </tML:ComponentMetadata>
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:element name="SchemaHistory" minOccurs="0" maxOccurs="unbounded">
          <xsd:annotation>
            <xsd:documentation xml:lang="en-GB">
              <tML:ComponentMetadata>
                <tML:UID>tML000005</tML:UID>
                <tML:Definition>The full life change history of the
schema</tML:Definition>
                <tML:ComponentType>Aggregate</tML:ComponentType>
                <tML:Status>Active</tML:Status>
              </tML:ComponentMetadata>
            </xsd:documentation>
          </xsd:annotation>
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="VersionNumber">
                <xsd:annotation>
                  <xsd:documentation xml:lang="en-GB">
                    <tML:ComponentMetadata>
                      <tML:UID>tML000006</tML:UID>
                      <tML:Definition>The version of the schema after the
change</tML:Definition>
                      <tML:ComponentType>Basic</tML:ComponentType>
                      <tML:Status>Active</tML:Status>
                    </tML:ComponentMetadata>
                  </xsd:documentation>
                </xsd:annotation>
              </xsd:element>
              <xsd:element name="ChangeDate" type="xsd:date">
                <xsd:annotation>
                  <xsd:documentation xml:lang="en-GB">
                    <tML:ComponentMetadata>
                      <tML:UID>tML000007</tML:UID>
                      <tML:Definition>The date the change was
completed</tML:Definition>
                      <tML:ComponentType>Basic</tML:ComponentType>
                      <tML:Status>Active</tML:Status>
                    </tML:ComponentMetadata>
                  </xsd:documentation>
                </xsd:annotation>
              </xsd:element>
              <xsd:element name="Reason">
                <xsd:annotation>
                  <xsd:documentation xml:lang="en-GB">
                    <tML:ComponentMetadata>
                      <tML:UID>tML000007</tML:UID>
                      <tML:Definition>The reason for the change</tML:Definition>
                      <tML:ComponentType>Basic</tML:ComponentType>
                      <tML:Status>Active</tML:Status>
                    </tML:ComponentMetadata>
                  </xsd:documentation>
                </xsd:annotation>
              </xsd:element>
              <xsd:element name="ChangedBy">
                <xsd:annotation>
                  <xsd:documentation xml:lang="en-GB">
                    <tML:ComponentMetadata>
                      <tML:UID>tML000008</tML:UID>
                      <tML:Definition>The person or organisation who performed the
change</tML:Definition>
                      <tML:ComponentType>Basic</tML:ComponentType>
                      <tML:Status>Active</tML:Status>
                    </tML:ComponentMetadata>
                  </xsd:documentation>
                </xsd:annotation>
              </xsd:element>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:simpleType name="ComponentType">
    <xsd:annotation>
      <xsd:documentation xml:lang="en-GB">
        <tML:ComponentMetadata>
          <tML:UID>tML000009</tML:UID>
          <tML:Definition>A enumerated list used to indicate the type of
component.</tML:Definition>
          <tML:Description>Aggregate means the component is made up of other
components.
                    Basic means the component is atomic
                  A component is any structural part of a schema.</tML:Description>
          <tML:ComponentType>Basic</tML:ComponentType>
          <tML:Status>Active</tML:Status>
        </tML:ComponentMetadata>
      </xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Basic"/>
      <xsd:enumeration value="Aggregate"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:element name="ComponentMetadata">
    <xsd:annotation>
      <xsd:documentation xml:lang="en-GB">
        <tML:ComponentMetadata>
          <tML:UID>tML000010</tML:UID>
          <tML:Definition>The set of metadata required for any
component</tML:Definition>
          <tML:ComponentType>Aggregate</tML:ComponentType>
          <tML:Status>Active</tML:Status>
        </tML:ComponentMetadata>
      </xsd:documentation>
    </xsd:annotation>
    <xsd:complexType>
      <xsd:sequence>
        <xsd:annotation>
          <xsd:documentation xml:lang="en-GB">
            <tML:ComponentMetadata>
              <tML:UID>tML000011</tML:UID>
              <tML:Definition>A unique identifier of the component</tML:Definition>
              <tML:Description>This will become the primary key for the component
                 within the Global Telecommunications Data Dictionary.
                 It is made up of two parts:
                   - three alpha-numeric character indicating the group responsible
                     for component, such as an Standards Development Organisation.
                   - A six digit number.</tML:Description>
              <tML:ComponentType>Basic</tML:ComponentType>
              <tML:Status>Active</tML:Status>
            </tML:ComponentMetadata>
          </xsd:documentation>
        </xsd:annotation>
        <xsd:element name="UID"/>
        <xsd:element name="Definition">
          <xsd:annotation>
            <xsd:documentation xml:lang="en-GB">
              <tML:ComponentMetadata>
                <tML:UID>tML000012</tML:UID>
                <tML:Definition>A short sentence indicating the semantic meaning of
the component</tML:Definition>
                <tML:ComponentType>Basic</tML:ComponentType>
                <tML:Status>Active</tML:Status>
              </tML:ComponentMetadata>
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:element name="Description" minOccurs="0">
          <xsd:annotation>
            <xsd:documentation xml:lang="en-GB">
              <tML:ComponentMetadata>
                <tML:UID>tML000013</tML:UID>
                <tML:Definition>An optional textual description of the
component.</tML:Definition>
                <tML:Description>This may include examples and general
remarks</tML:Description>
                <tML:ComponentType>Basic</tML:ComponentType>
                <tML:Status>Active</tML:Status>
              </tML:ComponentMetadata>
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:element name="Source" minOccurs="0">
          <xsd:annotation>
            <xsd:documentation xml:lang="en-GB">
```

```
              <tML:ComponentMetadata>
                <tML:UID>tML000014</tML:UID>
                <tML:Definition>The source of the component</tML:Definition>
                <tML:Description>This would only be included alongside the component
                     if the source is different from the source of indicated in the
                      SchemaMetadata</tML:Description>
                <tML:ComponentType>Basic</tML:ComponentType>
                <tML:Status>Active</tML:Status>
              </tML:ComponentMetadata>
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:element name="Type" type="tML:ComponentType" default="Aggregate">
          <xsd:annotation>
            <xsd:documentation xml:lang="en-GB">
              <tML:ComponentMetadata>
                <tML:UID>tML000015</tML:UID>
                <tML:Definition>The indicator of whether the component is Aggregate
or Basic</tML:Definition>
                <tML:ComponentType>Basic</tML:ComponentType>
                <tML:Status>Active</tML:Status>
              </tML:ComponentMetadata>
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:element name="Status" type="tML:StatusType">
          <xsd:annotation>
            <xsd:documentation xml:lang="en-GB">
              <tML:ComponentMetadata>
                <tML:UID>tML000016</tML:UID>
                <tML:Definition>The status of the component with regards to the life
cycle of a component</tML:Definition>
                <tML:Description>The actual values of this component will be defined
by the GTDD project.
                      The only allowed values until this point are Active and
inActive</tML:Description>
                <tML:ComponentType>Basic</tML:ComponentType>
                <tML:Status>Active</tML:Status>
              </tML:ComponentMetadata>
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:simpleType name="StatusType">
    <xsd:annotation>
      <xsd:documentation xml:lang="en-GB">
        <tML:ComponentMetadata>
          <tML:UID>tML000017</tML:UID>
          <tML:Definition>A enumerated list indicating the allowed values of the a
Status</tML:Definition>
          <tML:Description>Active means that the definition is still in use.
               Inactive means that the definition is either not fully agreed or is
no longer to be used.
                Inactive terms are retained for backward
compatability</tML:Description>
          <tML:ComponentType>Basic</tML:ComponentType>
          <tML:Status>Active</tML:Status>
        </tML:ComponentMetadata>
      </xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Active"/>
```

```
     <xsd:enumeration value="Inactive"/>
   </xsd:restriction>
  </xsd:simpleType>
</xsd:schema>
```

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series B	Means of expression: definitions, symbols, classification
Series C	General telecommunication statistics
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	**TMN and network maintenance: international transmission systems, telephone circuits, telegraphy, facsimile and leased circuits**
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks and open system communications
Series Y	Global information infrastructure and Internet protocol aspects
Series Z	Languages and general software aspects for telecommunication systems

23170

Printed in Switzerland
Geneva, 2003