



INTERNATIONAL TELECOMMUNICATION UNION

**TELECOMMUNICATION  
STANDARDIZATION SECTOR**

STUDY PERIOD 2001-2004

**COM 4 – LS 17 – E**

**English only**

**Original: English**

**Question(s):** 12/4, 17/4, 18/4

27 October - 7 November 2003

**Ref. : TD 42 Rev. 3 (PLEN)**

**Source:** Rapporteur, Q.12/4

**Title:** Structured probable cause

---

**LIAISON STATEMENT**

**To:** ITU-T SG11, SG13, SG15, SG17, ETSI TISPAN WG8, ETSI for forwarding to 3GPP, T1M1, IETF DISMAN WG, TeleManagement Forum

**Approval:** Agreed to at SG 4 meeting, Geneva, 27 October - 7 November 2003

**For:** Action by 26 April 2004

**Deadline:**

**Contact:** **Knut Johannessen, Q12/4 Rapporteur**  
Telenor  
Norway

**Tel: +47 90 10 18 10**  
**Fax: +47 940 53 977**  
**Email: knut-hakon.johannessen@telenor.com**

SG4 has received contributions on structured probable cause. These contributions were reviewed in a joint meeting between Q12/4, Q14/4 and Q17/4 at the Study Group 4 meeting held 27 October - 7 November 2003 and with the following result:

There is general agreement that the information conveyed using the structured probable cause is valuable and to *document* the structured probable cause.

It is suggested that the structured probable cause would first be described at the protocol neutral level and then mapped to the various protocol specific designs. In this manner, the existing mechanisms for communication of probable cause could be retained.

It was further suggested that the proposal is really a design pattern that could be documented in the methodology (similar to the <<notifyDispatch>> stereotype).

There is agreement that the structured probable cause should not break existing solutions. When progressing the work on structured probable cause, SG4 will strive to develop specifications that will not impact interoperability for existing solutions. It has been suggested that it might be useful to separate between presentation of the structured probable cause in a GUI (and the processing within systems) and how information about probable cause is communicated on the wire.

Please find enclosed COM4-D145 containing contribution on general principles for the proposed structured probable cause.

---

**Attention:** Some or all of the material attached to this liaison statement may be subject to ITU copyright. In such a case this will be indicated in the individual document.  
Such a copyright does not prevent the use of the material for its intended purpose, but it prevents the reproduction of all or part of it in a publication without the authorization of ITU.



---

**Questions:** 12/4, 17/4

27 October - 7 November 2003

**STUDY GROUP 4 – DELAYED CONTRIBUTION 145****Source:** NORTEL NETWORKS (CANADA)**Title:** STRUCTURE PROBABLE CAUSES – GENERAL PRINCIPLES

---

**1 Introduction**

This proposal contains an update and refinement to the alarm information to be found in and based on X.733[5], X.721 [4] and M.3100[1][2].

The proposal suggests an extension and a structuring of the list of "probable causes". The extension is necessary to add to the list of actual probable causes those that have been found necessary in actual network elements as well as to accommodate probable causes specified in other standards bodies (TMF814, 3GPP, and IETF). Structuring is suggested in order to make these easier to interpret (by humans and machines) and to control the combinatory explosion that can occur in certain areas (e.g. Performance Monitoring (PM) thresholds). This is done in a way that is compatible with existing systems. This proposal defines both a structure and an encoding. The intention is to submit this proposal to other standards bodies in addition to ITU-T.

Existing issues and general introductions to the structure and encoding are presented here as an introduction to the proposed amendments described in a separate proposal. This document describes the principles, separate submissions define the details [Structure Probable Causes – X.733 Amendment - Probable Cause Text][Structured Probable Causes - M.3100 Amendment – Probable Cause Text].

**1.1 Issues with Existing Probable Causes**

The list of probable causes is defined in section 8.1.2.1 of [5], section 14.2 of [4] and section 10.2 of [1][2]. These have been used for many years to identify the underlying problem referenced in an alarm message. The probable cause is a very valuable field as it describes the condition that some component (object instance) is experiencing. This information enables an operator to begin the process of diagnosis in order to fix the underlying problem. The alarm message also contains the object instance which describes the precise component where the condition was detected. There are also other useful fields (time, severity, etc.) but the probable cause and object instance are the critical fields that define the problem and the precise item where the problem was detected. Given

---

**Contact:** Sharon Chisholm  
Nortel Networks (Canada)  
Canada

Tel: (613) 765-6067

Fax:

Email: schishol@nortelnetworks.com

**Attention:** This is not a publication made available to the public, but an **internal ITU-T Document** intended only for use by the Member States of the ITU, by ITU-T Sector Members and Associates, and their respective staff and collaborators in their ITU related work. It shall not be made available to, and used by, any other persons or entities without the prior written consent of the ITU-T.

the importance of the probable cause, it is valuable to have a relatively fixed list of these probable causes. This list enables automated applications to exploit the meaning and a common language for fault processing activities that avoids unnecessary differences (e.g. just in textual description) and duplication.

Vendors have been attempting to map the "alarm texts" that they tend to produce to probable cause values for many years. For some time, however, various problems have been encountered:

1. Often the closest probable cause meaning is very vague compared with the original text. This leads to loss of information in the interests of having a standard list. In particular there are insufficient probable causes to deal with protection and timing problems. E.g. "Unstable Sync Status Message" would map to timingProblem or synchronizationSourceMismatch. SynchronisationSourceMismatch is not an accurate mapping, whereas timingProblem is very vague (even though the original message is not vendor specific).
2. This loss of information also has the problem that the <object instance><probable cause> pair are not sufficient to match clear notifications with set notifications when alarm indications are to be cleared (as two underlying scan points have been collapsed to one).
3. Many of the probable causes are unnecessarily technology specific. E.g. ExcessiveBER can only be used where the technology is bit based. Excessive error rate would be applicable to any technology. Whether the detection is based on packet or bit could be additional information.
4. There has been significant technology advancement since the original list. It is very difficult for standards to support the addition of new probable causes in a non-disruptive way, in a timely fashion to keep up with the needs of new technology and equipment.
5. Developers of new devices have a hard time searching through the lists of existing probable causes to determine which are appropriate for their device. The large number, the flat list format, and the technology specific nature makes it virtually un-navigable due to its lack of structure.

In summary the list needs extending such that "alarm texts" can be mapped to an entry in the list while maintaining precision. Ideally the mapping would be 1-1. Clearly a standard cannot define all the items that may be suffering a condition; however it could list the standard "conditions" that could be experienced. Examples of the latter are fail, mismatch, and suspect.

## 1.2 Rationale for Structured Probable Causes

This proposal suggests extending the list of probable causes and the addition of structure to the probable cause field, which partitions a probable cause into components where the values one of the components is restricted to some degree.

A break down of the probable cause into <condition>.<attribute effected by condition> is the first step. Note that the attribute is really an attribute of the object instance, which is elsewhere in the alarm information. Examples would be:

mismatch.replaceableUnit  
mismatch.pathTrace  
fail.replaceableUnit  
fail.timingModule

This breakdown is suitable for simple alarms. Threshold crossing alarms for Performance Monitoring (PM) parameters are, however, more complex. An example PM "alarm text" might be "RX SES FE 24H" meaning that the PM parameter counting "severely errored

seconds" (SES) for the receiver (RX) at the far end (FE) has crossed a 24 hour (24H) threshold. Thus the text really represents a combination of <parameter> (SES, ES, BBE, UAT), <direction> (TX, RX), <location> (NE, FE), and <time period> (24H, 15M, etc.). Certain high level interfaces (e.g. TMF814, 3GPP) require these to be retrieved as individual values from this field. It would be useful, therefore, to be able to break the <attribute effected by condition> into components in cases like this. Note that subsequent qualifiers are siblings, and not children, of the previous qualifiers. Hence the above alarm text would map to the structured probable cause:

thresholdCrossed.QOS(param=SES.Direction=RX.Location=FE.TimePeriod=24H)

The other reason for this detailed structure is to control the combinatory explosion that would otherwise occur (in certain circumstances). This is achieved by standardizing a phrase with 4 fields, each of which has 2 to 4 values rather than  $2*2*4*2$  values. It is accepted that there is only value in doing this where there would otherwise be a combinatory explosion.

It should be noted that the structured probable cause remains a single item. One comment could be that a higher level OSS would have to be quite complex to interpret this structure. Although this is true, an OSS is not compelled to interpret the structure. The OSS could consider the whole structured probable cause as a single item, as it may currently do. I.e. the OSS has the option to continue to treat the structured probable cause as a single item or to get additional value by breaking it into parts. See section 6.2 for details on the mechanism for encoding this information.

It is accepted that many operators believe that there are too many texts and that these are difficult to deal with in an operational environment. In many cases operators would prefer more correlation to root-cause and service affected. This proposal recognizes this but does not attempt to solve it directly. Instead it attempts to provide a rigorous set of possible structured texts that vendors of equipment and applications could use when appropriate at different layers of the management system. The focus is on precision. It is expected and hoped that this effort will help application developers reduce the amount of alarms that the customer needs to be aware of. This in turn will provide value to operators by providing more correlation to root-cause and service affected.

In summary this document proposes that the list of probable causes is extended so that "alarm texts" can be mapped such that information is not lost and the mapping from an alarm condition to a probable cause is 1-1. In addition, the probable causes will be structured so that they can be more easily broken up into parts. The value of structuring them is as follows:

- Easier for operators and designers to understand the meaning.
- Control of combinatory explosion. Instead of standardizing  $N*M*P*Q$  combinations, we standardize  $N+M+P+Q$  components.
- Mapping to high level interfaces that require these items to be broken out (e.g. TMF814 with PM threshold alarms).
- Enables tighter control of certain parts (condition) while leaving other parts open-ended (<attribute> when <condition> is "fail"). This provides extensibility of the list of probable causes while maintaining control over the structure and the key aspects of the entries.
- Accommodates simple applications that treat the whole "probable cause" as a single text string, and sophisticated applications that can break the "probable cause" into parts.

## 2.1 CCITT Recommendations| International Standards

- 1 CCITT Recommendation M.3100 (1995): 1995, *Generic Network Information Model*.

- 2 CCITT Recommendation M.3100 Amendment 2 (1999): 1999, *Generic Network Information Model*.
- 3 CCITT Recommendation X.720 (1992) | ISO/IEC 10165-1: 1992, *Information technology - Open Systems Interconnection - Structure of management information: Management information model*.
- 4 CCITT Recommendation X.721 (1992) | ISO/IEC 10165-2: 1992, *Information technology - Open Systems Interconnection - Structure of management information: Definition of management information*.
- 51 CCITT Recommendation X.733 (1992) | ISO/IEC 10164-4: 1992, *Information technology - Open Systems Interconnection - Systems Management: Alarm reporting function*.
- 6 CCITT Recommendation X.736 (1992) | ISO/IEC 10164-7: 1992, *Information technology - Open Systems Interconnection - Systems Management: Security alarm reporting function*
- 7 ITU-T Recommendation X.680 (2002) | ISO/IEC 8824-1:2002, *Abstract Syntax Notation One (ASN.1): Specification of Basic Notation*.
- 8 Telecommunications Management Forum, *Multi-Technology Network Management Solution Set NML-EML Interface version 2.1 (TMF 814)*.
- 9 3GPP Specification 32111-2-330 v3.3.0 (2000), *Technical Specification Group Services and System Aspects; Telecommunication Management; Fault Management; Part 2: Alarm Integration Reference Point; Information Service Version 1*.
- 10 JSR 90 (2002), *OSS/J Quality of Service Interface*.
- 11 GSM 12.11, *Maintenance of the Base Station System*.

## 2.2 Definitions

For the purposes of this Recommendation | International Standard, the following definitions apply:

**Error**—a deviation of a system from normal operation.

**Fault**—the physical or algorithmic cause of a malfunction; faults manifest themselves as errors.

**Alarm**—a notification, of the form defined by this function, of a specific event. An alarm may or may not represent an error.

**Alarm Detection Point**—the entity that detected the alarm.

## 2.3 Conventions

When describing formal syntax the following notational conventions are used:

<X> To indicate that “X” is required.

[Y] To indicate that “Y” is optional.

P | Q To indicate either “P” or “Q”.

These symbols can be used in conjunction, for instance:

[O -- <R>] means that the entire combination is optional, but that if present R is required.

### 3 Proposal

#### 3.1 Proposed Structure of Probable Cause

The proposed structure of the probable cause is as follows:

<probableCause>=<condition>.<qualified attribute that condition effects>.[<additional information>]

where <condition> = {fail|mismatch|suspect|etc} (This list is defined in section 8.1.2.2.2)

<qualified attribute that condition effects>= <affected attribute>|  
<affected attribute>(<qualifier>[.<qualifier>]\*)

and

<affected attribute> is a string representing the attribute (e.g. circuitPack)  
<qualifier> is either a string or of the form <name>=<value>, where <name> and <value> are strings.

<additional info>=<additional info item>|  
(<additional info item>[.<additional info item>]\*)

and

<additional info item> is either a string or of the form <name>=<value>, where <name> and <value> are strings.

#### Examples of structured probable causes

Structured Probable Cause	Basic Probable Cause	M.3100 Integer Value
fail.replaceableUnit	replaceableUnitProblem	69
mismatch.trailTrace	pathTraceMismatch	13
thresholdFatal.errorRate.basis=bit	excessiveBER	12
thresholdCrossed.timePeriodParam( Param=SES. Direction=RX. Location=FE. TimePeriod=24H)	a specific case of thresholdCrossed	549

#### Notes on structured probable cause

1. Technology specific items should be accommodated by additional info, but can also be accommodated by qualification of the attribute, if necessary for uniqueness.
2. The event type is formally excluded from the probable cause. However it is often convenient to reference the pair. The conventional notation is to reference the pair as follows <eventType>.<probableCause>. An example would be equipment.fail.circuitPack.
3. The <condition>.<qualified attribute that condition effects> must be sufficient for uniqueness (without the <additional info>).

4. It is expected that new values will need to be standardized as time moves on. In the meantime vendors will on occasion find that they cannot assign a structured probable cause to an existing problem. There are two different circumstances when this will be the case:
  - a. When a structured probable cause text is not in the list by accidental omission (but it makes sense to standardize this in future).
  - b. When the probable cause is vendor specific.

It is suggested that the string FS\_ (for future standard) be used in the first case, and VS\_ (for vendor specific) be used in the second case.

### **3.2 Structured Probable Cause Encoding**

Currently the definitions in X.721 and M.3100 are in terms of definitions of enumerated type values (integer) in ASN.1. This proposal suggests using a structured text type on interoperability interfaces. This text will be an engineering mnemonic text similar to the enumerated type names (which are already based on English). It is structured so that it is machine readable and can be used on a machine to machine interface. There are a number of reasons for replacing numbers with structured text as follows:

- The management of number assignment is avoided (currently different standards have used the same number for different probable causes).
- The text is human interpretable, leading to more clarity of meaning.
- The text itself is structured in a flexible way meaning that the ASN.1 definition does not change as texts are added or structured. Note how the ASN.1 does not change as interpreters are designed to exploit the structure within the text string that is the probable cause.

The text can also be displayed, for human readability, where this is of value to the operator. When displayed, it can be displayed in other languages. This proposal defines the display texts for English (which are the same as the engineering mnemonics used on the interface). It does not define display texts for other languages but allows for them. The ASN.1 in X.721 and M.3100 will add an attribute probableCauseText wherever probableCause exists. This attribute will use the cstring type of ASN.1.

### **2.4 Backwards compatibility**

The probableCauseText field will be used by existing systems in the following manner while migration to this new field occurs:

1. Existing applications use the integer value probableCause.
2. This proposal adds probableCauseText as a structured string value.
3. New applications that understand these values should read the probableCauseText. If this is null or not present they should read the probableCause (as a number) and process according to the existing meanings.
4. New applications that set these values should set the probableCauseText attribute according to this proposal, and set the probableCause field according to the best value available in the existing list.