HM # 90/29

802.1 TECH.

PLENARY

PRESENTATION ON

OSI STRUCTURE OF MANAGEMENT INFORMATION STANDARDS PARTS 1 & 4

- INFORMATION MODEL
- GDMO

TONY ~~JEFFREE~~

MARCH 14

DP 10165-1 : SMI INFORMATION MODEL

KEY CONCEPTS :-

- MANAGED OBJECTS PROVIDE THE "MANAGEMENT VIEW" OF A RESOURCE BY MEANS OF THE ATTRIBUTES, OPERATIONS AND NOTIFICATIONS VISIBLE AT THE OBJECT BOUNDARY.

- THE OBJECT BOUNDARY PROVIDES AN ARCHITECTURAL DISTINCTION BETWEEN THE OBJECT CLASS DEFINERS' (LAYER GROUPS) WORK AND THE DEFINITIONS OF THE REMAINDER OF SYSTEMS MANAGEMENT.

- AN OBJECT CLASS SPECIFICATION CONSISTS OF:

  - ATTRIBUTES VISIBLE AT THE BOUNDARY;
  - OPERATIONS AND NOTIFICATIONS AVAILABLE AT THE BOUNDARY;
  - CONDITIONAL PACKAGES;
  - BEHAVIOUR DEFINITIONS;
  - INHERITED SPECIFICATIONS;
  - ALLOMORPHIC CAPABILITY.

- SUCH SPECIFICATION IS ACHIEVED BY MEANS OF TEMPLATES. A MANAGED OBJECT THAT SUPPORTS THE SPECIFICATION CONTAINED IN SUCH A TEMPLATE IS SAID TO BE A MEMBER OF THE CLASS THAT IS DEFINED BY THE TEMPLATE.

- OBJECT CLASSES ARE DEFINED BY A PROCESS OF SPECIALISATION WHICH ADDS TO OR MODIFIES THE SPECIFICATION INHERITED FROM ONE OR MORE SUPERCLASSES. ALL SUBCLASSES ARE ULTIMATELY DERIVED FROM THE ROOT OF THE INHERITANCE HIERARCHY, KNOWN AS TOP.

- ALLOMORPHISM ALLOWS A SUBCLASS TO BE DEFINED THAT MAY, UNDER SOME CIRCUMSTANCES & RESTRICTIONS, RESEMBLE THE BEHAVIOUR OF ONE OR MORE OF ITS SUPERCLASSES.

- ATTRIBUTES ARE "DATA-LIKE" ASPECTS OF AN OBJECT THAT MAY BE READ AND/OR MODIFIED. THEY MAY BE SINGLE-VALUED OR SET-VALUED.

NAMING & CONTAINMENT

- AN OBJECT EXISTS ONLY IF ITS CONTAINING OBJECT EXISTS. THE APEX OF THE CONTAINMENT HIERARCHY IS KNOWN AS ROOT.

- THE CONTAINMENT RELATIONSHIP BETWEEN OBJECTS IS USED FOR NAMING.

- THE NAME OF AN OBJECT CONSISTS OF:
  - THE NAME OF ITS SUPERIOR OBJECT;
  - AN INFORMATION THAT UNIQUELY IDENTIFIES IT WITHIN THE SCOPE OF ITS SUPERIOR OBJECT.

RDN ≡

- POSSIBLE SUPERIOR OBJECTS ARE DEFINED BY MEANS OF NAME BINDINGS. THE COLLECTION OF THESE NAMING RULES IS KNOWN AS A NAMING SCHEMA.

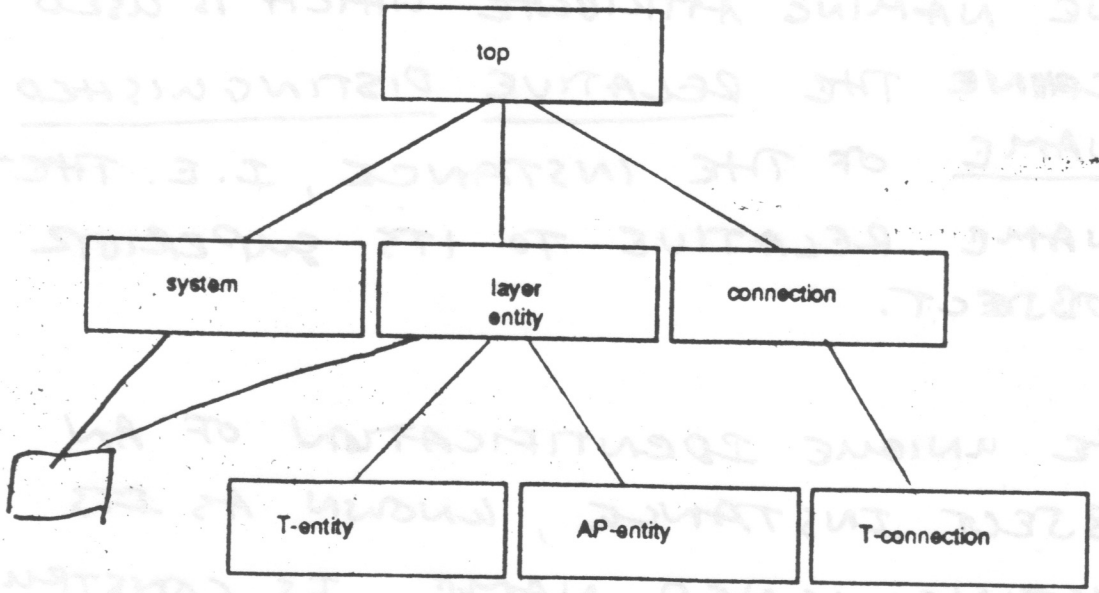
- OBJECT CLASSES AND ATTRIBUTES ARE IDENTIFIED BY ASN.1 "OBJECT IDENTIFIERS"

- OBJECT INSTANCES INCLUDE AT LEAST ONE NAMING ATTRIBUTE WHICH IS USED TO DERIVE THE RELATIVE DISTINGUISHED NAME OF THE INSTANCE, I.E. THE NAME RELATIVE TO ITS SUPERIOR OBJECT.

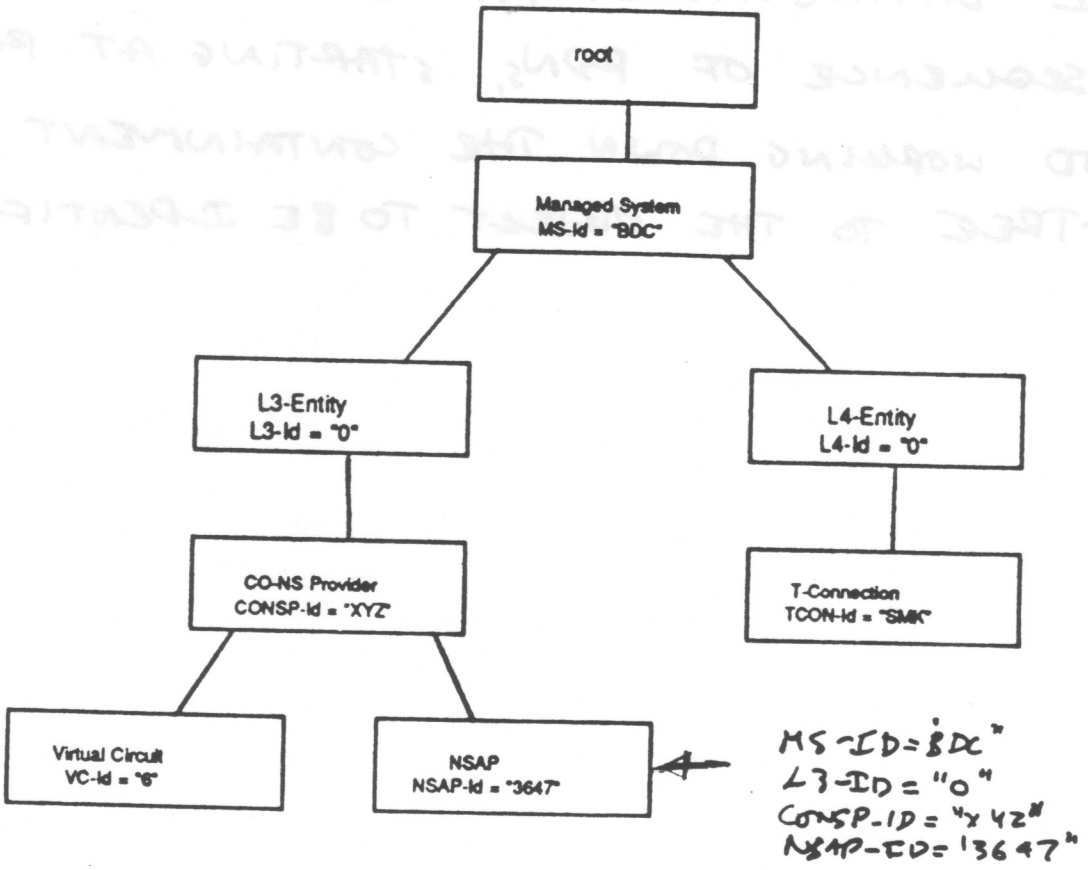
- THE UNIQUE IDENTIFICATION OF AN OBJECT INSTANCE, KNOWN AS ITS DISTINGUISHED NAME, IS CONSTRUCTED BY CONCATENATING ITS RDN WITH THE DISTINGUISHED NAME OF ITS SUPERIOR OBJECT.

- THE DISTINGUISHED NAME IS THEREFORE A SEQUENCE OF RDNs, STARTING AT ROOT AND WORKING DOWN THE CONTAINMENT TREE TO THE OBJECT TO BE IDENTIFIED.

INHERITANCE EXAMPLE :-



CONTAINMENT EXAMPLE :-



DP 10165-4 SMI PART 1: GUIDELINES FOR THE DEFINITION OF MANAGED OBJECTS

KEY CONCEPTS:-

- PROVIDES GENERAL GUIDANCE AS TO THE PRINCIPLES TO BE USED WHEN DEFINING M.O.s;
- PROVIDES COMMENTARY ON GLOBAL ISSUES SUCH AS CONFORMANCE, OPTIONALITY, etc.;
- DESCRIBES ITS RELATIONSHIP TO OTHER OSI MANAGEMENT STANDARDS;
- DEFINES A NOTATION FOR THE DEFINITION OF MANAGED OBJECTS;
- PROVIDES THE MEANS WHEREBY SMI PART 1 MAY BE TRANSLATED INTO REAL OBJECT SPECIFICATIONS.

GDMO NOTATIONAL TOOLS:

- PROVIDE A FORMAL SPECIFICATION "LANGUAGE" BASED ON THE CONCEPT OF TEMPLATES;
- M.O. DEFINITION ACHIEVED BY "FILLING IN" THE TEMPLATES IN AN APPROPRIATE MANNER;
- TEMPLATES PROVIDE A LINK TO THE ASN.1 DATA TYPE DEFINITIONS THAT ARE REQUIRED BY CMIP;
- TEMPLATES THEMSELVES ARE NOT WRITTEN IN ASN.1 NOTATION.

GDMO CONVENTIONS:

[ ] SURROUND OPTIONAL STRINGS

[ ]* SURROUND OPTIONAL STRINGS THAT MAY BE REPEATED

< > SURROUND STRINGS THAT MUST BE REPLACED ON A PER-INSTANCE BASIS

KEYWORDS IN UPPER CASE

| DELIMIT CHOICES

SPACE / CR DELIMIT STRINGS

SUPPORTING PRODUCTIONS:-

definition-label → syntactic-definition

LABELS:-

[<document-identifier>:]<label-string>

- document-identifier must be unique;
- label-string must be unique within docum

```
    [DERIVED FROM          <immediate-superclass-label>
                           [,<immediate-superclass-label>]* ;
    ]
    [POLYMORPHIC SET       <superclass-label>
                           [,<superclass-label>]* ;
    ]
    [CHARACTERIZED BY:

        [BEHAVIOUR DEFINITIONS <behaviour-definition-label>
                               [,<behaviour-definition-label>]* ;
        ]
        [ATTRIBUTES            <attribute-label> <propertylist> [<specific-error-label>]
                               [,<attribute-label> <propertylist> [<specific-error-label>]]* ;
        ]
        [GROUP ATTRIBUTES      <group-label> [<attribute-label>]*
                               [,<group-label> [<attribute-label>]*]* ;
        ]
        [OPERATIONS
            [CREATE            [<modifier>[,<modifier>]][<specific-error-label>] ;
            ]
            [DELETE            <delete-modifier>[<specific-error-label>] ;
            ]
            [ACTIONS           <action-label>[<specific-error-label>]
                               [,<action-label>[<specific-error-label>]]* ;
            ]
        ]
        [NOTIFICATIONS <notification-label>[<specific-error-label>]
                       [,<notification-label>[<specific-error-label>]]* ;
        ]
        [PACKAGE       <package-label> PRESENT IF <condition-definition> ;
        ]*
    ]
REGISTERED AS <object-identifier> ;

supporting productions

propertylist ->        [SET TO DEFAULT
                       DEFAULT VALUE <value-definition>]
                       [GET | REPLACE | GET-REPLACE]
                       [ADD | REMOVE | ADD-REMOVE]

modifier ->            with-reference-object | with-automatic-instance-naming

delete-modifier ->     only-if-no-contained-objects | deletes-contained-objects
```

PACKAGES: -

```
<package-label> CONDITIONAL PACKAGE

    [BEHAVIOUR DEFINITIONS <behaviour-definition-label>
                           [,<behaviour-definition-label>]* ;
    ]
    [ATTRIBUTES            <attribute-label> <propertylist> [<specific-error-label>]
                           [,<attribute-label> <propertylist> [<specific-error-label>]]* ;
    ]
    [GROUP ATTRIBUTES      <group-label> [<attribute-label>]*
                           [,<group-label> [<attribute-label>]*]* ;
    ]
    [OPERATIONS
        [CREATE            [<modifier>[,<modifier>]][<specific-error-label>] ;
        ]
        [DELETE            <delete-modifier>[<specific-error-label>] ;
        ]
        [ACTIONS           <action-label>[<specific-error-label>]
                           [,<action-label>[<specific-error-label>]]* ;
        ]
    ]
    [NOTIFICATIONS         <notification-label>[<specific-error-label>]
                           [,<notification-label>[<specific-error-label>]]* ;
    ]

REGISTERED AS <object-identifier> ;
```

ATTRIBUTES :-

<attribute-label> ATTRIBUTE

[DERIVED FROM <attribute-label> ;]
[WITH ATTRIBUTE SYNTAX <syntax-label> ;]
[MATCHES FOR <qualifier> [, <qualifier>]* ;]
[PERMITTED VALUES <range-syntax-label> ;]
[BEHAVIOUR <behaviour-definition-label> ;]

[REGISTERED AS <object-identifier> ;]

supporting productions

qualifier -> Equality | Ordering | Substrings | Set Comparison | Set Intersection

GROUPS :-

<group-label> GROUP ATTRIBUTE

GROUP ELEMENTS <attribute-label> [, <attribute-label>]* ;

REGISTERED AS <object-identifier> ;

BEHAVIOUR :-

<behaviour-definition-label> BEHAVIOUR

DEFINED AS <behaviour-description> ;

ACTIONS :-

<action-label> ACTION

ACTION BEHAVIOUR <behaviour-definition-label> ;
[WITH DATA SYNTAX <syntax-label> ;]
[WITH RESULT[SYNTAX <syntax-label>] ;]

REGISTERED AS <object-identifier> ;

NOTIFICATIONS :-

<notification-label> NOTIFICATION

BEHAVIOUR <behaviour-definition-label> ;
[WITH DATA SYNTAX <syntax-label> ;]
[WITH RESULT[SYNTAX <syntax-label>] ;]

REGISTERED AS <object-identifier> ;

ERRORS :-

<specific-error-label> SPECIFIC ERROR

WITH ERROR SYNTAX <syntax-label> ;

REGISTERED AS <integer> | <object-identifier> ;

NAMING :-

<name-binding-label> NAME BINDING

SUBORDINATE OBJECT CLASS <class-label> ;
NAMED BY
SUPERIOR OBJECT CLASS <class-label> ;
WITH ATTRIBUTE <attribute-label> ;
[CONSTRAINTS <behaviour-definition-label> ;
]

REGISTERED AS <object-identifier> ;

Examples of Use of the Guidelines

A.1 Managed Object Class definition

```
ExampleObjectClass MANAGED OBJECT CLASS

    DERIVED FROM ISO/IEC 10165-2:top ;

    CHARACTERIZED BY:

        BEHAVIOUR DEFINITIONS     ExampleClassBehaviour ;
        ATTRIBUTES                ObjectName GET,
                                  QOS-Error-Cause GET ;
        OPERATIONS    CREATE with-automatic-instance-naming ;
                      DELETE deletes-contained-objects ;

REGISTERED AS {joint-iso-ccitt ms(9) object class(3) ObjectClass 1} ;
```

Note. In a real example of use of this template, {ObjectClass 1} would be replaced by the value of the OBJECT IDENTIFIER under which the object class definition had been registered.

A.2 Attribute definition

```
ObjectName ATTRIBUTE

    WITH ATTRIBUTE SYNTAX     AttributeModule.ObjectName ;
    MATCHES FOR               Equality ;

REGISTERED AS {joint-iso-ccitt ms(9) attribute(4) AttributeID 1} ;


QOS-Error-Cause ATTRIBUTE

    WITH ATTRIBUTE SYNTAX     AttributeModule.QOSErrorCause ;
    MATCHES FOR               Equality ;
    BEHAVIOUR                 QOSErrorBehaviour ;

REGISTERED AS {joint-iso-ccitt ms(9) attribute(4) AttributeID 2} ;
```

Note In a real example of use of this template, {AttributeID N} would be replaced by the value of the OBJECT IDENTIFIER under which the attribute definition had been registered.

A.3 Action definition

*** *Editor's Notes:*
(1) It would be nice to have one of these.

A.4 Notification definition

The Notification template:-

```
CommunicationError NOTIFICATION

    BEHAVIOUR                 CommunicationErrorBehaviour ;
    WITH DATA SYNTAX          EventModule.ErrorInfo ;
    WITH RESULT SYNTAX    EventModule.ErrorResult ;

REGISTERED AS             {joint-iso-ccitt ms(9) Notification(?) EventID 1} ;
```

A.5 Behaviour definition

QOSErrorBehaviour BEHAVIOUR

DEFINED AS The QOS Error Cause attribute indicates the reason for a failure in quality of service associated with the managed object.

Note: The relationship between the permitted attribute values and the operation of the object itself are defined by the behaviour definitions associated with the object class definition. ;

CommunicationErrorBehaviour BEHAVIOUR

DEFINED AS The CommunicationError notification is generated by the object class when a communication error is detected by the object. The notification may contain any combination of the parameters Probable Cause, Severity, Trend Indication, Backed Up Status, Diagnostic Info, Proposed Repair Action, Threshold Info, State Change and Other Info.

Note: The precise definition of what constitutes a communication error and the parameter values that apply is object class specific. In a practical example, this Behaviour definition could, for example, refer to pieces of specification in a base standard in order to specify the behaviour. ;

ExampleClassBehaviour BEHAVIOUR

DEFINED AS <....Description of object class behaviour, including:
- How its attributes attain particular values & what they mean;
- What circumstances cause notifications to be generated;
- Etc. > ;

A.6 ASN.1 modules

```
AttributeModule.DEFINITIONS ::= BEGIN

QOSErrorCause ::= INTEGER {
                  responseTimeExcessive (0),
                  queueSizeExceeded (1),
                  bandwidthReduced (2),
                  retransmissionRateExcessive (3) }

END

EventModule DEFINITIONS ::= BEGIN

IMPORTS
                  Severity, TrendIndication, ... FROM ...

ErrorInfo ::= SET{

            [0] ProbableCause OPTIONAL,
            [1] Severity OPTIONAL,
            [2] TrendIndication OPTIONAL,
            [3] BackedUpStatus OPTIONAL,
            [4] DiagnosticInfo OPTIONAL,
            [5] ProposedRepairAction OPTIONAL,
            [6] ThresholdInfo OPTIONAL,
            [7] StateChange OPTIONAL,
            [8] OtherInfo OPTIONAL }

ErrorResult ::= NULL

ProbableCause ::= ... - Etc.

END
```