Final draft EN 301 098 V1.1.1 (1998-05)

European Standard (Telecommunications series)

Telecommunications Management Network (TMN);
Scheduling function;
Support object classes

Reference

DEN/TMN-00035 (a6000ico.PDF)

Keywords

management, network, scheduling

ETSI

Postal address

F-06921 Sophia Antipolis Cedex - FRANCE

Office address

650 Route des Lucioles - Sophia Antipolis
Valbonne - FRANCE
Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16
Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Internet

secretariat@etsi.fr
http://www.etsi.fr
http://www.etsi.org

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETR 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available **free of charge** from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://www.etsi.fr/ipr).

Pursuant to the ETSI Interim IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETR 314 (or the updates on http://www.etsi.fr/ipr) which are, or may be, or may become, essential to the present document.

Foreword

This European Standard (Telecommunications series) has been produced by ETSI Technical Committee Telecommunications Management Network (TMN) and is now submitted for the Voting phase of the ETSI standards Two-step Approval Procedure.

Proposed national transposition dates	
Date of latest announcement of this EN (doa):	3 months after ETSI publication
Date of latest publication of new National Standard or endorsement of this EN (dop/e):	6 months after doa
Date of withdrawal of any conflicting National Standard (dow):	6 months after doa

1 Scope

For some management applications, activities and/or operations have to be scheduled. This scheduling can be dependent of various factors (e.g. time of day, type of day) and it can be repeated (e.g. daily, weekly, bi-weekly). Scheduling activities can also require more information than a simple on/off scheduling, if e.g. the scheduling can select one out of three or more alternatives.

To manage this type of scheduling a generic solution needs to be applied. This solution shall be useful for a number of different management applications.

The scope of the present document is to model the support functions for this generic solution at the NE/OS interface.

Requirements for temporary override and automatic/manual fall-back to a previous management schema are outside the scope of the present document.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, subsequent revisions do apply.
- A non-specific reference to an ETS shall also be taken to refer to later versions published as an EN with the same number.

[1] ITU-T Recommendation M.3100 (1995): "Generic network information model".

[2] ITU-T Recommendation X.208 (1993): "Specification of Abstract Syntax Notation One (ASN.1)".

[3] ITU-T Recommendation X.721 (1992): "Information technology - Open Systems Interconnection - Structure of management information: definition of management information".

[4] ITU-T Recommendation X.746 (1995): " Information technology - Open Systems Interconnection - Systems management: scheduling function".

[5] ITU-T Recommendation X.720 (1992): "Information technology - Open Systems Interconnection - Structure of management information: management information model".

3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

ASN.1	Abstract Syntax Notation One
NE	Network Element
OC	Object Class
OS	Operations System
SMO	Scheduled Managed Object
SO	Scheduler Object

4 Scheduling function

4.1 Functional requirements

Some management applications (e.g. charging or routing) need time/type of day dependent scheduling function. To manage these dependencies, the functional requirements are the following:

R.1 Scheduling information that is communicated to the scheduled object shall be independent of the action the scheduled object performs. The scheduler has no knowledge about this action.

R.2 Several independent schedulers can co-exist.

R.3 A scheduler can schedule several objects.

R.4 One SMO can be scheduled by several SO.

R.5 Scheduler shall be able to handle trigger points.

R.6 Scheduler shall be able to handle intervals.

R.7 Scheduler shall be able to schedule activities that have more than two states.

R.8 Scheduling shall be possible on base of type of day (e.g. weekend, Christmas, bank holiday).

R.9 At least daily and weekly periodicity shall be possible for the periodicity of trigger points and intervals.

R.10 It shall be possible to limit the duration of periodical repetitions.

R.11 Overlapping intervals shall be allowed. Precedence rules are therefore needed in order to decide which interval is active.

R.12 It shall be possible to retrieve from the scheduler all the objects it is scheduling.

Besides these requirements, it is assumed that the relations between the scheduler, the scheduling information and the corresponding actions to be performed are existing in the scheduled object.

4.2 References analysis

The scheduling function described in ITU-T Recommendation X.746 [4] provides a function that can schedule a number of activities within multiple managed objects by a single scheduler and is able to specify the time duration that the schedule is active.

This scheduler covers only a part of the functional requirement given in the subclause 4.1 and it is designed for starting and stopping of activities.

It could be possible to use it by straining the modelling to the maximum extent. It is not used for the following reasons:

- it is not possible to define, for every point in time, different activities for each SMO;
- the scheduler schedules activities between two states;
- the type of day dependency is not covered;
- overlapping intervals are not supported.

5 Conformance

In order to claim conformance to this specification, a system needs to support the management functions for all managed object classes defined in clauses 7 and 8 of the present document. Therefore the clauses 7 and 8 form the mandatory part of the present document.

6 Information model

6.1 Introduction to the object model

The scheduling function is realised by two entities, the *multiScheduler* and the *typeOfDayController*.

The OC multiScheduler extends the functionality of the interval scheduling and aperiodic scheduling as described in ITU-T Recommendation X.746 [4]. For information, refer to model description in clause 7 of that document.

For interval scheduling, the ITU-T scheduler allows the transition of one activity in a SMO between the active and inactive state. The multiScheduler allows the transition of an activity between several (two or more) states. Each such state is associated with an index value. The OC multiScheduler also allows overlapping intervals.

For aperiodic scheduling, the OC multiScheduler allows the triggering of (possibly different) activities depending on an index value. Aperiodic scheduling in the OC multiScheduler can also be used for operation scheduling, as described in ITU-T Recommendation X.746 [4].

For both types of scheduling, the possibility exists to schedule activities depending on the type of day, e.g. holidays, weekdays.

The OC typeOfDayController is used to group days into categories according to their type, e.g. 1st January can be classified as specialDay1.

6.2 Functional model

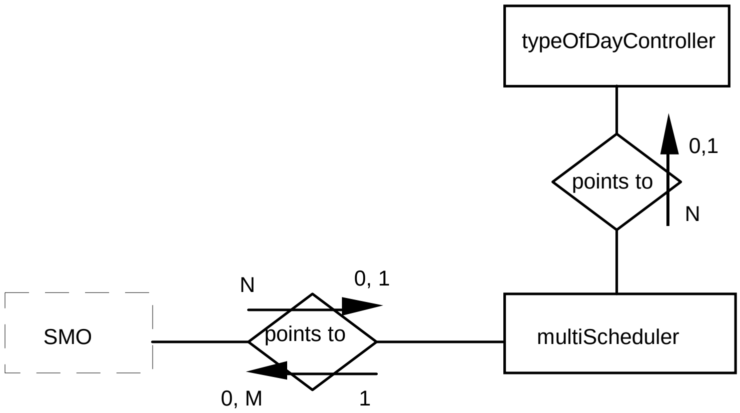


Figure 1: Scheduled selection E-R diagram

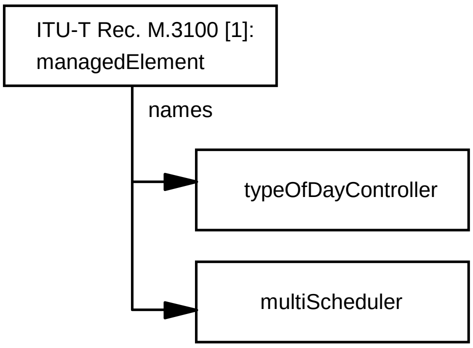


Figure 2: Naming Relations

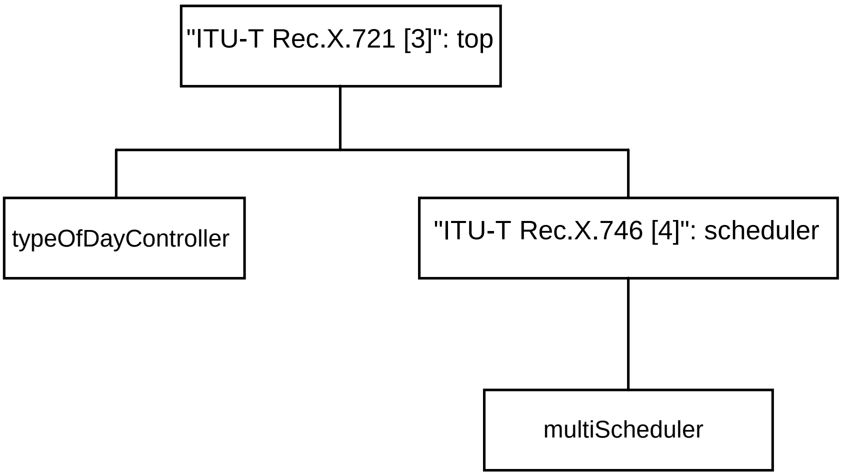


Figure 3: Inheritance tree

7 Information model description

This clause describes the object classes of the information model. For OC's descriptions the following table style is used.

Table 1

Object Class: "Object class name"			
Attributes	M/C	Value Set	Operation
Actions	M/C		
Notifications	M/C		

The column M/C indicates whether the information presented by the attributes/actions/notifications is mandatory(M) or conditional(C).

The column "Value Set" indicates whether the attribute is single-valued or set-valued (see ITU-T Recommendation X.720 [5], subclause 5.1.2.2).

The column "Operation" indicates the operations that are possible on the attribute.

7.1 multiScheduler

The OC multiScheduler is inherited from ITU-T Recommendation X.746 [4] scheduler. It provides the ability to control activities for which more information is required then a simple on/off scheduling. This OC multiScheduler allows the definition of multiple independent schedules, each of which is associated with an activity. These activities are associated to index values or are triggered by operations. In the case of index values, the association between a particular index value and an activity is defined within the SMO.

Table 2

Object Class: multiScheduler			
Attributes	M/C	Value Set	Operation
schedulingData	M	Set	GET-REPLACE ADD-REMOVE
defaultIndex	C	Single	GET-REPLACE REPLACE-WITH-DEFAULT
typeOfDayControllerInstance	C	Single	GET-REPLACE
Notifications			
"ITU-T Recommendation X.746 [4]": operationNotificationPackage	C		

The following attributes describe the OC multiScheduler:

- schedulingData;

  This attribute contains a set of schedules and related data controlling the activities;

- defaultIndex;

  This attribute gives the index that is applicable when none of the intervals specified in the attribute schedulingData is valid or when the scheduler is suspended or deleted;

- typeOfDayControllerInstance;

  This attribute identifies the instance of OC typeOfDayController that is relevant for this instance of OC multiScheduler.

7.2 typeOfDayController

The OC typeOfDayController provides management information needed to map a specific date or a weekday to a type of day and contains the currently valid typeOfDay.

Table 3

Object Class: typeOfDayController			
Attributes	M/C	Value Set	Operation
typeOfDayControllerId	M	Single	GET
currentTypeOfDay	M	Single	GET
dateTranslationList	M	Set	GET-REPLACE REPLACE-WITH-DEFAULT ADD-REMOVE
weekDayTranslationList	M	Set	GET-REPLACE REPLACE-WITH-DEFAULT
Notifications			
"ITU-T Recommendation M.3100 [1]: (1995)": objectManagementNotificationsPackage	M		

The following attributes describe the OC typeOfDayController:

- typeOfDayControllerId;

  This attribute is the object identifier attribute (RDN) of OC typeOfDayController;

- currentTypeOfDay;

  This attribute indicates the value of typeOfDay that is currently valid;

- dateTranslationList;

This attribute is a table that contains the mapping of dates to typeOfDay values;

- weekDayTranslationList;

  This attribute is a table that contains the mapping of weekdays to typeOfDay values.

8 Object class definitions

This clause contains the formal object class's definitions.

8.1 Managed object class definition

8.1.1 multiScheduler

```
multiScheduler  MANAGED OBJECT CLASS
    DERIVED FROM "Recommendation X.746 [4]:1995": scheduler;
    CHARACTERIZED BY
        multiSchedulerPackage PACKAGE
            BEHAVIOUR multiSchedulerPackageBeh BEHAVIOUR
                DEFINED AS
"See subclause 7.1.";;
            ATTRIBUTES
                schedulingData
                    GET-REPLACE
                    ADD-REMOVE;;
    CONDITIONAL PACKAGES
        intervalSchedulingPackage PACKAGE
            BEHAVIOUR intervalSchedulingPackageBeh BEHAVIOUR
                DEFINED AS
"Because the schedules defined in the attribute schedulingData are independent, the possibility
exists that none of the intervals are valid. In that situation, the value indicated by the default
index is applicable.
The following restrictions apply to the schedulingData attribute:
- index is the only valid component of IndexOrOperSpec,
- intervalsOfDayWps is the only valid component of TimeOfDayWps.
When the interval scheduler is created or resumed at a time that is within the scheduler's defined
duration, the activity within the SMO will be set according to the index defined by the interval
scheduler's schedule. When the interval scheduler is deleted or suspended, or the scheduler exists
at a time outside the scheduler's defined duration, the activity within the SMO will be set
according to the default index.";;
            ATTRIBUTES
                defaultIndex
                    REPLACE-WITH-DEFAULT
                    DEFAULT VALUE ASN1TypeModule.defaultDefaultIndex
                    GET-REPLACE;
        REGISTERED AS {package 1};
        PRESENT IF "Instance provides interval scheduling",
        triggerSchedulingPackage PACKAGE
            BEHAVIOUR triggerSchedulingPackageBeh BEHAVIOUR
                DEFINED AS
"The following restrictions apply to the schedulingData attribute:
- TimesOfDayWps shall have the component triggerTimes;
- the optional component priority shall be absent.
When the trigger scheduler is created or resumed at a time that is within the scheduler's defined
duration, the first triggering of an activity within the SMO will occur according to the schedule.
When the trigger scheduler is deleted or suspended, or the scheduler exists at a time outside the
scheduler's defined duration, the scheduled activities within the SMO will not be triggered.";;
        REGISTERED AS {package 2};
        PRESENT IF "Instance provides trigger scheduling",
        typeOfDayControllerInstancePackage
            BEHAVIOUR typeOfDayControllerInstancePackageBeh BEHAVIOUR
                DEFINED AS
"The attribute typeOfDayControllerInstance identifies the instance of OC typeOfDayController that is
relevant for this instance of OC multiScheduler.";;
            ATTRIBUTES
                typeOfDayControllerInstance
                    GET-REPLACE;
        REGISTERED AS {package 3};
PRESENT IF "More than one instance of OC typeOfDayController can exist in managedElement and if
scheduler instance uses 'type of day' scheduling",
"ITU-T Recommendation X.746 [4]":operationNotificationPackage
PRESENT IF "the results of an operation need to be reported or an operation performed is a GET
operation.";
REGISTERED AS {managedObjectClass 1};
```

8.1.2 typeOfDayController

```
typeOfDayController MANAGED OBJECT CLASS
    DERIVED FROM "ITU-T Recommendation X.721 [3]:1992":top;
    CHARACTERIZED BY
        typeOfDayControllerPackage PACKAGE
            BEHAVIOUR   typeOfDayControllerPackageBeh BEHAVIOUR
                DEFINED AS
"As in subclause 7.2. The mapping is maintained in the attributes ‘dateTranslationList’ and
‘weekDayTranslationList’.
The typeOfDay value that is specified for specific dates (via dateTranslationList attribute) has
precedence on the typeOfDay value for a week day (via weekDayTranslationList attribute). ";;
            ATTRIBUTES
                typeOfDayControllerId
                    GET,
                currentTypeOfDay
                    INITIAL VALUE DERIVATION RULE
                        currentTypeOfDayAlgorithm BEHAVIOUR
                            DEFINED AS
"The currentTypeOfDay attribute is determined in accordance to the value of the attributes
‘dateTranslationList’ and ‘weekDayTranslationList’.";;
                    GET,
                dateTranslationList
                    DEFAULT VALUE ASN1TypeModule.defaultDateTranslationList
                    REPLACE-WITH-DEFAULT
                    GET-REPLACE
                    ADD-REMOVE,
                weekDayTranslationList
                    DEFAULT VALUE ASN1TypeModule.defaultWeekDayTranslationList
                    REPLACE-WITH-DEFAULT
                    GET-REPLACE,

        "ITU-T Recommendation M.3100 [1]: 1995":objectManagementNotificationsPackage;;
REGISTERED AS {managedObjectClass 2};
```

8.2 Packages

This subclause contains the formal packages definitions used in several Ocs.

No common packages were identified.

8.3 Attributes

This subclause contains the formal attributes definitions.

8.3.1 currentTypeOfDay

```
currentTypeOfDay ATTRIBUTE
    WITH ATTRIBUTE SYNTAX ASN1TypeModule.TypeOfDay;
    MATCHES FOR EQUALITY;
REGISTERED AS {attribute 1};
```

8.3.2 dateTranslationList

```
dateTranslationList ATTRIBUTE
    BEHAVIOUR dateTranslationListBeh BEHAVIOUR
            DEFINED AS
"The translation list is a set that contains the mapping of a particular date into a type of day.
The typeOfDay can be either a weekday type ('workday' or 'weekend'), or one of the special days that
are defined. A specific date value may only occur once in dateTranslationList attribute.";;
    WITH ATTRIBUTE SYNTAX ASN1TypeModule.DateTranslationList;
    MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
REGISTERED AS {attribute 2};
```

8.3.3 defaultIndex

```
defaultIndex ATTRIBUTE
        BEHAVIOUR defaultIndexBeh BEHAVIOUR
            DEFINED AS
"As in subclause 7.1.";;
    WITH ATTRIBUTE SYNTAX ASN1TypeModule.DefaultIndex;
    MATCHES FOR EQUALITY;
REGISTERED AS {attribute 3};
```

8.3.4 schedulingData

```
schedulingData  ATTRIBUTE
    BEHAVIOUR schedulingDataBeh BEHAVIOUR
            DEFINED AS
"As in subclause 7.1.
Intervals within one member of the set shall be non-overlapping, but intervals defined in different
members of the set can overlap. In that case, the (optional) sequence member 'priority' shall be
present in each member of the set containing an interval that overlaps. The priority associated with
each member of the set determines which activity will be scheduled. The higher priority numerical
value has precedence on the lower one.";;
    WITH ATTRIBUTE SYNTAX ASN1TypeModule.SchedulingData;
    MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
REGISTERED AS {attribute 4};
```

8.3.5 typeOfDayControllerInstance

```
typeOfDayControllerInstance ATTRIBUTE
    BEHAVIOUR typeOfDayControllerInstanceBeh BEHAVIOUR
            DEFINED AS
                "This attribute points to an instance of OC typeOfDayController.";;
    WITH ATTRIBUTE SYNTAX ASN1TypeModule.ObjectInstance;
    MATCHES FOR EQUALITY;
REGISTERED AS {attribute 5};
```

8.3.6 typeOfDayControllerId

```
typeOfDayControllerId ATTRIBUTE
    WITH ATTRIBUTE SYNTAX ASN1TypeModule.NameType;
    MATCHES FOR EQUALITY;
REGISTERED AS {attribute 6};
```

8.3.7 weekDayTranslationList

```
weekDayTranslationList ATTRIBUTE
    BEHAVIOUR weekDayTranslationListBeh BEHAVIOUR
            DEFINED AS
"The translation list is a set that contains the mapping of a particular day of the week into a type
of day. The typeOfDay can be either a weekday type ('workday' or 'weekend'), or one of the special
days that are defined. In weekDayTranslationList attribute, there shall be one mapping for every
week day and a weekday value shall only occur once.";;
    WITH ATTRIBUTE SYNTAX ASN1TypeModule. WeekDayTranslationList;
    MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
REGISTERED AS {attribute 7};
```

8.4 Name bindings

This subclause contains the formal name bindings definitions.

8.4.1 multiScheduler-managedElement

```
multiScheduler-managedElement NAME BINDING
    SUBORDINATE OBJECT CLASS
            multiScheduler
            AND SUBCLASSES;
    NAMED BY SUPERIOR OBJECT CLASS
            "ITU-T Recommendation M.3100 [1]": managedElement
            AND SUBCLASSES;
    WITH ATTRIBUTE
            "ITU-T Recommendation X.746 [4]":schedulerID;
    CREATE
            WITH-REFERENCE-OBJECT,
            WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE;
REGISTERED AS {nameBinding 1}
```

8.4.2 typeOfDayController-managedElement

```
typeOfDayController-managedElement NAME BINDING
    SUBORDINATE OBJECT CLASS
            typeOfDayController
            AND SUBCLASSES;
    NAMED BY SUPERIOR OBJECT CLASS
            "ITU-T Recommendation M.3100 [1]": managedElement
            AND SUBCLASSES;
    WITH ATTRIBUTE
```

```
            typeOfDayControllerId;
    CREATE
            WITH-REFERENCE-OBJECT,
            WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE;
REGISTERED AS {nameBinding 2}
```

8.5 ASN.1 Type Definitions

```
ASN1TypeModule  {ccitt (0) identified-organisation (4) etsi (0) SchedulingInformationManagement
(301098) informationModel(0) asn1Module (2) asn1TypeModule (0)}
DEFINITION IMPLICIT TAGS ::=
BEGIN
IMPORTS
ObjectInstance
FROM CMIP-1 {joint-iso-ccitt ms(9) cmip(1) version1(1) protocol(3)}
NameType
FROM ASN1DefinedTypesModule {ccitt recommendation m gnm(3100) informationModel(0) asn1Module(2)
asn1DefinedTypesModule(0)}
OperationSpecifications, SequenceOfDays, SequenceOfMonths, SequenceOfWeeks, TimesOfDayWps
FROM Schedule-ASN1Module { joint-iso-ccitt ms(9) function(2) part15(15) modules(1) } ;
-- i.e. ITU-T Recommendation X.746 [4]
-- Important Remark: It is assumed that in ITU-T Recommendation X.746 [4] (1995),
--"TimesOfDay" should be read as "TimesOfDaysWps"
-- and that SequenceOfDays ::= SEQUENCE OF TimesOfDaysWps.
informationModel    OBJECT IDENTIFIER ::= {ccitt (0) identified-organisation (4) etsi (0)
SchedulingInformationManagement (301098) informationModel (0)}

managedObjectClass  OBJECT IDENTIFIER ::= {informationModel managedObjectClass (3)}
package OBJECT IDENTIFIER ::= {informationModel package (4)}
nameBinding OBJECT IDENTIFIER ::= {informationModel nameBinding (6)}
attribute   OBJECT IDENTIFIER ::= {informationModel attribute (7)}
defaultDateTranslationList  DateTranslationList ::= {}
defaultDefaultIndex DefaultIndex ::=0
defaultWeekDayTranslationList   WeekDayTranslationList ::= {
    {weekDay    monday, typeOfDay   workday},
    {weekDay    tuesday,    typeOfDay   workday},
    {weekDay    wednesday,  typeOfDay   workday},
    {weekDay    thursday,   typeOfDay   workday},
    {weekDay    friday, typeOfDay   workday},
    {weekDay    saturday,   typeOfDay   weekend},
    {weekDay    sunday, typeOfDay   weekend}}
Date                    ::= VisibleString (SIZE(8))  --i.e. YYYYMMDD
-- according to example in ITU-T Recommendation X.208 [2]
DateTranslationList             ::= SET OF SEQUENCE {
                date            [1] Date,
                typeOfDay   [2] TypeOfDay }
DefaultIndex            ::= INTEGER
IndexOrOperSpec         ::= CHOICE {
                index       [0] INTEGER,
                operation   [1] OperationSpecifications}
                            -- imported from ITU-T Recommendation X.746 [4]
Schedule                ::= CHOICE {
                daily       [0] SequenceOfDays,
                weekly      [1] SequenceOfWeeks,
                monthly     [2] SequenceOfMonths,
                typeOfDay   [3] SequenceOfTypeOfDays}
SchedulingData          ::= SET OF SEQUENCE {
                indexOrOperSpec [0] IndexOrOperSpec,
                schedule        [1] Schedule,
                priority            [2] INTEGER OPTIONAL}
SequenceOfTypeOfDays        ::= SEQUENCE SIZE(1) OF TypeOfDaysMaskWps
-- SIZE(1) because more has no meaning because there is no implied periodicity, contrary to
-- SequenceOfDays, SequenceOfWeeks, SequenceOfMonths. Therefore, this can be seen as the limit to
-- the parallelism with what is defined in ITU-T RecommendationX.746 [4].
TypeOfDay               ::= INTEGER {
                workday (0),
                weekend (1),
                specialDay1 (2),
                specialDay2 (3),
                specialDay3 (4),
                specialDay4 (5),
                specialDay5 (6),
                specialDay6 (7),
                specialDay7 (8),
                specialDay8 (9),
                specialDay9 (10),
                specialDay10    (11),
                specialDay11    (12),
                specialDay12    (13),
                specialDay13    (14),
                specialDay14    (15),
                specialDay15    (16),
                specialDay16    (17) }
TypeOfDays              ::= BIT STRING {
                workday (0),
                weekend (1),
                specialDay1 (2),
                specialDay2 (3),
                specialDay3 (4),
                specialDay4 (5),
                specialDay5 (6),
                specialDay6 (7),
                specialDay7 (8),
                specialDay8 (9),
                specialDay9 (10),
                specialDay10    (11),
                specialDay11    (12),
                specialDay12    (13),
                specialDay13    (14),
                specialDay14    (15),
                specialDay15    (16),
                specialDay16    (17) }
TypeOfDaysMaskWps           ::= SET OF SEQUENCE {
                typeOfDays  TypeOfDays,
                timesOfDay  TimesOfDayWps}
WeekDay             ::= ENUMERATED {
    sunday  (0),
    monday  (1),
    tuesday (2),
    wednesday   (3),
    thursday    (4),
    friday  (5),
    saturday    (6)}
WeekDayTranslationList      ::= SET OF SEQUENCE {
                weekday         [1] WeekDay,
                typeOfDay       [2] TypeOfDay}
END
```

Annex A (informative): Example of SMO and SO

In order to use the index scheduling mechanism, the SMO shall contain a scheduling attribute that provides the mapping of each index value to the appropriate activity in the SMO. The general form for the scheduling attribute is a SET OF SEQUENCE where the SEQUENCE contains an index component and another component (describing the activity) as simple as a numerical value or as complicated as a serie of operations.

Here follows an example:

OC A (SMO)

```
namingAttributeA
schedulingAttribute
    {   -- SET OF
        {   -- SEQUENCE
            index: 1,
            scheduledAttribute: B1
        },
        {   -- SEQUENCE
            index: 2,
            scheduledAttribute: B2
        },
        {   -- SEQUENCE
            index: 3,
            scheduledAttribute: B3
        }
    }
activeScheduledAttribute
OC B (pointed at OC)
namingAttributeB        (with possible values B1, B2, B3)
OC multiScheduler
.../... from X.746
schedulingData
    {   -- SET OF
        {   -- SEQUENCE
            indexOrOperSpec: index: 1,...
```

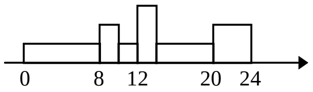
This attribute will contain scheduling information for three index values. For detailed examples see annex B.

Annex B (informative): Example of schedulingData values

B.1 Example 1: Shows several index values use

B.1.1 Interval solution

Every day from 8:00 to 10:00 and 20:00 to 24:00, use index 1. From 12:00 to 14:00, use index 2. Otherwise, use index 0.

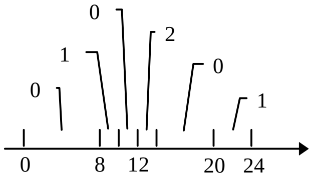


The attribute schedulingData would look like:

```
schedulingData:
     { -- SET OF
         { -- SEQUENCE
             indexOrOperSpec: index: 1,
             schedule: daily: -- daily, so
             { -- SEQUENCE OF TimesOfDayWps
                 intervalsOfDayWps:
                 { -- SET OF
                     {
                         intervalStart: { hour: 8 },
                         intervalEnd: { hour: 10 }
                     },
                    {
                         intervalStart: { hour: 20 },
                         intervalEnd: { hour: 0 }
                     }
                 }
             }
             -- no priority
         },
         { -- SEQUENCE
             indexOrOperSpec: index: 2,
             schedule: daily: -- daily, so
             { -- SEQUENCE OF TimesOfDayWps
                 intervalsOfDayWps:
                 { -- SET OF
                     {
                         intervalStart: { hour: 12 },
                         intervalEnd: { hour: 14 }
                     }
                 }
             },
             -- no priority
         }
     }
 defaultIndex: 0
```

NOTE: Priority is not required as intervals are not overlapping.

B.1.2 Trigger solution



The value of the attribute schedulingData would look like:

```
{   -- SET OF
    {   -- SEQUENCE
        indexOrOperSpec: index: 0,
        schedule: daily:    -- daily, so
        {   -- SEQUENCE OF TimesOfDayWps
            triggerTimes:
            {   -- SET OF Times24Wps
                {hour: 0},
                {hour : 10},
                {hour : 14}
            }
        }
        -- no priority
    },
    {   -- SEQUENCE
        indexOrOperSpec: index: 1,
        schedule: daily:    -- daily, so
        {   -- SEQUENCE OF TimesOfDayWps
            triggerTimes:
            {   -- SET OF Times24Wps
                {hour : 8},
                {hour : 20}
            }
        }
        -- no priority
    },
    {   -- SEQUENCE
        indexOrOperSpec: index: 2,
        schedule: daily:    -- daily, so
        {   -- SEQUENCE OF TimesOfDayWps
            triggerTimes:
            {   -- SET OF Times24Wps
                {hour : 12}
            }
        }
        -- no priority
    }
}
```

B.2 Example 2: Shows typeOfDay use

Index 3 for every workday from 0:00 to 6:00 and from 20:00 to 00:00, and for the weekends the entire day. Otherwise, use index 0.

B.2.1 Interval solution

```
schedulingData:

     { -- SET OF
         { -- SEQUENCE
             indexOrOperSpec: index: 3,
             schedule: typeOfDay: -- typeOfDay, so
             { -- SEQUENCE OF TypeOfDaysMaskWps
                 { -- SET OF
                     {
                         typeOfDays: { workday },
                         timesOfDay: intervalsOfDayWps:
                         { -- SET OF
                             {
                                 intervalStart: { hour: 0 },
                                 intervalEnd: { hour: 6 }
                             },
                             {
                                 intervalStart: { hour 20 },
                                 intervalEnd: { hour: 0 }
                             }
                         }
                     },
                     {
                         typeOfDays: { weekend },
                         timesOfDay: intervalsOfDayWps:
                         { -- SET OF
                             {
                                 intervalStart: { hour: 0 },
                                 intervalEnd: { hour: 0 }
                             }
                         }
                     }
                 }
             },
             -- no priority
         }
     },
 defaultIndex: 0
```

NOTE: Priority is not required as intervals are not overlapping.

B.2.2 Trigger solution

Trigger for index = 3 on workday 0:00 and 20:00 and trigger for index = 0 on workday 6:00. (Implicitly entire weekend will have index 3).

The value of the attribute schedulingData would look like:

```
    {   -- SET OF
        {   -- SEQUENCE
            indexOrOperSpec: index: 3,
            schedule: typeOfDay -- typeOfDay, so
            {   -- SEQUENCE OF TypeOfDaysMaskWps
                {   -- SET OF
                    {   -- SEQUENCE
                    typeOfDays: {workday},  --workday,
                    timesOfDay : triggerTimes:
                        {   -- SET OF Time24Wps
                            {hour: 20}
                        }
                    }
                }
            }
            -- no priority
        },
        {   -- SEQUENCE
            indexOrOperSpec: index: 0,
            schedule: typeOfDay -- typeOfDay, so
            {   -- SEQUENCE OF TypeOfDaysMaskWps
                {   -- SET OF
                    {   -- SEQUENCE
                    typeOfDays: {workday},  --workday,
                    timesOfDay : triggerTimes:
                        {   -- SET OF Time24Wps
                            {hour: 6}
                        }
                    }
                }
            }
            -- no priority
        }
}
```

B.3 Example 3: Shows overlapping intervals, implicit repetition (months periodicity) and mixed scheduler types

Use index 2 from the second day of the month starting at 8:00 until the fifth day of the month ending at 18:00, for every third month (March, June, September, December). Otherwise, use index 3 on Mondays and Fridays from 12:00 to 16:00. Otherwise, use index 0. The Monday/Friday setting has higher priority than the setting of index 2.

```
schedulingData:

     { -- SET OF
         { -- SEQUENCE
             indexOrOperSpec: index: 2,
             schedule: monthly:
             { -- SEQUENCE OF Monthmask
                 { -- SET OF
                     {
                         daysOfMonth:
                         {
                             daysFromFirst: { ‘’B },
                             daysFromLast: { ‘’B }
                         }
                         timesOfDaysWps: intervalsOfDayWps: {} -- empty set
                     }
                 }, -- January, April, July, and October
                 { -- SET OF
                     {
                         daysOfMonth:
                         {
                             daysFromFirst: { ‘’B },
                             daysFromLast: { ‘’B }
                         }
                         timesOfDaysWps: intervalsOfDayWps: {} -- empty set
                     }
                 }, -- February, May, August, and November
                 { -- SET OF
                     {
                         daysOfMonth:
                         {
                             daysFromFirst: {‘01’B }, -- 2nd day
                             daysFromLast: { ‘’B }
                         }
                         timesOfDaysWps: intervalsOfDayWps:
                         { -- SET OF
                             {
                                 intervalStart: { hour: 8 },
                                 intervalEnd: { hour: 0 }
                             }
                         }
                     },
                     {
                         daysOfMonth:
                         {
                             daysFromFirst: {‘0011’B }, -- 3rd, 4th day
                             daysFromLast: { ‘’B }
                         }
                         timesOfDaysWps: intervalsOfDayWps:
                         { -- SET OF
                             {
                                 intervalStart: { hour: 0 },
                                 intervalEnd: { hour: 0 }
                             }
                         }
                     },
                     {
                         daysOfMonth:
                         {
                             daysFromFirst: {‘00001’B }, -- 5th day
                             daysFromLast: { ‘’B }
                         }
                         timesOfDaysWps: intervalsOfDayWps:
                         { -- SET OF
                             {
                                 intervalStart: { hour: 0 },
                                 intervalEnd: { hour: 18 }
                             }
                         }
```

```
                    }
                }
            }
            priority: 1
        },
        { -- SEQUENCE
            indexOrOperSpec: index: 3,
            schedule: weekly:
            { -- SEQUENCE OF WeekMaskWps
                { -- SET OF
                    {
                        daysOfWeek: { ‘0100010’B } -- Monday and Friday
                        timesOfDayWps: intervalsOfDayWps:
                        { -- SET OF
                            {
                                intervalStart: { hour: 12 },
                                intervalEnd: { hour: 16 }
                            }
                        }
                    }
                }
            }
            priority: 2
        }
    },
defaultIndex: 0
```

History

Document history		
V1.1.1	October 1997	Public Enquiry PE 9807: 1997-10-17 to 1998-02-13
V1.1.1	May 1998	Vote V 9830: 1998-05-18 to 1998-07-31