The main purpose of this module is to provide a Configuration Service Constituent which may be instantiated and loaded into a User's Invocation Analyzer as a service.

The module defines one possible example design and implementation of the called side of a telephone communication.

The module brings together the functionality defined in the Network module, the Invocation module and the Abstract Telephone Communication module in addition to its own functionality.
Report overview
Chapter 1
Introduction

This document describes the Called Telephone Service OOram Module on the service sub-layer of the Service Constituent Creator Layer.

The main purpose of this module is to provide simple connections between two parties. A secondary purpose is to export models which are suitable for importing into more elaborate service definitions.

Figure 1. IN-Lab2 support value chain.
The arrows denote the using relationships between the modules. Notice that the User SM does not export to the Conceptual Schema, this means that the User objects are here a fixed part of the programs and not subject to configuration.
The models of this module are as follows:

1. **Import models**
   - *Invocation model (Inv2InvFW)*. Framework defining the invocation environment for all services.
   - *Abstract telephone connection model (Abstr2IntEXP)*. Mechanism defining the interaction between the calling and called sides of a telephone connection invocation.
   - *Network connection point (Netw2ConnSPC)*. Client-Server defining Connection Points in the Switching Domain.

2. **Role models**
   - *Called User interaction (TelB2UserRM)*. Role model defining the interaction between the called User and service.
   - *Access Point Selector (TelB2APSelRM)*. Role model defining the selection of a suitable termination point, which may either be a Network Access Point or a User for Call Forwarding.

3. **Object Type Specifications**
   - *Called Telephone Service SPC (TelB2SCSPC)*. Specifies the Called Telephone service object.

4. **Export models**
   - *TelB2 User-Service Interaction EXP (TelB2UserEXP)*. Mechanism describing the interaction between the called service and its User.
   - *Called Telephone Service CSC (TelB2SrvcCSC)*. Configuration Service Component for Called telephone service.
   - *Called Telephone selector CSC (TelB2SelCSC)*. Configuration Service Component for termination point selectors.

![Figure 2. Module model structure.](image-url)
Chapter 2
Called Telephone module export models summary

2.1 TelB2 User-Service Interaction EXP (TelB2UserEXP) {Export Model}

Report including the following selections automatically generated 28 July 1993: Export Models; Area of Concern stimuli; Role list diagram; Scenarios explanation

2.1.1 Area of Concern

Defines the interaction between the User object and the CalledTelephone service.

2.1.2 The Roles

![Diagram of TelB2 Srvc and User]

2.2 Called Telephone Service CSC (TelB2SrvcCSC) {Export Model}

Report including the following selections automatically generated 28 July 1993: Export Models; Area of Concern stimuli; Role list diagram; Scenarios explanation
2.2.1 Area of Concern

This model is a type specification for the called end of a telephone connection service.

2.2.2 The Roles

![Diagram of roles]

2.3 Called Telephone selector CSC (TelB2SelCSC) {Export Model}

Report including the following selections automatically generated 28 July 1993: Export Models; Area of Concern stimuli; Role list diagram; Scenarios explanation

2.3.1 Area of Concern

This model is a type specification for the called end of a telephone connection service.
2.3.2 The Roles
Chapter 3
Monitored executions

There are no monitored executions for this module.
Chapter 4
Called Telephone role models

4.1 Called Telephone Service User Interaction (TelB2UserRM) {Role Model}

Report including the following selections automatically generated 28 July 1993: Role Models; Area of Concern stimuli; Role list diagram, role explanation, role import map; Scenarios explanation; Message Sets port explanation, contract explanation, message explanation; FSM

4.1.1 Area of Concern

The minimal interaction between a User and its Calling Telephone Service.

4.1.2 The Roles
1. **TelB2User (Role)**. This object represents a Terminal which can accept or refuse a request for a full duplex, point-to-point connection in the Switch.

2. **TelB2Service (Role)**. This object represents a Service which can accept (or refuse) a request for a full duplex, point-to-point connection in the Switch.

### 4.1.3 Message Sets

1. **TelB2Service (Role)**
   - user (u) *(Port)*.
     - TelB2User<TelB2Service *(Contract)*.
       - openRequest *(Message)*. Return true iff it is OK to open a new connection.

### 4.2 Access Point Selector (TelB2APSelRM) *(Role Model)*

**Report including the following selections automatically generated 28 July 1993:** Role Models; Area of Concern *stimuli*; Role list *diagram, role explanation, role import map*; Scenarios *explanation*; Message Sets *port explanation, contract explanation, message explanation*; FSM

#### 4.2.1 Area of Concern

This role model describes an Access Point selector. The selection is based on parameters like day of week, time of day, and the busy-state of Access Points. For each of the candidate access points there may be a condition on day of week, time of day.
4.2.2 The Roles

1. **TelB2SelClient (TelB2SelClnt) {Role}**.
2. **Tel2SelSrv {Role}**. An object which may or may not select a connection point according to its selection criteria attributes.

4.2.3 Message Sets

1. **TelB2SelClient (TelB2SelClnt) {Role}**
   - selectors (s) {Port}.
   - **Tel2SelSrv<TelB2SelClnt {Contract}**.
     - **kind {Message}**. Return the kind of selector. Possible values are #APSelector or #CallForward.
     - **connID {Message}**. Return the selected ID, this is an Access Point ID in the case of #APSelector, and a UserID in the case of #CallForward.
     - **selectRequest {Message}**. Return a Boolean, true iff this selector’s conditions have been met.
Chapter 5
Called Telephone Object type specifications

5.1 Called Telephone Service SPC (TelB2SCSPC) {Type Specification}

Report including the following selections automatically generated 28 July 1993: Type Specifications; Area of Concern stimuli; Role list diagram, role explanation, role import map; Scenarios explanation; Message Sets port explanation, contract explanation, message explanation; FSM

5.1.1 Area of Concern

This model is a type specification for the called end of a telephone connection service.
5.1.2 The Roles
1. **TelB2Srcv** *(Type)*. This object represents a Service which can accept (or refuse) a request for a full duplex, point-to-point connection in the Switch.

This role represents the called part of a telephone connection service. It is responsible for establishing a Leg in the Switching Domain from the Called User to a Connection Point provided by the A-Service. We assume this functionality is encapsulated in the object, and do not specify how it accomplishes it.

The service instance, Netw2ConnClient, views the network as a virtual switch object and represents a user of the Connection Control.

The Service role represents the target application for the client. Any service to be invocated by this model must have an object which plays this role.

2. **TelB2User** *(Type)*. This object represents a Terminal which can accept or refuse a request for a full duplex, point-to-point connection in the Switch.

3. **Abstract Telephone Service-A (Abstr2AServ)** *(Type)*. This role represents the calling part of a telephone connection service. It takes the initiative to establish a service. It is responsible for establishing a Connection Point service in the Switching Domain, and for establishing a leg from the Calling User to this connection point. We assume this functionality is encapsulated in the object, and do not specify how it accomplishes it.

This object represents any client of the Invocation Manager, an Analyzer and possibly one of the Analyzer's Services.

4. **Netw2Connection Leg (Netw2ConnLeg)** *(Type)*. This role represents a communication path from a connection point towards an addressable entity in the network (e.g. an end user terminal). A Leg is always associated with a ConnectionPoint. Leg is responsible for storing the information characterizing the leg.

5. **Netw2Connection Point (Netw2ConnPoint)** *(Type)*. This role represents an interconnection or logical bridge allowing information to flow between legs.

6. **Tel2SelSrv** *(Type)*. An object which may or may not select a connection point according to its selection criteria attributes.

7. **InvManager (InvMngr)** *(Type)*. The Invocation Manager's role is to accept requests from Clients and start the Invocation Analyzer (IA) for further processing and to start specific services on request by the IA. The Invocation Analyzer is unique for a given User, and an Invocation Analyzer can be responsible for analyzing requests for many services. The Invocation Manager can be considered a part of the operating system of the network.

8. **Inv2Analzr** *(Type)*. The Invocation Analyzer must perform a client-dependent analysis that will identify the particular service requested. This implies that the Invocation Analyzer must analyze the information received from the client, look up relevant information in its internal databases, load the service object and finally initiate it with the required information.
<table>
<thead>
<tr>
<th>TelB2SCSPC</th>
<th>TelB2UserRM</th>
<th>Abstr2IntEXP</th>
<th>Netw2ConnSPC</th>
<th>TelB2APSelRM</th>
<th>Inv2InvFW</th>
</tr>
</thead>
<tbody>
<tr>
<td>TelB2Srcv</td>
<td>TelB2Service</td>
<td>Abstr2BServ</td>
<td>Netw2ConnClient</td>
<td>TelB2SelClnt</td>
<td>Inv2Serv</td>
</tr>
<tr>
<td>TelB2User</td>
<td>TelB2User</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Abstr2AServ</td>
<td>Abstr2AServ</td>
<td></td>
<td></td>
<td></td>
<td>Inv2Client</td>
</tr>
<tr>
<td>Netw2ConnLeg</td>
<td></td>
<td>Netw2ConnLeg</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Netw2ConnPoint</td>
<td></td>
<td>Netw2ConnPoint</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tel2SelSrv</td>
<td></td>
<td></td>
<td></td>
<td>Tel2SelSrv</td>
<td></td>
</tr>
<tr>
<td>InvMngr</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>InvMngr</td>
</tr>
<tr>
<td>Inv2Analzr</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Inv2Analzr</td>
</tr>
</tbody>
</table>

Table 3. Import relationships for Called Telephone Service SPC [RMTTypeSpecification].
Chapter 6
Called Telephone module export models

6.1 TelB2 User-Service Interaction EXP (TelB2UserEXP) {Export Model}

Report including the following selections automatically generated 28 July 1993: Export Models; Area of Concern stimuli; Role list diagram, role explanation; Scenarios explanation; Message Sets imports, port explanation, contract explanation, message explanation

6.1.1 Area of Concern

Defines the interaction between the User object and the CalledTelephone service.

6.1.2 The Roles

1. **TelB2Srcv [Type]**. This role represents the called part of a telephone connection service. It is responsible for establishing a Leg in the Switching Domain from the Called User to a Connection Point provided by the A-Service. We assume this functionality is encapsulated in the object, and do not specify how it accomplishes it.

2. **TelB2User [Type]**. This object represents a User (e.g. part of a Terminal) which can accept or refuse a request for a full duplex, point-to-point connection in the Switch.
6.1.3 Message Sets

1. TelB2Srcv \{Type\}
   - user (u) \{Port\}.
   - TelB2User<TelB2Service \{Contract\}.
     - openRequest \{Message\}. Return true iff it is OK to open a new connection.

6.2 Called Telephone Service CSC (TelB2SrvcCSC) \{Export Model\}

Report including the following selections automatically generated 28 July 1993: Export Models; Area of Concern stimuli; Role list diagram, role explanation; Scenarios explanation; Message Sets imports, port explanation, contract explanation, message explanation

6.2.1 Area of Concern

This model is a type specification for the called end of a telephone connection service.

6.2.2 The Roles

![Diagram of roles](image-url)
1. **Inv2Analzr** (**Type**). The Invocation Analyzer must perform a client-dependent analysis that will identify the particular service requested. This implies that the Invocation Analyzer must analyze the information received from the client, look up relevant information in its internal databases, load the service object and finally initiate it with the required information.

2. **TelB2Srcv** (**Type**). This object represents the calling part of a Service which can accept (or refuse) a request for a full duplex, point-to-point connection in the Switch.

### 6.2.3 Message Sets

1. **Inv2Analzr** (**Type**)
   - service (s) (**Port**).
     - **InvServ<InvAnalzr** (**Contract**).
       - **setUp**: parameters client: aClient analyzer: anAnalyzer invocator: anInvoc (**Message**). Create and return an instance of a Service with the given parameters (one of them being owner UserID, another being the current Client).
       - **kind** (**Message**). Return aSymbol, which represents the kind of service.
       - **protocol** (**Message**). Answer the Client protocol permitted by this service.

2. **TelB2Srcv** (**Type**)
   - invocationAnalyzer (ia) (**Port**).
     - **InvAnalzr<InvServ** (**Contract**).
       - **userID** (**Message**). Return the user ID of the user.
       - **user** (**Message**). Return the User object of the InvocationAnalyzer.

### 6.3 Called Telephone selector CSC (TelB2SelCSC) (**Export Model**)

*Report including the following selections automatically generated 28 July 1993*: Export Models; Area of Concern stimuli; Role list diagram, role explanation; Scenarios explanation; Message Sets imports, port explanation, contract explanation, message explanation

### 6.3.1 Area of Concern

This model is a type specification for the called end of a telephone connection service.
6.3.2 The Roles

1. **TelB2Srcv** *(Type)*. This object represents a Service which can accept (or refuse) the called part of a request for a full duplex, point-to-point connection in the Switch.

2. **Tel2SelSrv** *(Type)*. An object which may or may not select a connection point according to its selection criteria attributes.

6.3.3 Message Sets

1. **TelB2Srcv** *(Type)*
   - selectors (s) *(Port)*.
     - **Tel2SelSrv<TelB2SelClnt** *(Contract)*.
       - **kind** *(Message)*. Return the kind of selector. Possible values are #APSelector or #CallForward.
       - **connID** *(Message)*. Return the selected ID, this is an Access Point ID in the case of #APSelector, and a UserID in the case of #CallForward.
       - **selectRequest** *(Message)*. Return a Boolean, true iff this selector's conditions have been met.
Appendix 1
Implementations for monitored execution

App 1.1  Called Telephone Service SPC (TelB2SCSPC) {Type Specification}

Report including the following selections automatically generated 28 July 1993: Type Specifications; Implementation

App 1.1.1  Called Telephone Service SPC {Implementation}

RMSObject1 subclass: #Tel2SelSrv0
    instanceVariableNames: ""
    classVariableNames: ""
    poolDictionaries: ""
    category: 'ZZZInLab-TeleB2+'

Instance variables:

Subclasses should not implement methods for:

Subclasses must implement the following messages:
   Tel2SelSrv<TelB2SelClnt
       connID
       kind
       selectRequest

Inv2Serv1 subclass: #TelB2Srcv0
    instanceVariableNames: 'aService connectionPoint legs selectors user '
    classVariableNames: ""
    poolDictionaries: ""
    category: 'ZZZInLab-TeleB2+'

This object represents a Service which can accept (or refuse) a request for a full duplex, point-to-point connection in the Switch. This role represents the called part of a telephone connection service. It is responsible for establishing a Leg in the Switching Domain from the Called User to a Connection Point provided by the A-Service. We assume this functionality is encapsulated in the object, and do not specify how it accomplishes it. The service instance, Netw2ConnClient, views the network as a virtual switch object and represents a user of the Connection Control. The Service role represents the target application for the client. This is the software defining some client's telecommunications service. All services to be invocated by this model must have an object which plays this role.
Instance variables:
- aService <RMSObject1>
- connectionPoint <RMSObject1>
- legs <RMSObject1> No Explanation
- selectors <Tel2SelSrv0>
- user <RMSObject1>

Subclasses should not implement methods for:

Subclasses must implement the following messages:
- Tel2BServ<Tel2AServ
  - call:qoc:
  - connected
  - createLeg:direction:qoc:toConnPt:
  - open
  - releaseFromA

Tel2SelSrv0 subclass: #Tel2SelSrv1
  - instanceVariableNames: 'weekdays startTime endTime apCondition apIDs lastUsedIndex '
  - classVariableNames: ''
  - poolDictionaries: ''
  - category: 'ZZZInLab-TeleB2+'

AccessPointSelector for testing if its accesspoint is acceptable under the current conditions.

Instance variables:
- weekdays <SequenceableCollection of: String>
- startTime <String>
- endTime <String>
- apCondition <Symbol>
- apIDs <SequenceableCollection of: Integer>
- lastUsedIndex <Integer>

TelB2Srcv0 subclass: #TelB2Srcv1
  - instanceVariableNames: 'callType qualityOfConnection state apID '
  - classVariableNames: 'SelectorClasses '
  - poolDictionaries: ''
  - category: 'ZZZInLab-TeleB2+'

Tel2SelSrv1 class>>protocol (constants)
  - protocol
    self halt: 'illegal'.
    ^#APSelector

Tel2SelSrv1 class>>protocolConst (constants)
  - protocolConst
    ^#APSelector

Tel2SelSrv1 class>>tmInitialize (class initialization)
  - tmInitialize
    Inv2Install1 addService: self protocolConst asClass: self

TelB2Srcv1 class>>protocol (constants)
  - protocol
    self halt: 'illegal'.
    ^#TelB
TelB2Srcv1 class>>protocolConst (constants)
protocolConst
^#TelB

TelB2Srcv1 class>>initialize (class initialization)
initialize
" TelB2Srcv1 initialize"
SelectorClasses := IdentityDictionary new
  at: #APSelector put: Tel2SelSrv1;
  at: #CallForward put: Tel2SelSrv2;
  yourself

TelB2Srcv1 class>>tmInitialize (class initialization)
tmInitialize
  Inv2Install1 addService: self protocolConst asClass: self

Tel2SelSrv1>>connID (Tel2SelSrv<TelB2SelClnt)
connID
" Return the selected ID, this is an Access Point ID in the case of "
" #APSelector, and a UserID in the case of #CallForward."
  self halt.
  lastUsedIndex := lastUsedIndex + 1 \ apIDs size.
  ^apIDs at: lastUsedIndex.

Tel2SelSrv1>>kind (Tel2SelSrv<TelB2SelClnt)
kind
" Return the kind of selector. Possible values are #APSelector or #CallForward."
  ^self class protocolConst

Tel2SelSrv1>>selectRequest (Tel2SelSrv<TelB2SelClnt)
selectRequest
" Return a Boolean, true iff this selector's conditions have been met."
  | dat start end |
  dat := Time dateAndTimeNow.
  (weekdays includes: dat first weekday) ifFalse: [^false].
  start := Time readFrom: (ReadStream on: startTime asString).
  start > dat last ifTrue: [^false].
  end := Time readFrom: (ReadStream on: endTime asString).
  end < dat last ifTrue: [^false].
  ^true

Tel2SelSrv1>>initializeFrom: (access)
initializeFrom: selDict
  weekdays := selDict at: 'weekdays'.
  startTime := (selDict at: 'startTime') asString.
  endTime := (selDict at: 'endTime') asString.
  apCondition := selDict at: 'apCondition'.
  apIDs := selDict at: 'apIDs'.
  lastUsedIndex := 0.

Tel2SelSrv1>>initialize (private)
initialize
  super initialize.
  weekdays := RMSOrderedCollection new.
  startTime := '00:00'.
  endTime := '24:00'.
  apCondition := #RoundRobin.
  apIDs := RMSOrderedCollection new.
  lastUsedIndex := 0.
TelB2Srcv1>>apID (private)
apID
  ^apID

TelB2Srcv1>>callForward: (private)
callForward: userID
  self halt: 'Not implemented'

TelB2Srcv1>>getAccessPoint (private)
geAccessPoint
  ^invocationAnalyzer user

TelB2Srcv1>>initialize (private)
initialize
  super initialize.
  selectors := RMSOrderedCollection new.

TelB2Srcv1>>postCopy (private)
postCopy
  super postCopy.

TelB2Srcv1>>selectors (private)
selectors
  ^selectors

TelB2Srcv1>>setState: (private)
setState: aSymbol
  state := aSymbol.

TelB2Srcv1>>kind (InvServ<InvAnalzr)
kind
  " Return aSymbol, which represents the kind of service. "
  ^#TelB

TelB2Srcv1>>setUp:client:analyzer:invocator: (InvServ<InvAnalzr)
setUp: parameters client: aClient analyzer: anAnalyzer invocator: anInvoc
  " Create and return an instance of a Service with the given parameters "
  " (one of them being owner UserID, another being the current Client). "
  self setState: #idle.
  invocationAnalyzer := anAnalyzer.
  invocationManager := anInvoc.

TelB2Srcv1>>initializeFrom: (InvService<InvInstaller)
initializeFrom: aDict
  " 930706 trygve(4.1): IN-Lab Service Contract. "
  | xxx |
  apiID := aDict at: 'apID'.
  xxx := 0.
  (aDict at: 'selectors' ifAbsent: [#()]) do:
    [:selDict |
      xxx := xxx + 1.
      sel := (SelectorClasses at: (selDict at: 'kind')) new.
      sel initializeFrom: selDict.
      selectors add: sel].

TelB2Srcv1>>call:qoc: (Tel2BServ<Tel2AServ)
call: callTyp qoc: qoc
  " This message provides input to service B about the type of call A "
  " wants to make, and some information about the state of service A. "
  " Just relevant state information for service B to be able to proceed "
  " should be given. "
  " "
  " Parameters:"
  " callTyp = {#Simplex, #Duplex} "
  " qoc is Integer representing required bandwidth. "
callType := callTyp.
qualityOfConnection := qoc.

TelB2Srcv1>>connected (Tel2BServ<Tel2AServ)
connected
  " This message tells service B that connections have been established "
  " and that the call has been successfully completed. "
user connected

TelB2Srcv1>>createLeg:direction:qoc:toConnPt: (Tel2BServ<Tel2AServ)
createLeg: tp direction: dir qoc: qoc toConnPt: conPt
  | activeSelector xxx connID |
  xxx := 0.
  activeSelector := self selectors detect:
    [:sel |
      xxx := xxx + 1.
      sel selectRequest] ifNone: [nil].
  activeSelector notNil
  ifTrue:
    [| kind |
      connID := activeSelector connID.
      kind := activeSelector kind.
      kind = #APSelector
      ifTrue: [*conPt createLeg: tp direction: dir to: connID qoc: qoc] ifFalse: [
        kind = #CallForward
        ifTrue: [*self callForward: connID]
        ifFalse: [self dpsError: activeSelector kind , ' is illegal selector kind'. ^nil]]]
  ifFalse:
    [connID := self apID.
     self dpsTrace: (Array with: 'default' with: connID).
     ^conPt createLeg: tp direction: dir to: connID qoc: qoc]

TelB2Srcv1>>open (Tel2BServ<Tel2AServ)
open
  " Start the service process. "
  self setState: #idle.
  client provideContext.
  self setState: #presenting.
  client userBWaiting: self getAccessPoint.

TelB2Srcv1>>releaseFromA (Tel2BServ<Tel2AServ)
releaseFromA
  self halt: 'not implemented'
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter 1</td>
<td>Introduction</td>
<td>2</td>
</tr>
<tr>
<td>Chapter 2</td>
<td>Called Telephone module export models summary</td>
<td>4</td>
</tr>
<tr>
<td>2.1</td>
<td>TelB2 User-Service Interaction EXP (TelB2UserEXP) {Export Model}</td>
<td>4</td>
</tr>
<tr>
<td>2.1.1</td>
<td>Area of Concern</td>
<td>4</td>
</tr>
<tr>
<td>2.1.2</td>
<td>The Roles</td>
<td>4</td>
</tr>
<tr>
<td>2.2</td>
<td>Called Telephone Service CSC (TelB2SrvcCSC) {Export Model}</td>
<td>4</td>
</tr>
<tr>
<td>2.2.1</td>
<td>Area of Concern</td>
<td>5</td>
</tr>
<tr>
<td>2.2.2</td>
<td>The Roles</td>
<td>5</td>
</tr>
<tr>
<td>2.3</td>
<td>Called Telephone selector CSC (TelB2SelCSC) {Export Model}</td>
<td>5</td>
</tr>
<tr>
<td>2.3.1</td>
<td>Area of Concern</td>
<td>5</td>
</tr>
<tr>
<td>2.3.2</td>
<td>The Roles</td>
<td>6</td>
</tr>
<tr>
<td>Chapter 3</td>
<td>Monitored executions</td>
<td>7</td>
</tr>
<tr>
<td>Chapter 4</td>
<td>Called Telephone role models</td>
<td>8</td>
</tr>
<tr>
<td>4.1</td>
<td>Called Telephone Service User Interaction (TelB2UserRM) {Role Model}</td>
<td>8</td>
</tr>
<tr>
<td>4.1.1</td>
<td>Area of Concern</td>
<td>8</td>
</tr>
<tr>
<td>4.1.2</td>
<td>The Roles</td>
<td>8</td>
</tr>
<tr>
<td>4.1.3</td>
<td>Message Sets</td>
<td>9</td>
</tr>
<tr>
<td>4.2</td>
<td>Access Point Selector (TelB2APSelRM) {Role Model}</td>
<td>9</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Area of Concern</td>
<td>9</td>
</tr>
<tr>
<td>4.2.2</td>
<td>The Roles</td>
<td>10</td>
</tr>
<tr>
<td>4.2.3</td>
<td>Message Sets</td>
<td>10</td>
</tr>
<tr>
<td>Chapter 5</td>
<td>Called Telephone Object type specifications</td>
<td>11</td>
</tr>
<tr>
<td>5.1</td>
<td>Called Telephone Service SPC (TelB2SCSPC) {Type Specification}</td>
<td>11</td>
</tr>
<tr>
<td>5.1.1</td>
<td>Area of Concern</td>
<td>11</td>
</tr>
<tr>
<td>5.1.2</td>
<td>The Roles</td>
<td>12</td>
</tr>
<tr>
<td>Chapter 6</td>
<td>Called Telephone module export models</td>
<td>15</td>
</tr>
<tr>
<td>6.1</td>
<td>TelB2 User-Service Interaction EXP (TelB2UserEXP) {Export Model}</td>
<td>15</td>
</tr>
<tr>
<td>6.1.1</td>
<td>Area of Concern</td>
<td>15</td>
</tr>
<tr>
<td>6.1.2</td>
<td>The Roles</td>
<td>15</td>
</tr>
<tr>
<td>6.1.3</td>
<td>Message Sets</td>
<td>16</td>
</tr>
</tbody>
</table>
6.2    Called Telephone Service CSC (TelB2SrvcCSC) {Export Model}............ 16
6.2.1  Area of Concern.................................................................................. 16
6.2.2  The Roles.............................................................................................. 16
6.2.3  Message Sets......................................................................................... 17
6.3    Called Telephone selector CSC (TelB2SelCSC) {Export Model}.......... 17
6.3.1  Area of Concern.................................................................................... 17
6.3.2  The Roles.............................................................................................. 18
6.3.3  Message Sets......................................................................................... 18

Appendix 1  Implementations for monitored execution................................. 19
App 1.1   Called Telephone Service SPC (TelB2SCSPC) {Type Specification}.... 19
App 1.1.1 Called Telephone Service SPC {Implementation}............................. 19
INDEX

Tel2SelSrv1 class>>protocol (constants) .......................................................... 20
Tel2SelSrv1 class>>protocolConst (constants) .................................................. 20
Tel2SelSrv1 class>>tmInitialize (class initialization) ...................................... 20
Tel2SelSrv1>>connID (Tel2SelSrv<TelB2SelClnt) ........................................... 21
Tel2SelSrv1>>initialize (private) ................................................................. 21
Tel2SelSrv1>>initializeFrom: (access) ............................................................ 21
Tel2SelSrv1>>kind (Tel2SelSrv<TelB2SelClnt) ................................................. 21
Tel2SelSrv1>>selectRequest (Tel2SelSrv<TelB2SelClnt) ................................... 21
TelB2Srcv1 class>>initialize (class initialization) ....................................... 21
TelB2Srcv1 class>>protocol (constants) ......................................................... 20
TelB2Srcv1 class>>protocolConst (constants) ............................................... 21
TelB2Srcv1 class>>tmInitialize (class initialization) ...................................... 21
TelB2Srcv1>>apID (private) ........................................................................... 22
TelB2Srcv1>>call:qoc: (Tel2BServ<Tel2AServ) .............................................. 22
TelB2Srcv1>>callForward: (private) ............................................................... 22
TelB2Srcv1>>connected (Tel2BServ<Tel2AServ) ............................................. 23
TelB2Srcv1>>createLeg:direction:qoc:toConnPt: (Tel2BServ<Tel2AServ) ....... 23
TelB2Srcv1>>getAccessPoint (private) .......................................................... 22
TelB2Srcv1>>initialize (private) .................................................................... 22
TelB2Srcv1>>initializeFrom: (InvService<InvInstaller) ................................. 22
TelB2Srcv1>>kind (InvServ<InvAnalzr) ........................................................... 22
TelB2Srcv1>>open (Tel2BServ<Tel2AServ) ...................................................... 23
TelB2Srcv1>>postCopy (private) ..................................................................... 22
TelB2Srcv1>>releaseFromA (Tel2BServ<Tel2AServ) ......................................... 23
TelB2Srcv1>>selectors (private) ..................................................................... 22
TelB2Srcv1>>setState: (private) ................................................................... 22
TelB2Srcv1>>setUp:client:analyzer:invocator: (InvServ<InvAnalzr) ............... 22