Publicly Available Specification

PAS 0001-10-4

Version: 0.0.1

Date: 30 September 1997

Source: TETRAPOL Forum

Work Item No: 0001

Key word: TETRAPOL

TETRAPOL Specifications;
Part 10: Inter System Interface;
Part 4: ISI conformance testing specification

TETRAPOL FORUM

TETRAPOL Secretariat

Postal address: BP 40 78392 Bois d'Arcy CEDEX - FRANCE
Office address: Rue Jean-Pierre Timbaud 78392 Bois d'Arcy CEDEX - FRANCE
Tel.: +33 1 34 60 55 88 - Fax: +33 1 30 45 28 35

Contents

Foreword

This document is the Publicly Available Specification (PAS) of the TETRAPOL Land Mobile Radio System, which shall provide digital Narrow Band Voice, Messaging, and Data Services. Its main objective is to provide specifications dedicated to the more demanding PMR segment: the Public Safety. These specifications are also applicable to most PMR networks.

This PAS is a multipart document which consists of:

Part 1 General Network Design

Part 2 Radio Air Interface

Part 3 Air Interface Protocol

Part 4 Data-Gateway to X.400

Part 5 Interface to Dispatch Centre

Part 6 Line Connected Terminal Interface

Part 7 CODEC

Part 8 Radio conformance tests

Part 9 Air Interface Protocol conformance tests

Part 10 Inter System Interface

Part 11 Gateway to external networks

Part 12 Network Management Centre Interface

Part 13 User Data Terminal to System Terminal Interface

Part 14 Mobile Station and Base Station Simulators

Part 15 Data Gateway to External Data Terminal (EDT)

Part 16 Security

Part 17 Performances objectives

Part 18 Base station (BS) to Radioswitch (RSW) interface

Part 19 Stand Alone Dispatch Position (SADP) Interface

1. Scope

Interoperability between PMR systems requires a common inter-system interface. This network-to-network interface is referred to as the TETRAPOL ISI.

This document corresponds to sub-part 10.4 of the TETRAPOL inter-system interface which is divided into several sub-parts:

- Part 10.1 ISI Technical requirements
- Part 10.2 ISI Architecture
- Part 10.3 ISI Protocol design
- Part 10.4 ISI Conformance testing specification

This sub-part specifies the ISI-specific conformance testing specification, as an extension of the conformance specification for the QSIG protocol [12] [13]. It contains information as a basis to build a test suite structure and a test purpose for the TETRAPOL inter-system interface protocol [2].

The objective of this test specification is to provide a basis for approval test for the TETRAPOL ISI for switching and management infrastructures, to allow interoperability between different manufacturers equipments.

ISO 9646 is the methodology of conformance testing considered as a basis for the test methodology.

2. Normative references

This PAS incorporates by dated and undated reference, provisions from other publications. These normative references are cited at the appropriate places in the text and the publications are listed hereafter.

[1] PAS 0001-1: TETRAPOL Specifications; General network design.

[2] PAS 0001-10: TETRAPOL Specifications; Inter-system Interface.

[3] ETS 300 402: Integrated Services Digital Network (ISDN); Digital subscriber signalling system 1 (DSS1) protocol; Data link layer.

[4] ETS 300 172: Private Integrated Services Network (PISN); Inter-exchange signalling protocol; Circuit mode basic services. [ISO/IEC 11572 (1994) modified]

[5] ETS 300 239: Private Integrated Service Network (PISN); Inter-exchange signalling protocol; Generic functional protocol for the support of supplementary services. [ISO/IEC 11582 (1995), modified]

[6] ITU-T Recommendation X.25: Interface between data terminal equipment (DTE) and data circuit terminating equipment (DCE) for terminals operating in the packet mode and connected to public data networks by dedicated circuits.

[7] ITU-T Recommendation X.229: Remote operations; protocol specification

[8] ITU-T Recommendation X.209: Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1)

[9] ITU-T Recommendation Z.100: Specification and Description Language

[10] draft prETS 300 392-3 TETRA interworking (march 1997)

[11] ETS 300 303: "Integrated Services Digital Network (ISDN); ISDN - Global System for Mobile communications (GSM) Public Land Mobile Network (PLMN) signalling interface".

[12] prETS 300 805 Private Integrated Services Network (PISN); Inter-exchange signalling protocol; Circuit mode basic services; Network Layer (NL)

[13] prETS 300 806 Private Integrated Services Network (PISN); Inter-exchange signalling protocol; protocol; Generic functional protocol for the support of supplementary services; Network Layer (NL)

3. Definitions, symbols and abbreviations

3.1. Definitions

For the purposes of this PAS, the following definitions apply:

Bearer: A bearer provides the capability for information transfer in a SwMI, between network access points, using one information transfer mode.

Call: A voice service invocation

Call master SwMI: The switching and management infrastructure (SwMI) in charge of the synchronisation of the call establishment over ISI and the call release procedures between SwMIs.

Connection: An association of logical channels established to provide a means for user information transfer.

Destination SwMI: A SwMI to which a call is routed over the ISI from the call master SwMI.

Home SwMI: The individual home SwMI of a system terminal is the SwMI that is designated by the country code and network code of its long individual explicit address. The group home SwMI of a network group or of a regional opertional group is the SwMI that is designated by the country code and network code of its long group address. Federal groups have no unique home SwMI.

Home location register: Database containing subscription information, for the purpose of terminal location and service profile management.

Incoming call: A call that is originated outside of the local system. A call over ISI is said to be an incoming call for a SwMI when its establishment propagates from the ISI to the SwMI.

Local SwMI: A SwMI a system manager is in charge of.

Network: The switching and management infrastructure and the base stations within a system.

Originating SwMI: The SwMI in which the calling user is registered when requesting a call establishment

Outgoing call: A call that is originated from the local system. A call over ISI is said to be an outgoing call for a SwMI when its establishment propagates from the SwMI over the ISI.

Participating SwMI: A SwMI that is involved in the coverage of a call, as defined before the call activation

Protocol data unit: Combination of protocol control information and service data unit to support the flow for a service provided at the related layer of a protocol stack.

Radio terminal: System terminal connected to the infrastructure by a radio link, equivalent to the Mobile termination unit MTU.

Service data unit: Signalling data and user information created at the upper layers of the protocol stack and that shall be transferred using the service of the related layer. A service data unit is the payload contents of a protocol data unit.

SwMI: Switching and management infrastructure

System: A system comprises a network and a set of terminals

System manager: The manager in charge of the operation of a system

System Terminal: A service access reference point provided to the user by the System. System terminals ST are Radio terminals, Line connected terminals, Stand alone dispatch positions

Visited SwMI: A SwMI is referred to as the visited SwMI of a terminal, if the terminal is attached to this SwMI and this SwMI is not the home SwMI of the teminal.

Visitor location register: Database containing sub-parts of a subscription profile and location indication for terminals viisting the base network of a SwMI.

3.2. Abbreviations

For the purpose of ISI, the following abbreviations apply:

A/I	Air Interface
ANF	Additional Network Feature
APDU	Application Protocol Data Unit
ASN.1	Abstract Syntax Notation 1
BC	Broadcast call
BN	Base Network
CC	Call control or country code
DCE	Data circuit-terminating equipment
DLCI	Data link connection identifier
DTE	Data terminal equipment
GC	Group call
GLR	Gateway location register of a SwMI
GSW	Gateway switch
HLR	Home location register of a SwMI
IC	Individual call
ISI	Inter-system interface
MM	Mobility management
MOCH	Multisite open channel
NC	Network code of a SwMI in a country
OG	Operational group
PINX	Private Integrated Services Network Exchange
PISN	Private Integrated Services Network
PDU	Protocol data unit
PSS1	Private Signalling System Number 1
RT	Radio terminal
RSW	Radio switch, providing routeing capability to base stations
SAP	Service Access Point
SDL	Specification and Description Language
ST	System terminal
SwMI	Switching and management infrastructure of a system
VLR	Visitor location register in a base network of a SwMI

4. SDL model of the inter-system interface (informative)

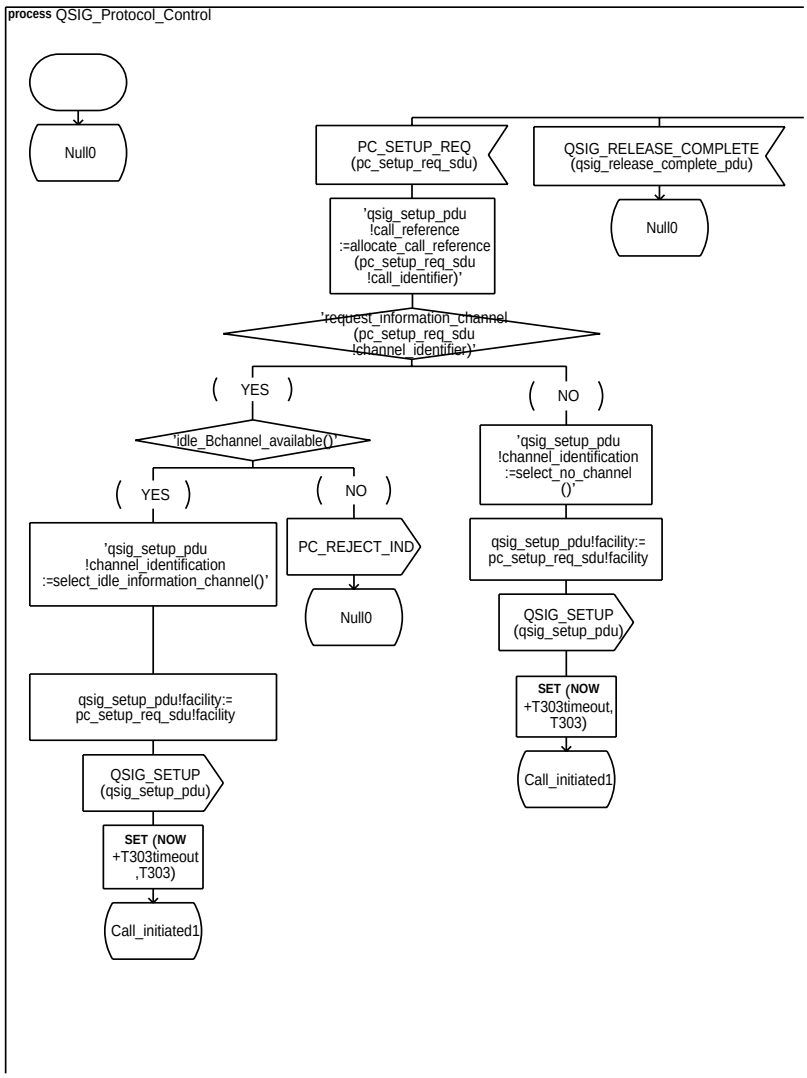
4.1. SDL model structure

The network layer of the inter-system interface relies on:
- a QSIG bearer service that handles the ISI conection control,
- several additional network features that handle the TETRAPOL service signalling.

QSIG protocol control and a generic ANF provide the dynamic behaviour of the interface as SDL automata [9], as described hereafter.

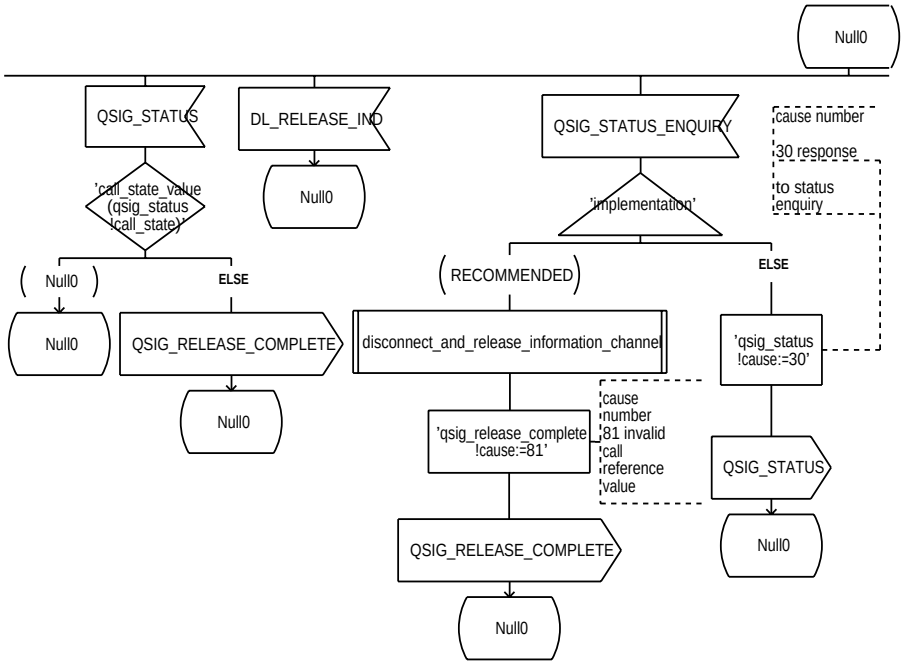
4.2. QSIG model

QSIG_Protocol_Control	Null0	Mon Aug 11 16:40:43 1997



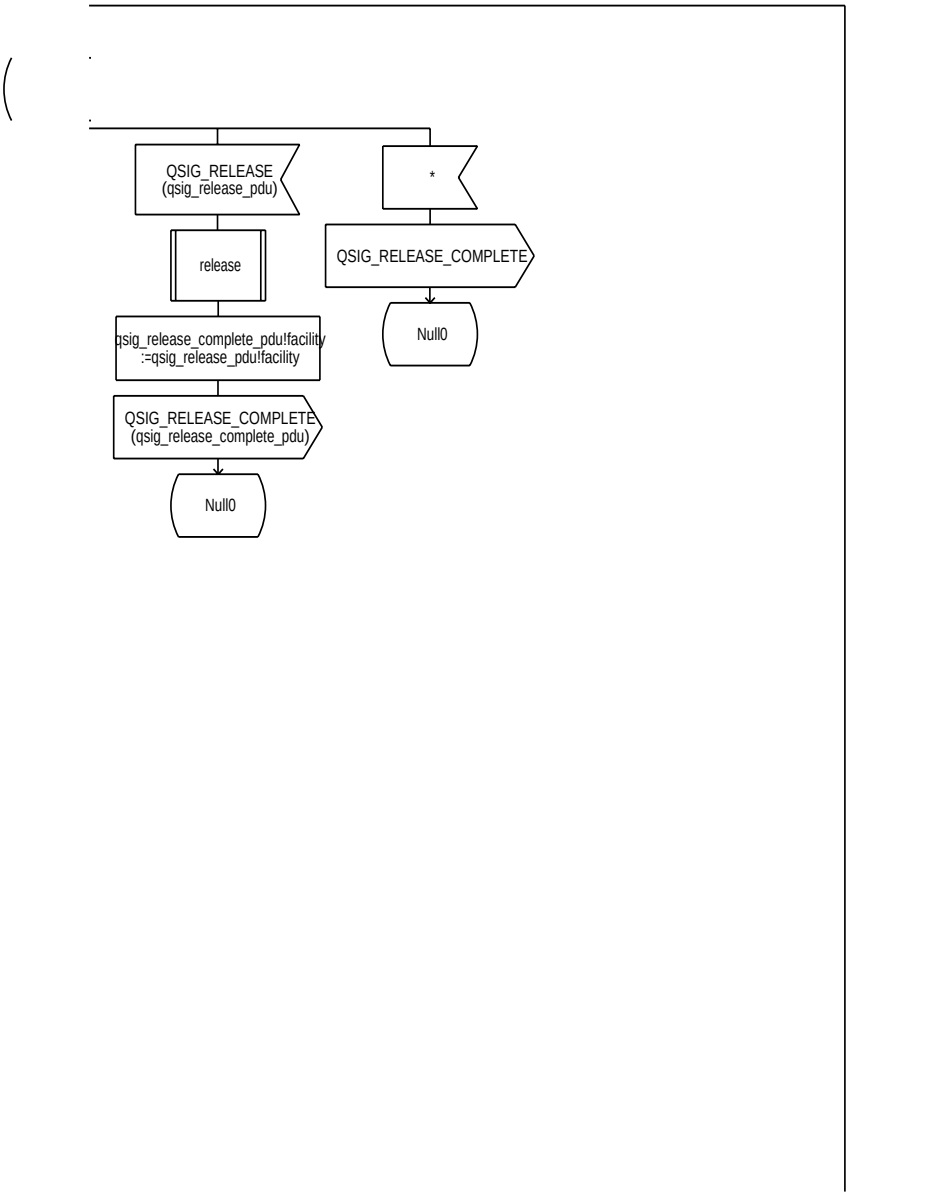
	/home/trimouli/GEODE/ANF2/QSIG1.pr	View: 1 / Page: 1

QSIG_Protocol_Control	Null0	Mon Aug 11 16:40:43 1997

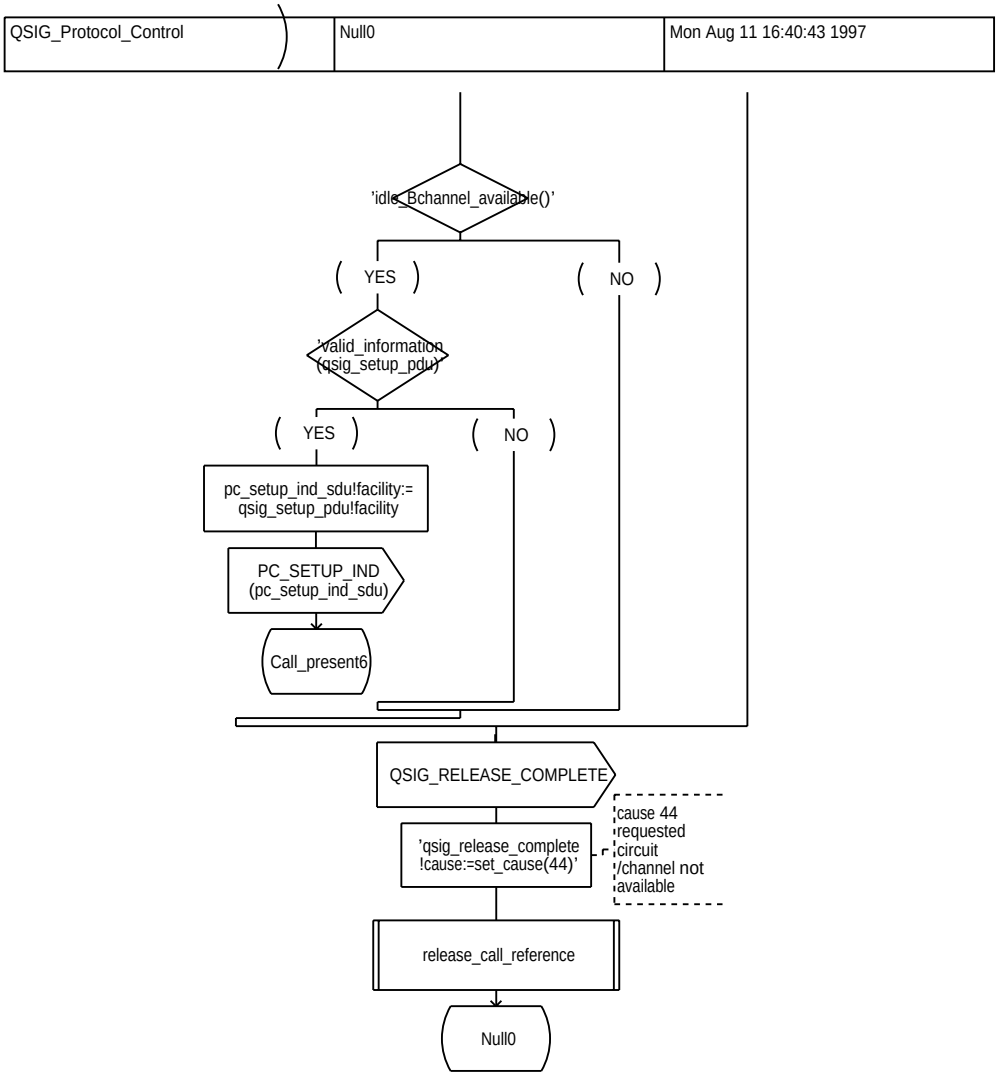


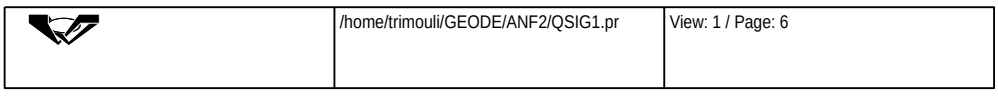
	/home/trimouli/GEODE/ANF2/QSIG1.pr	View: 1 / Page: 3

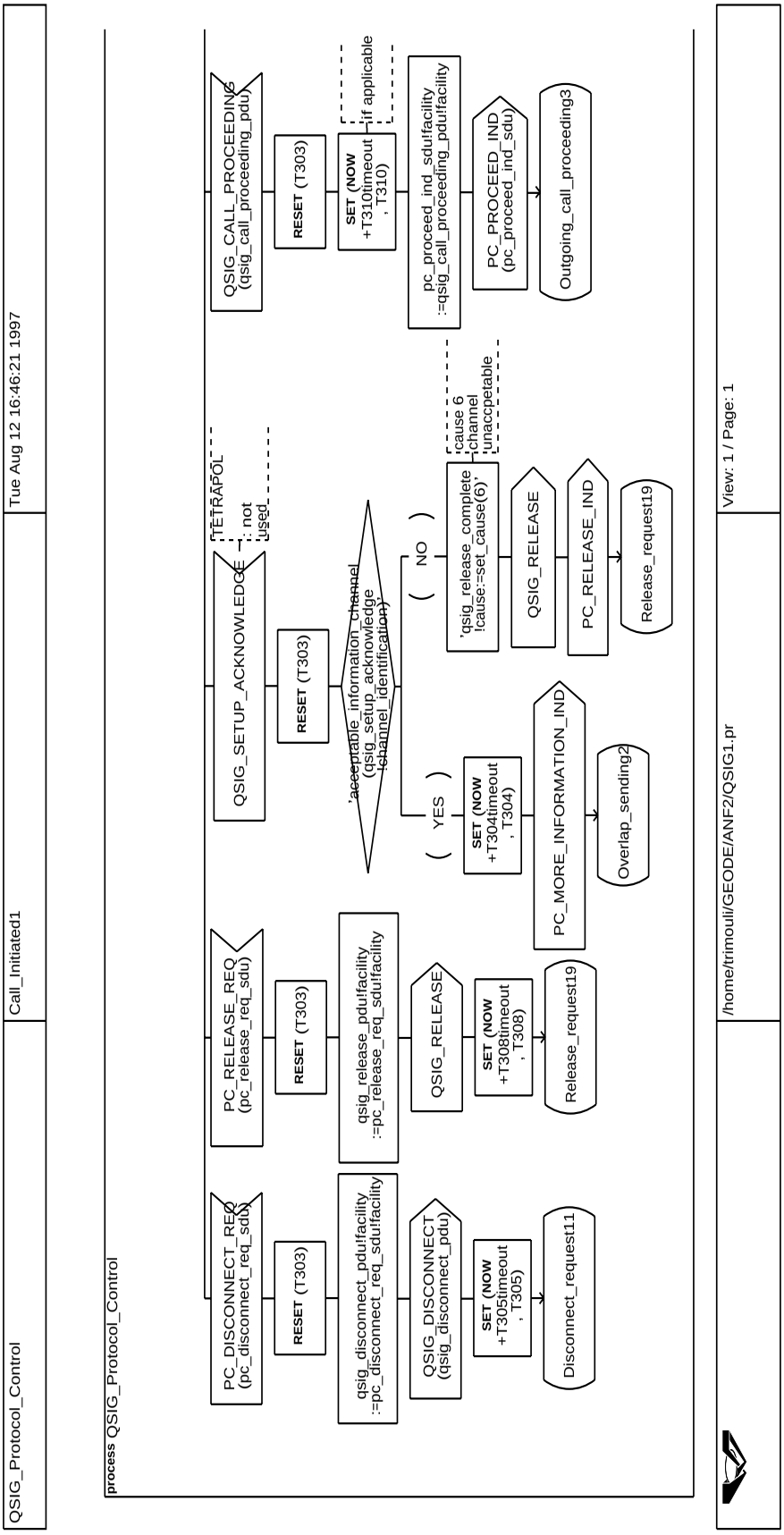
QSIG_Protocol_Control	Null0	Mon Aug 11 16:40:43 1997

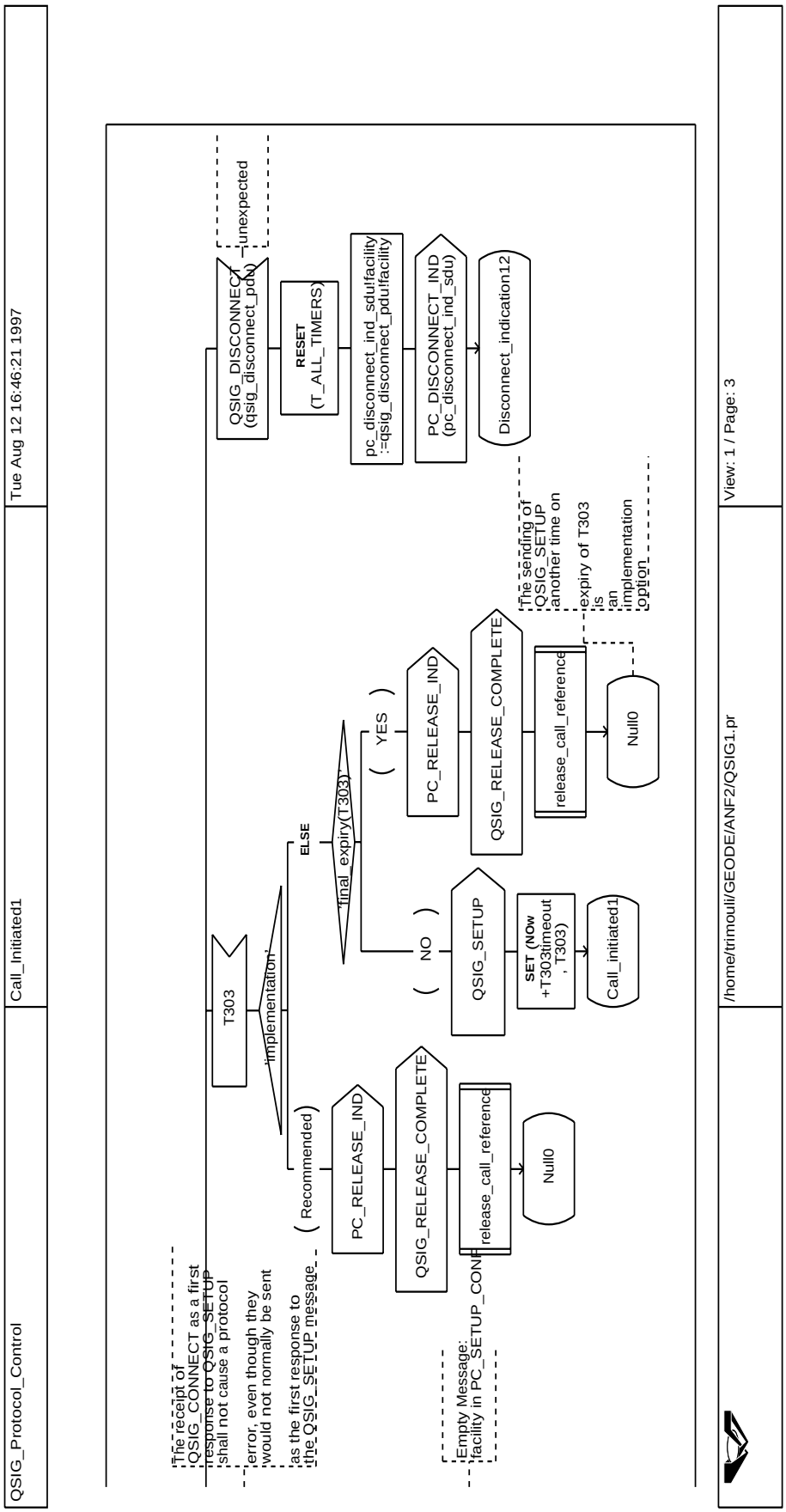


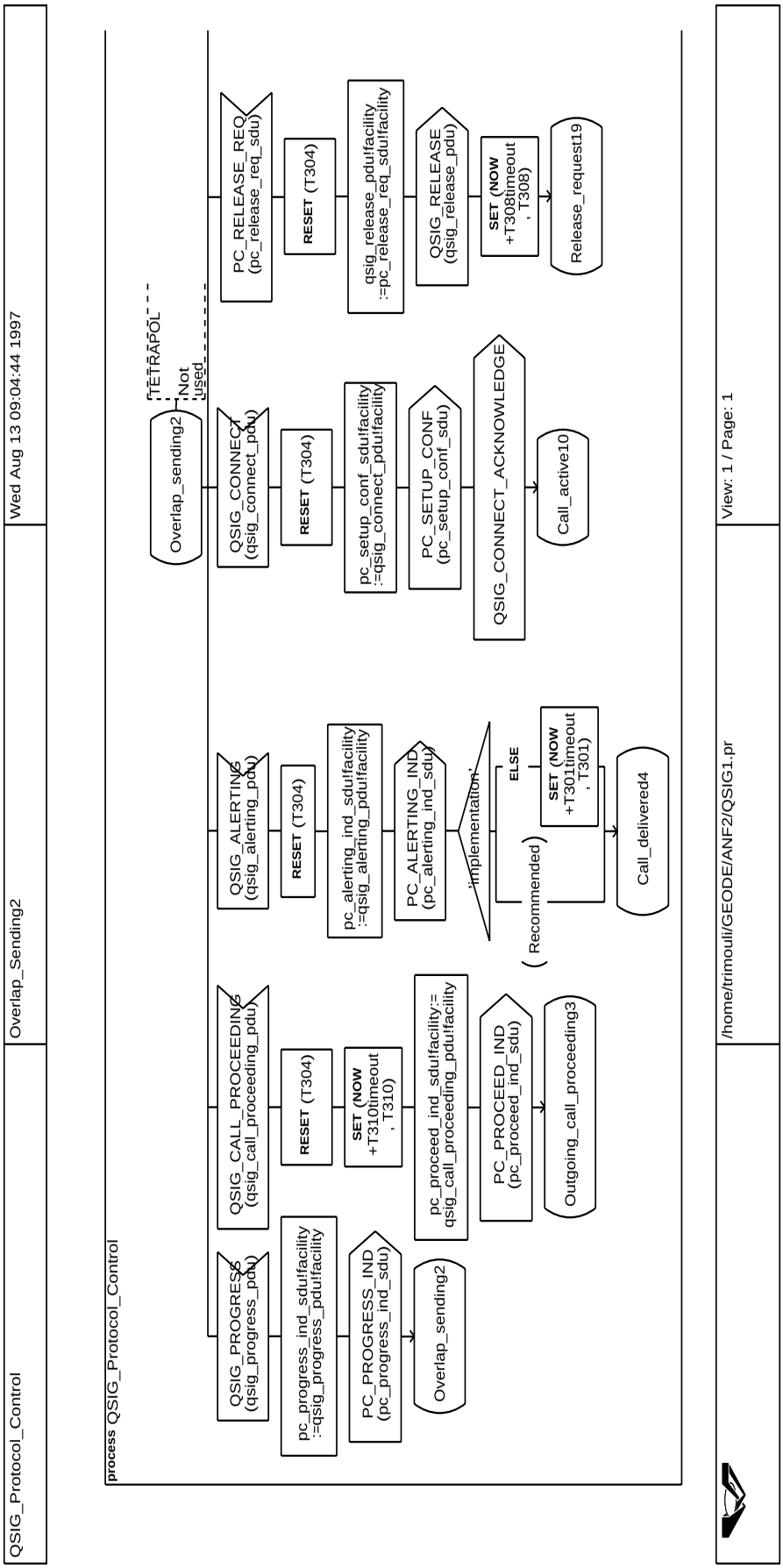
	/home/trimouli/GEODE/ANF2/QSIG1.pr	View: 1 / Page: 4

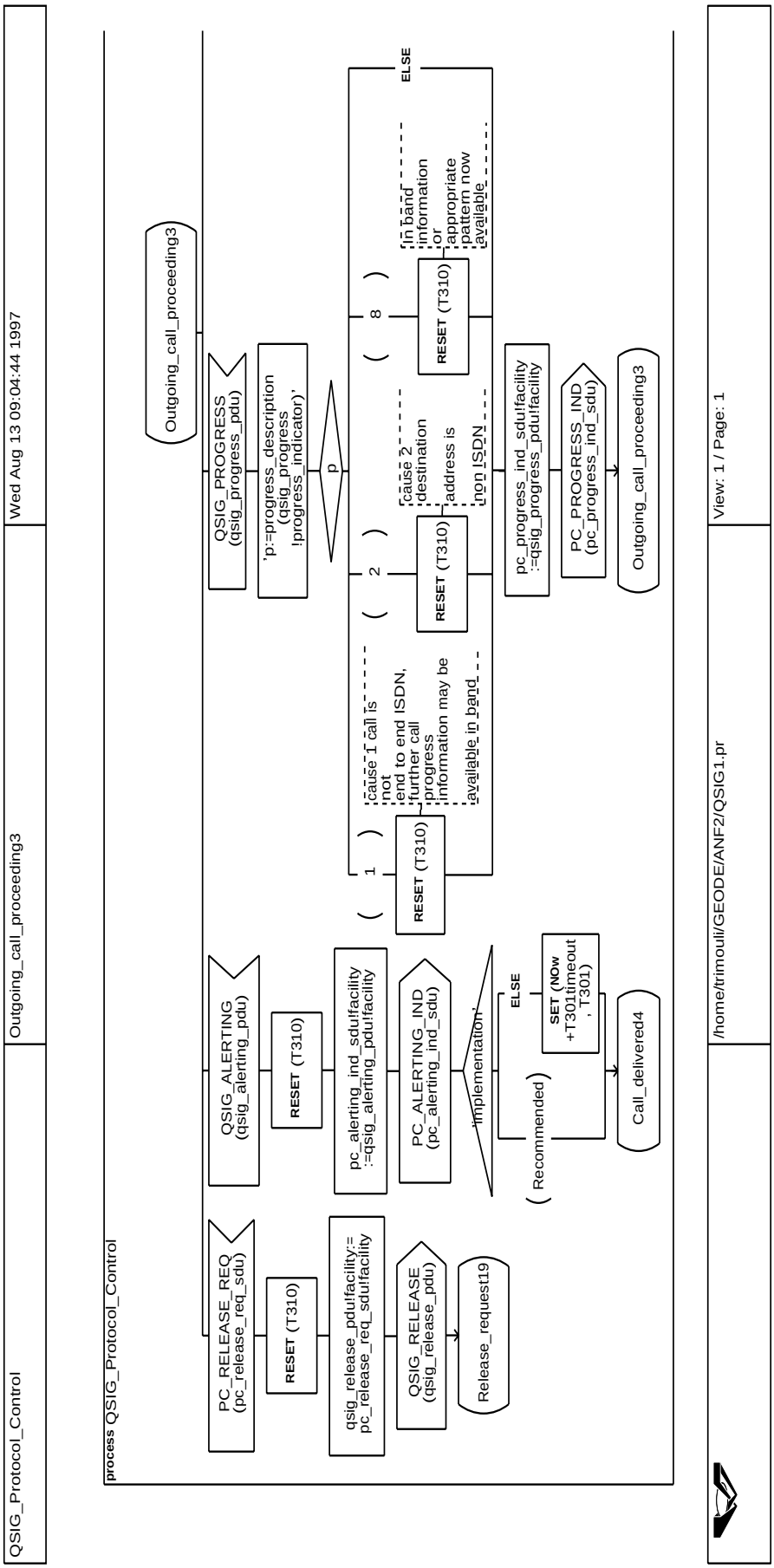


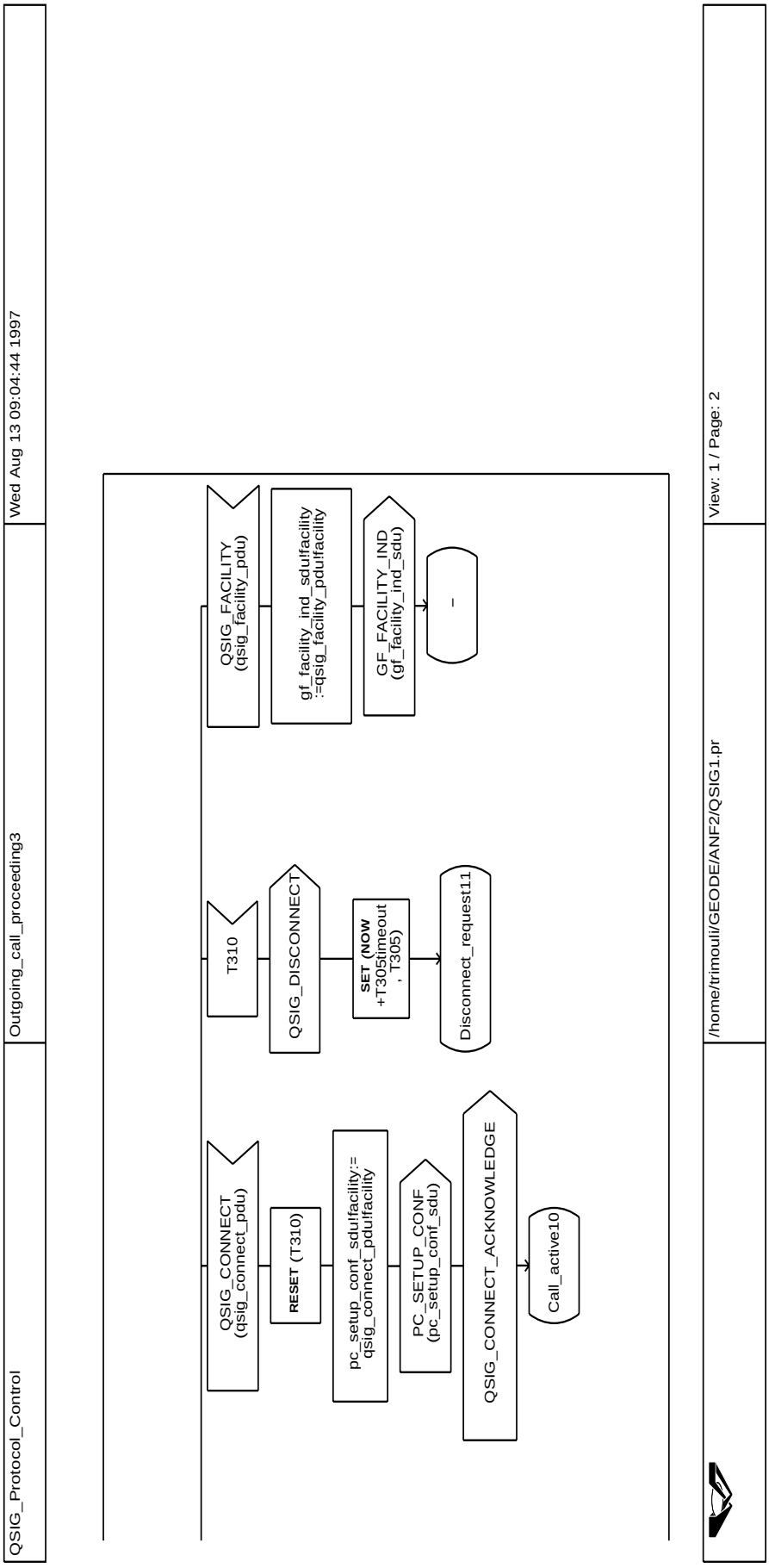


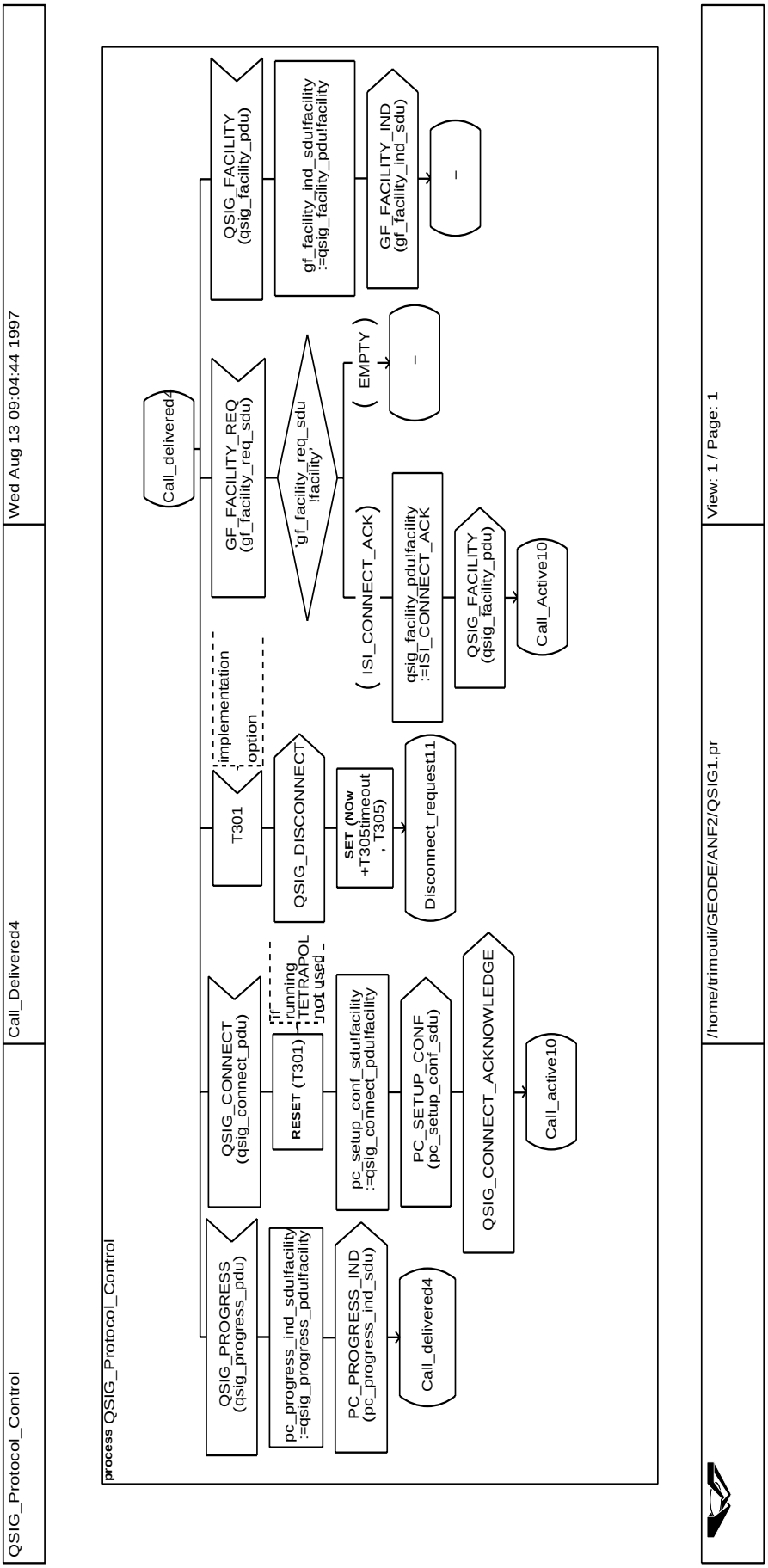


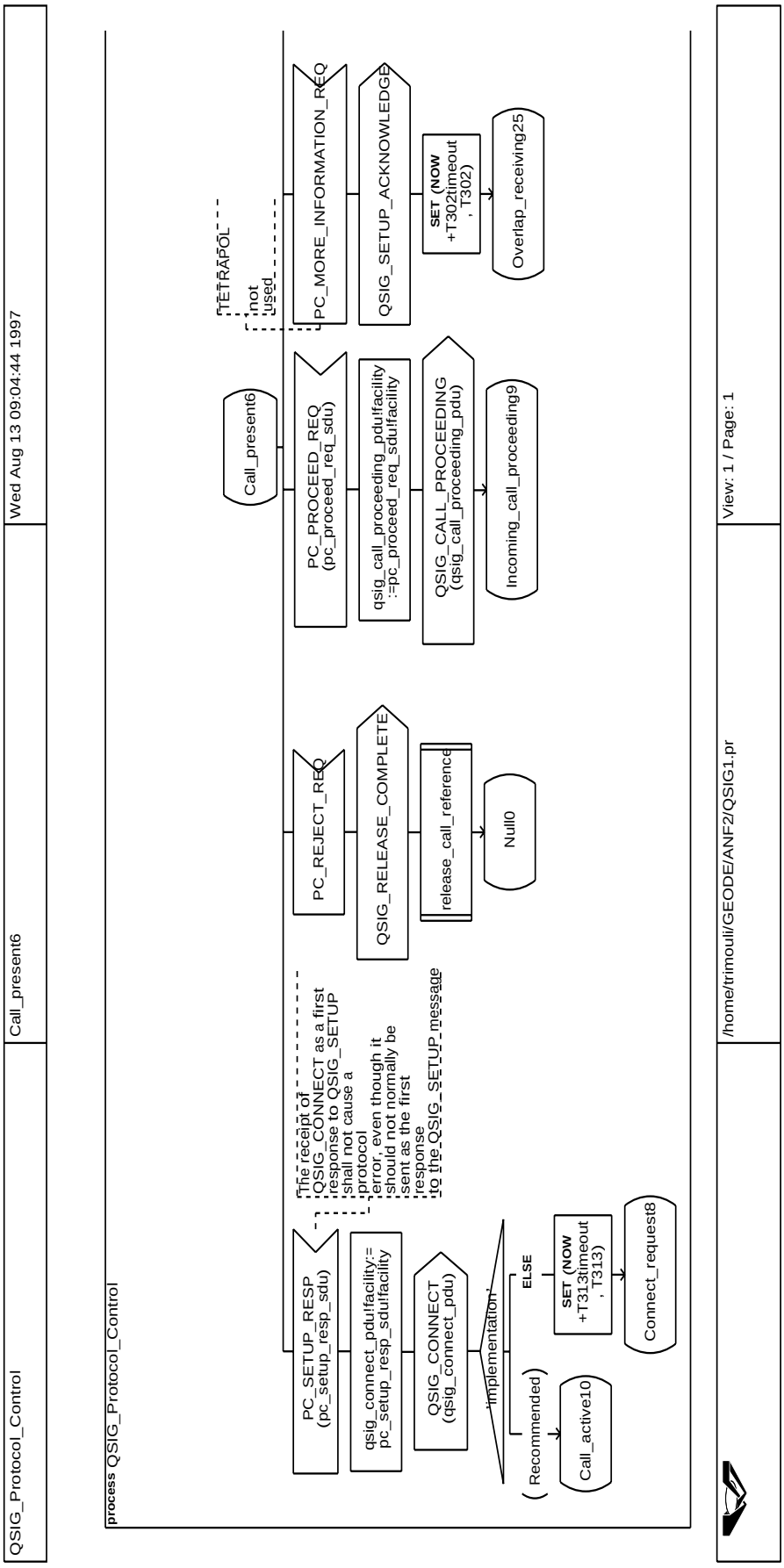


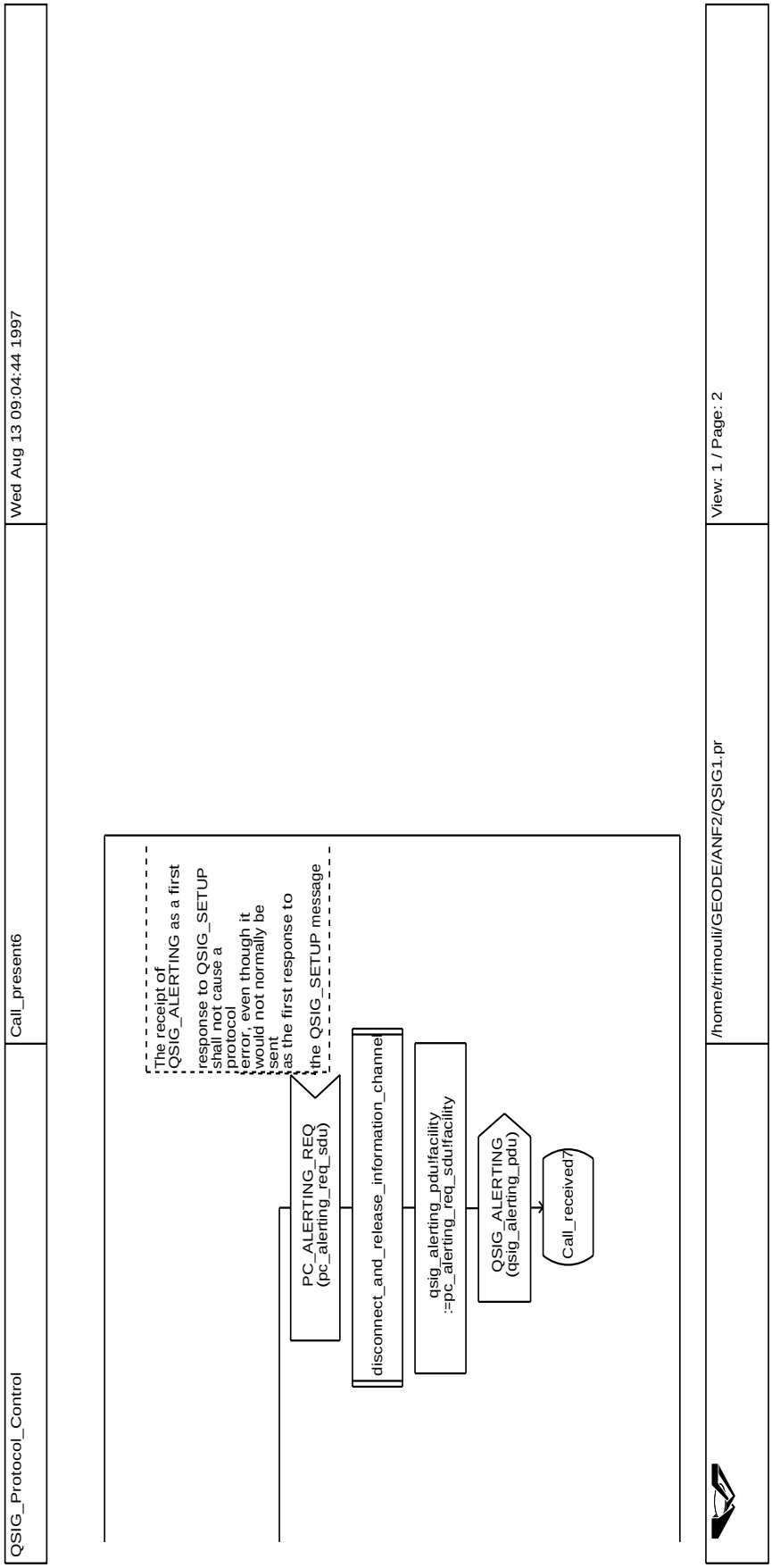


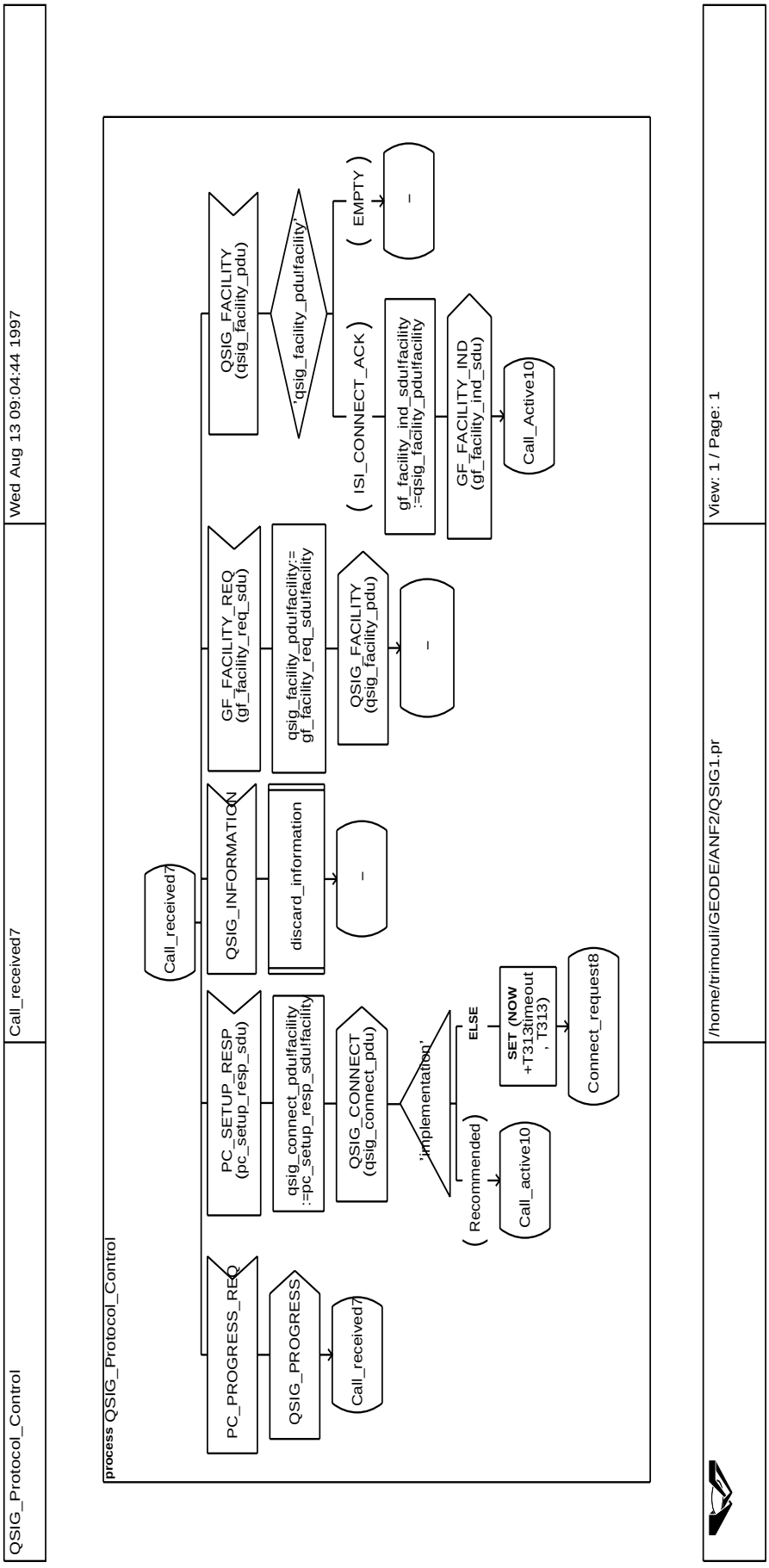


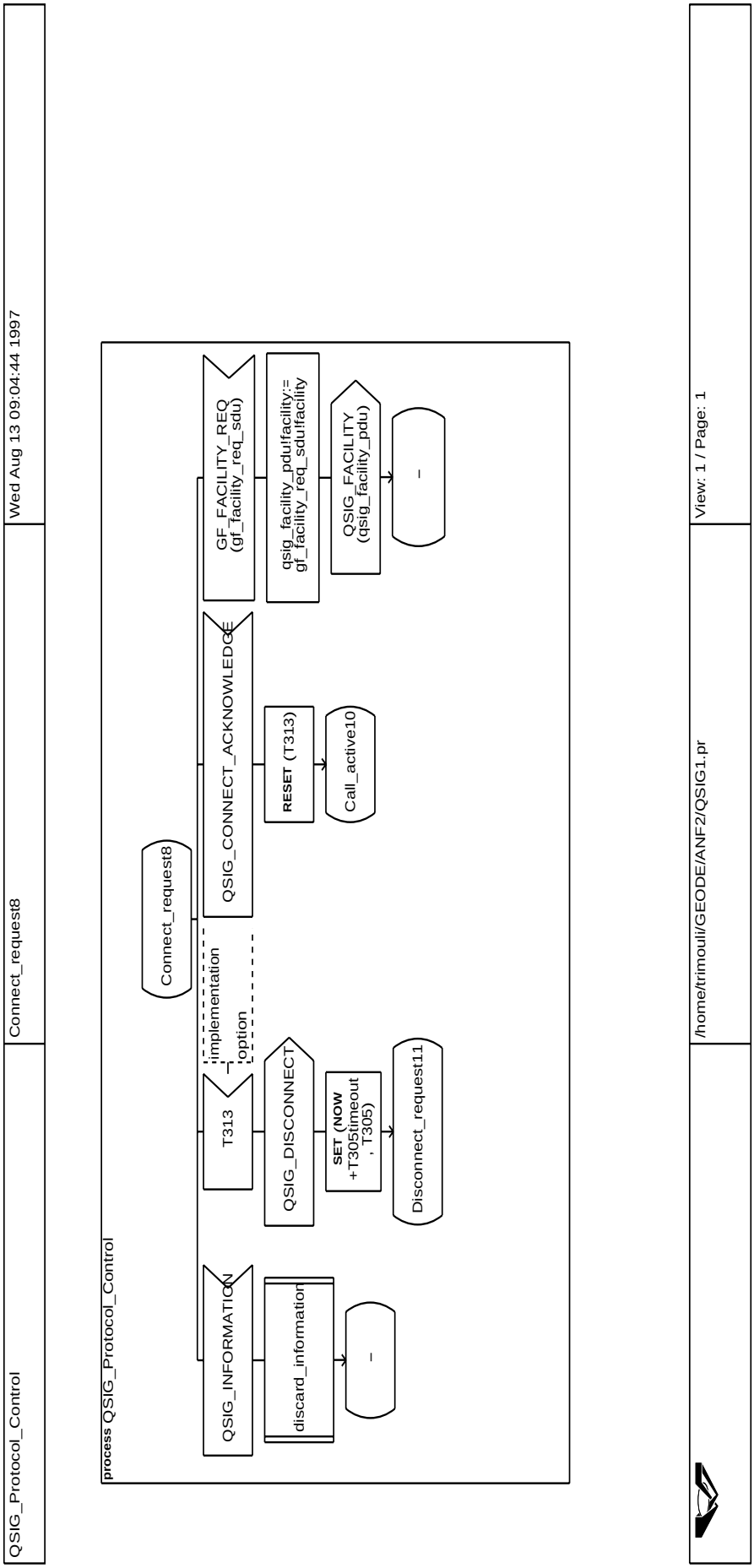


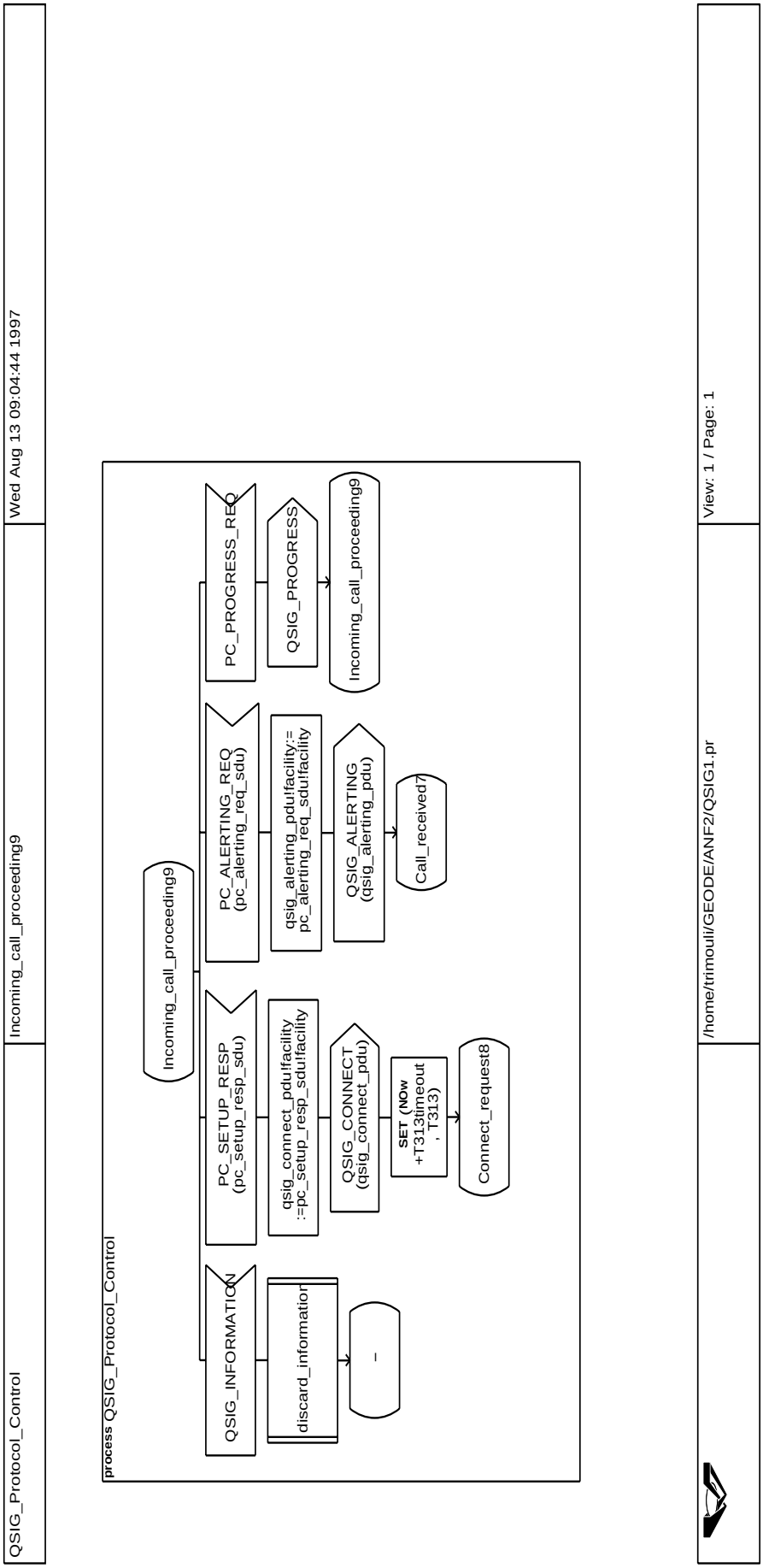


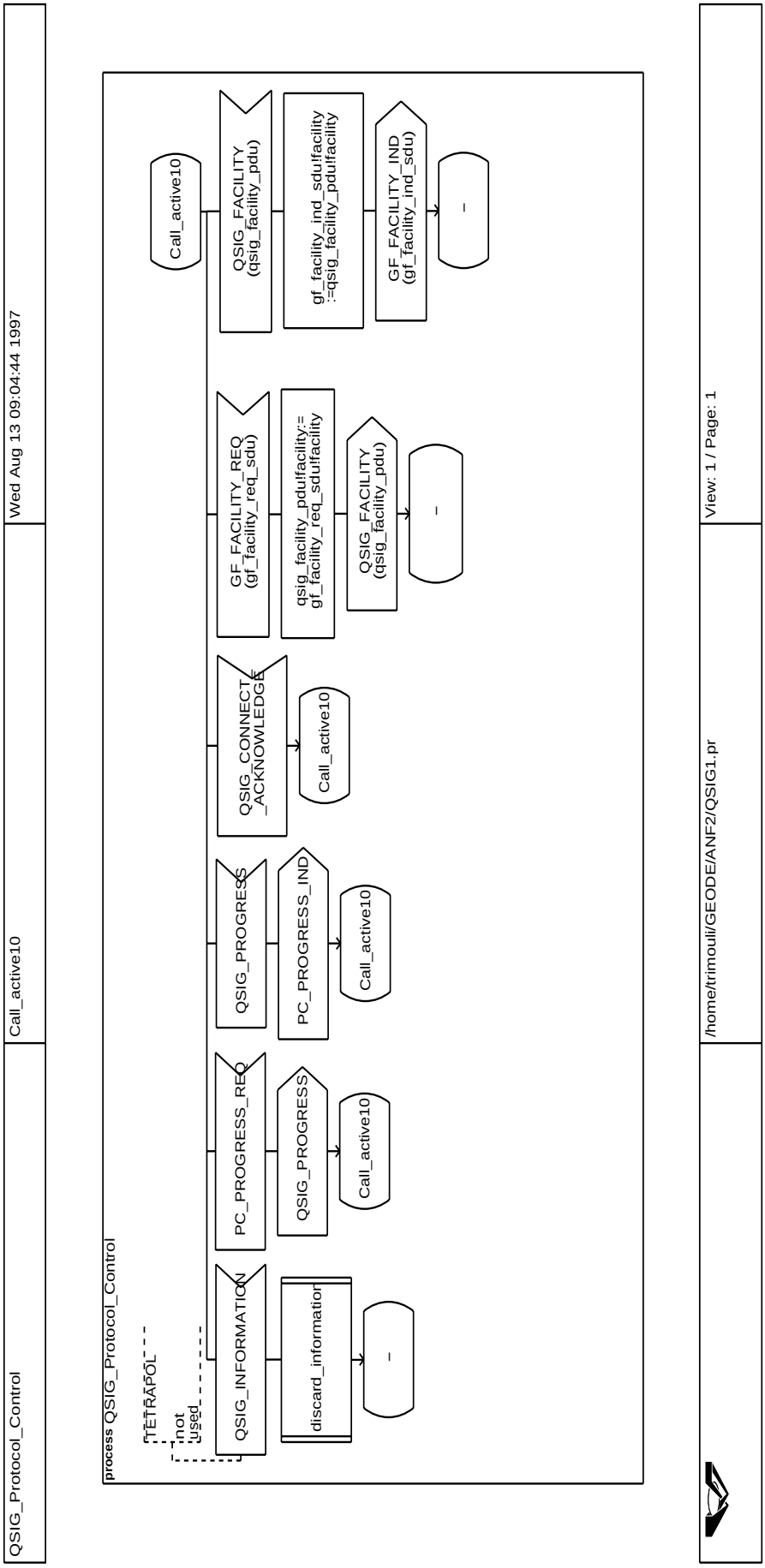




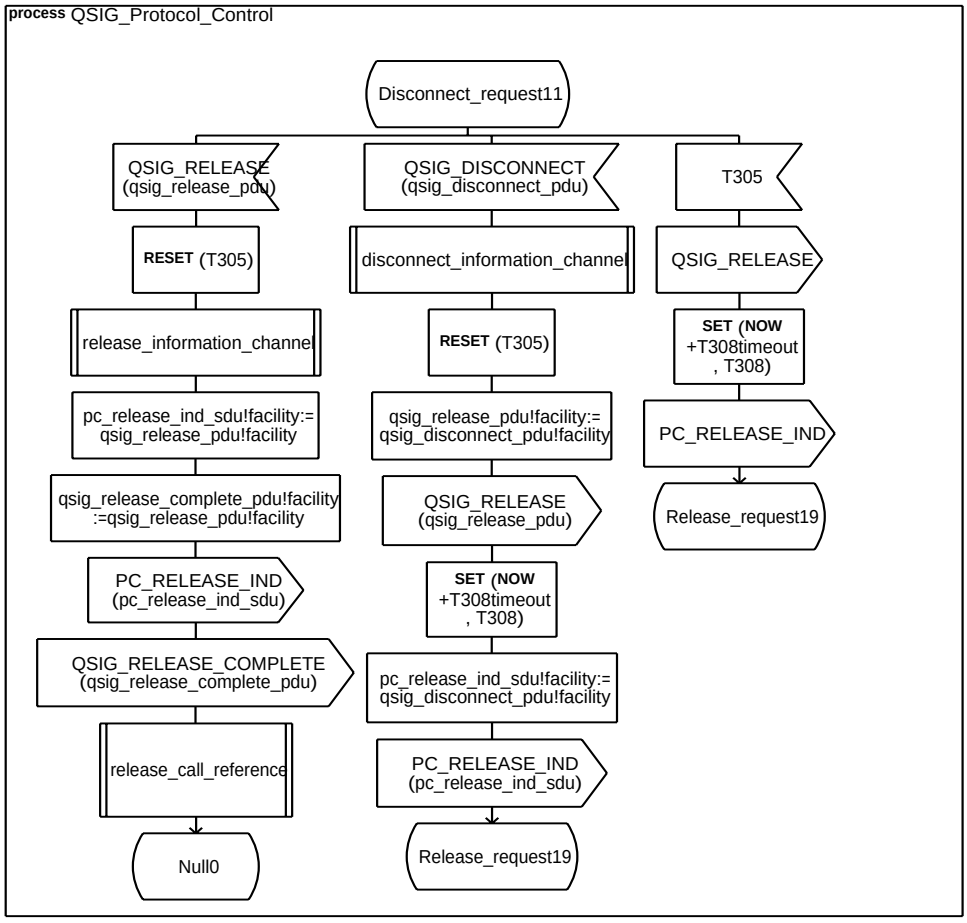




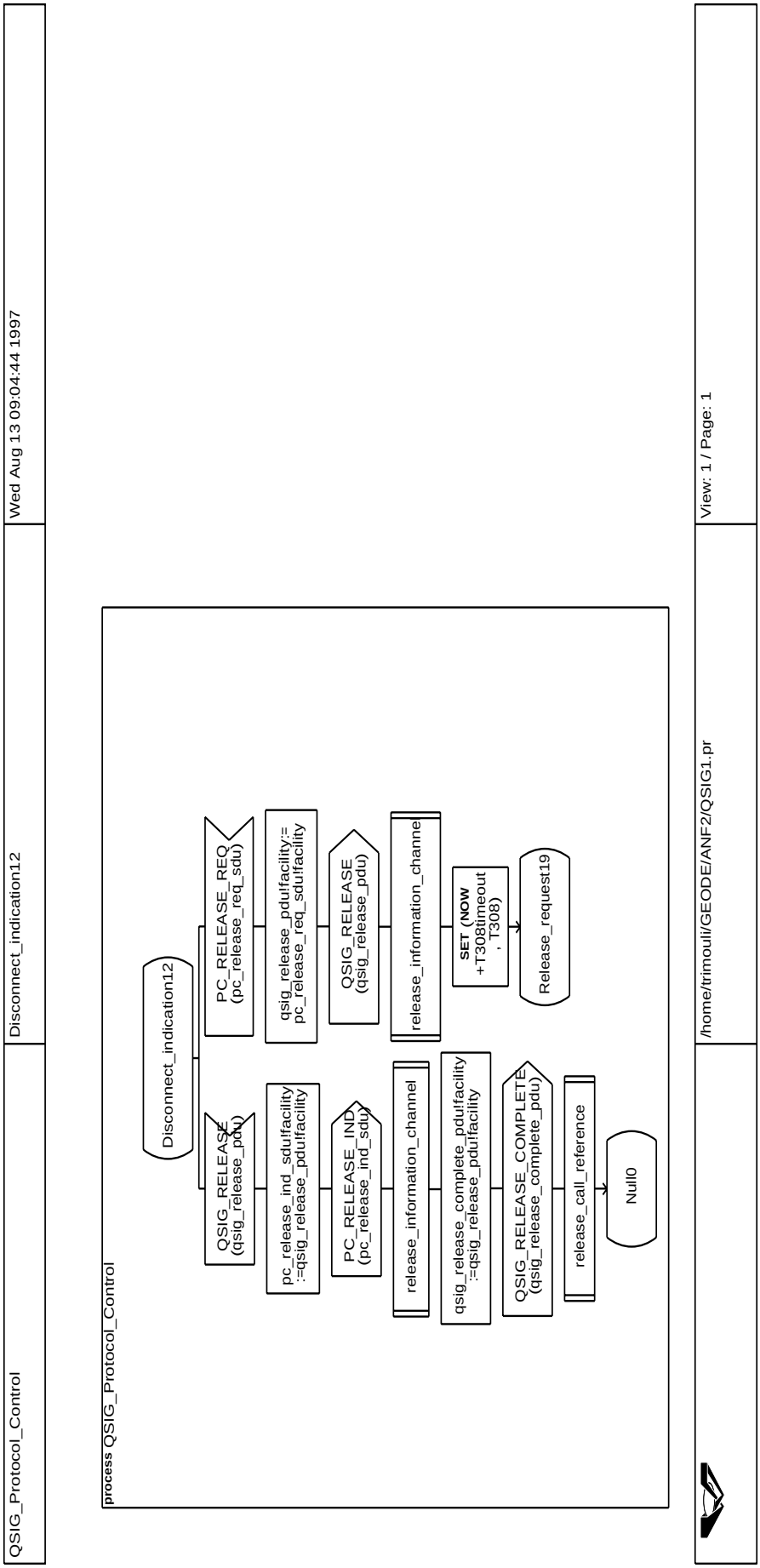


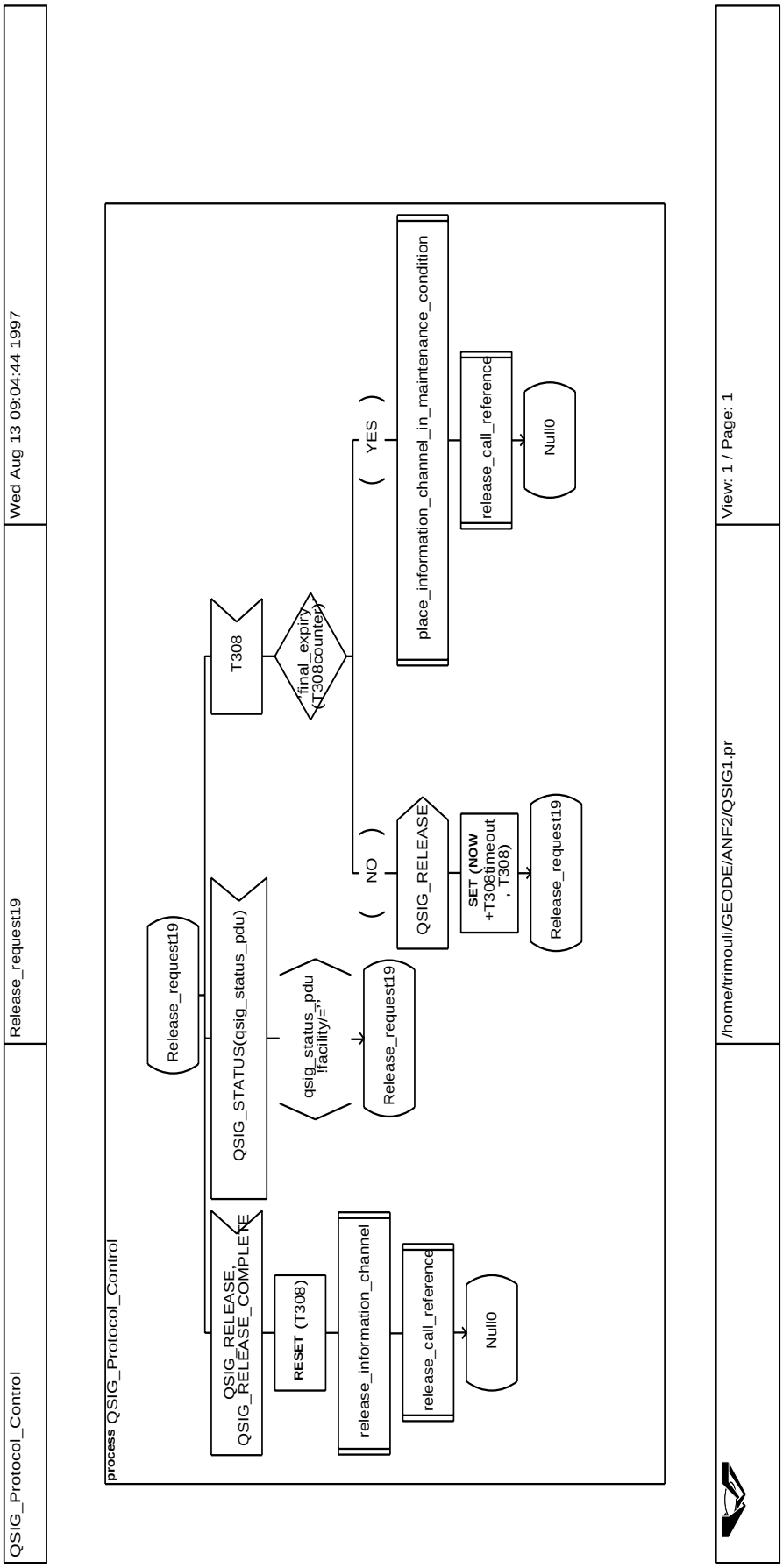


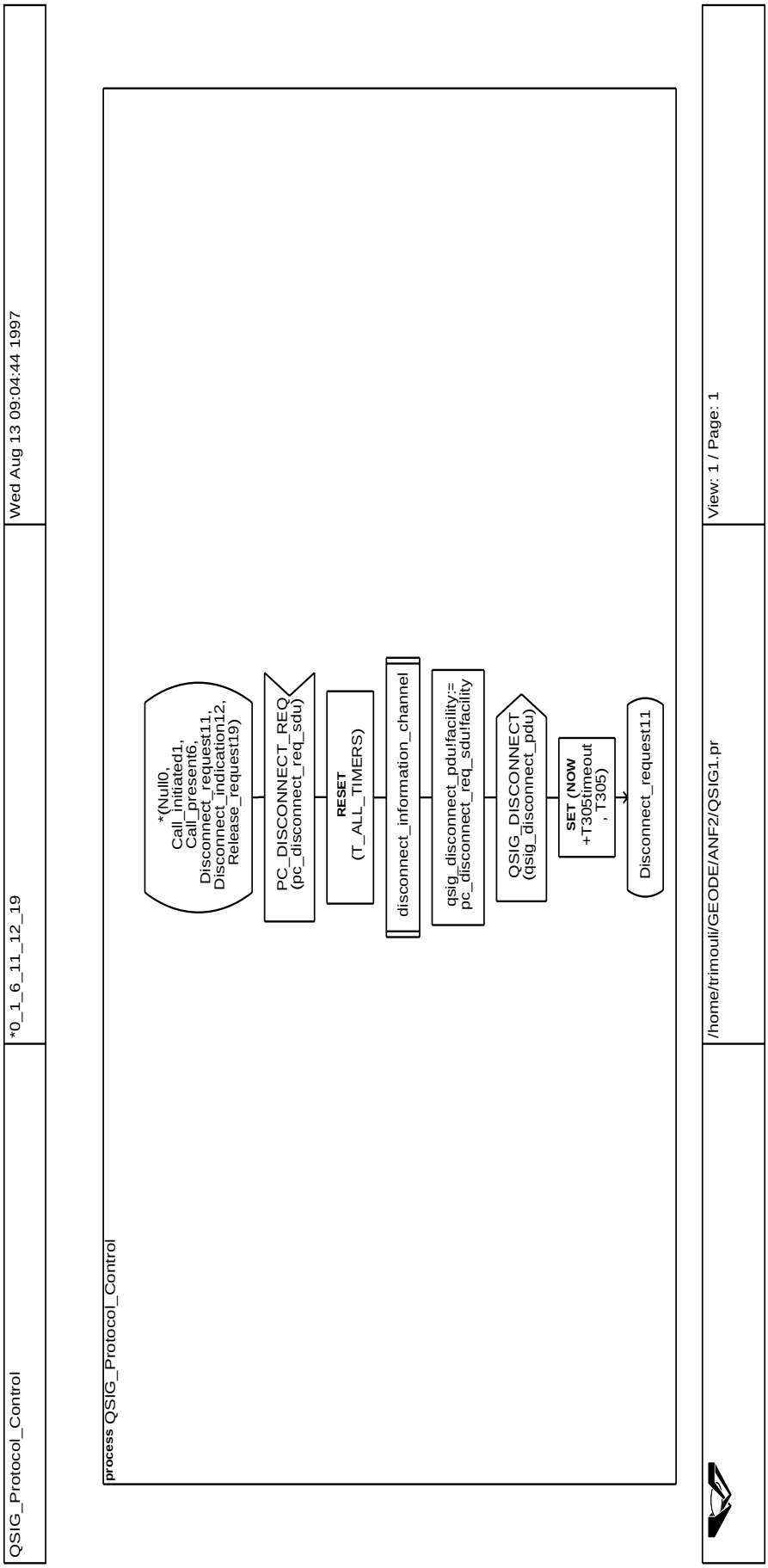
QSIG_Protocol_Control	Disconnect_request11	Wed Aug 13 09:04:44 1997



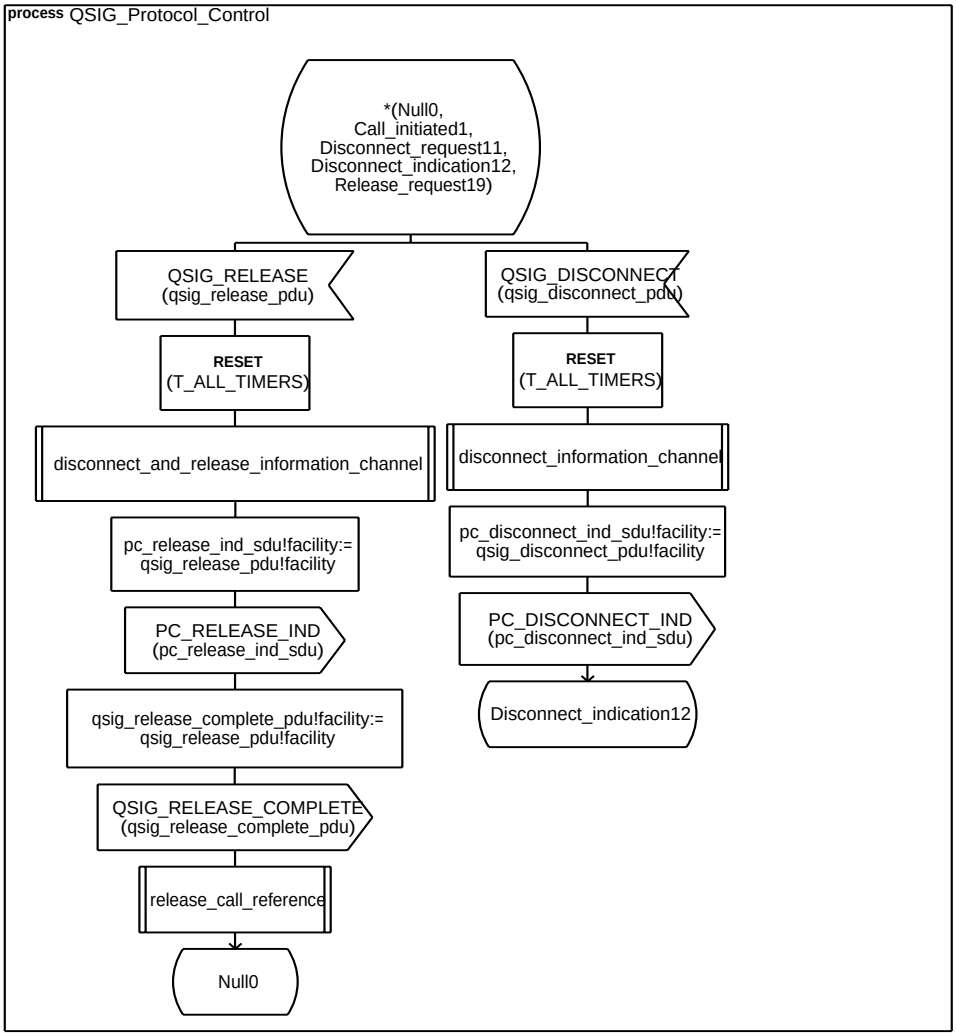
	/home/trimouli/GEODE/ANF2/QSIG1.pr	View: 1 / Page: 1



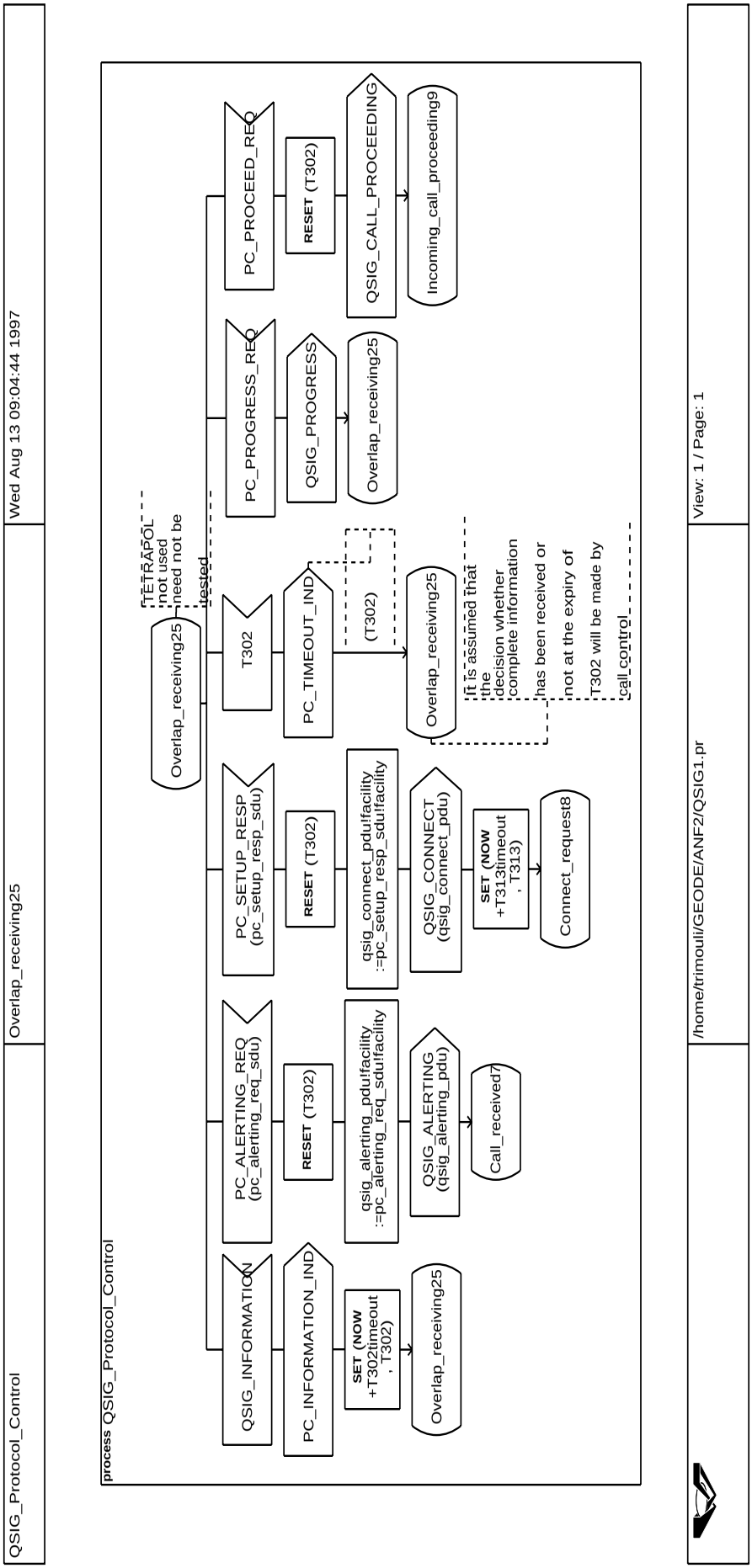


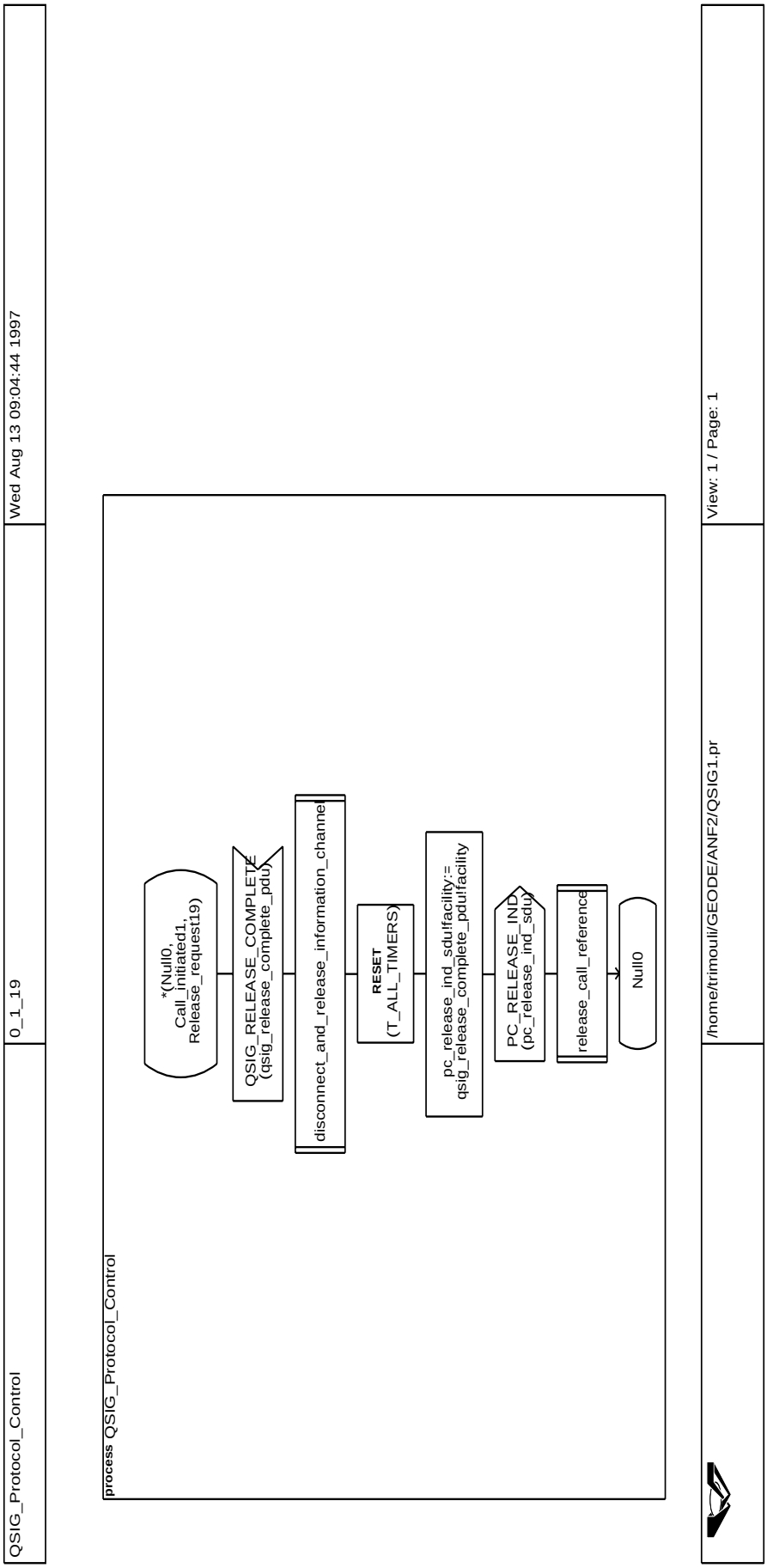


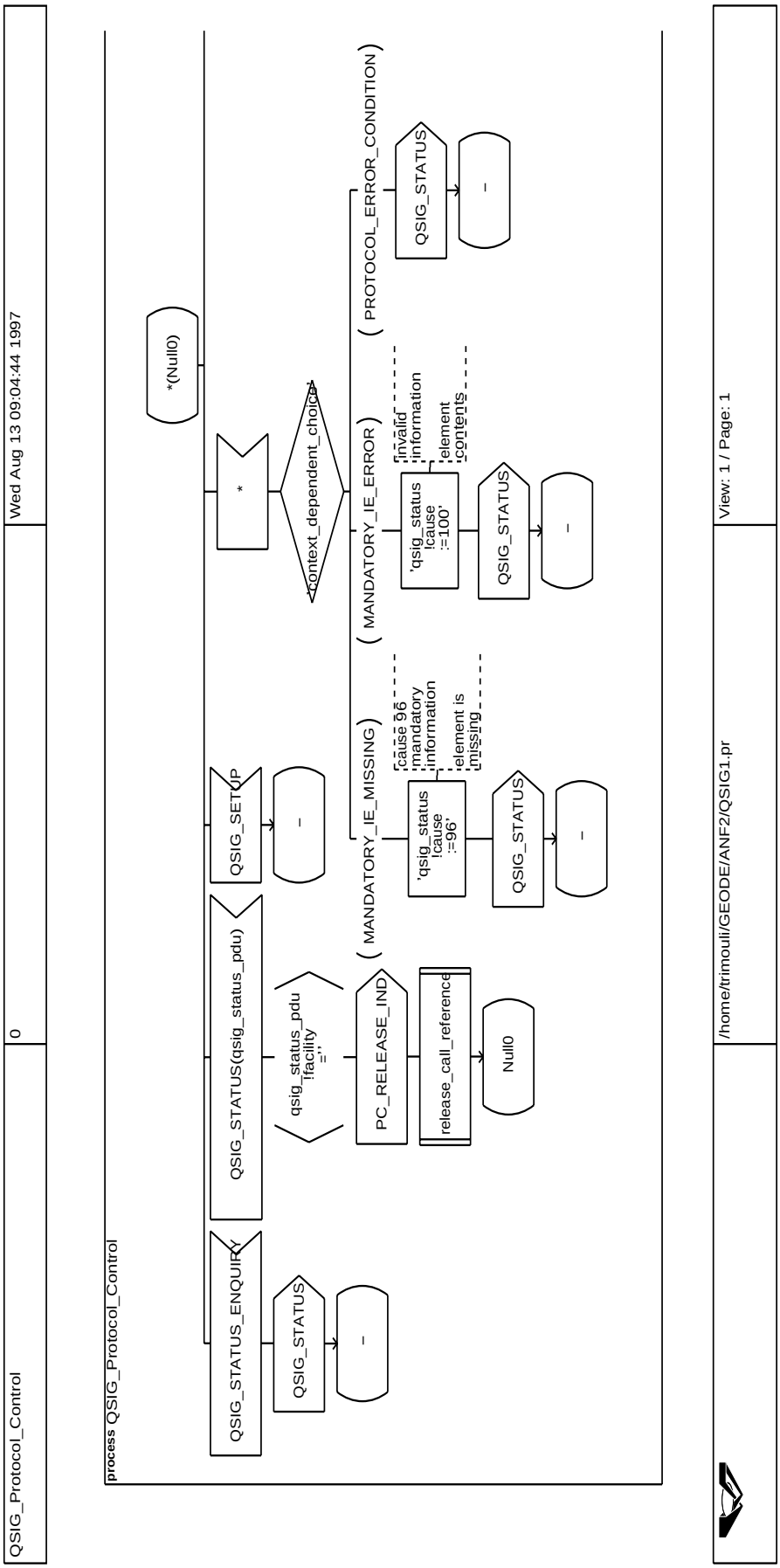
QSIG_Protocol_Control	0_1_11_12_19	Wed Aug 13 09:04:44 1997

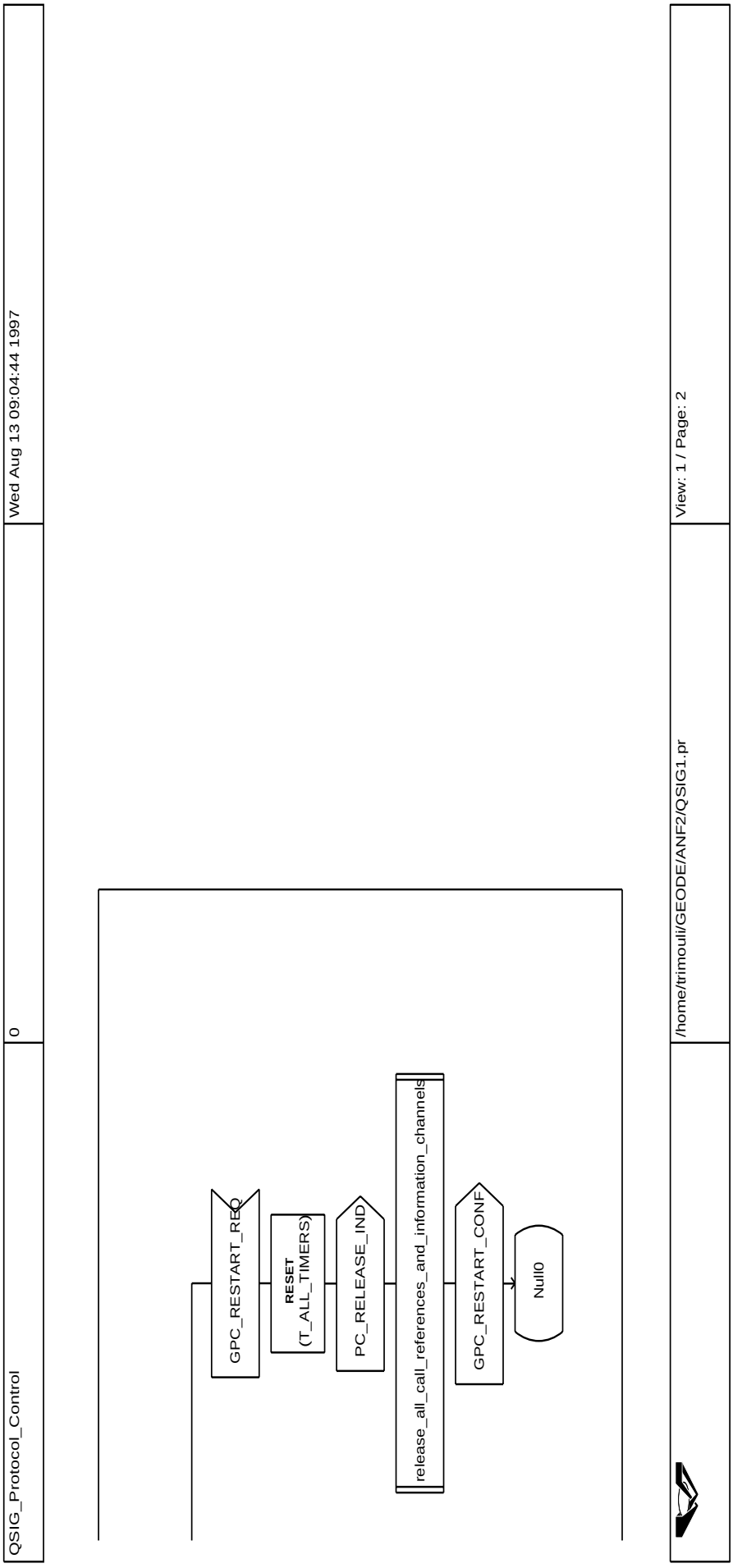


	/home/trimouli/GEODE/ANF2/QSIG1.pr	View: 1 / Page: 1

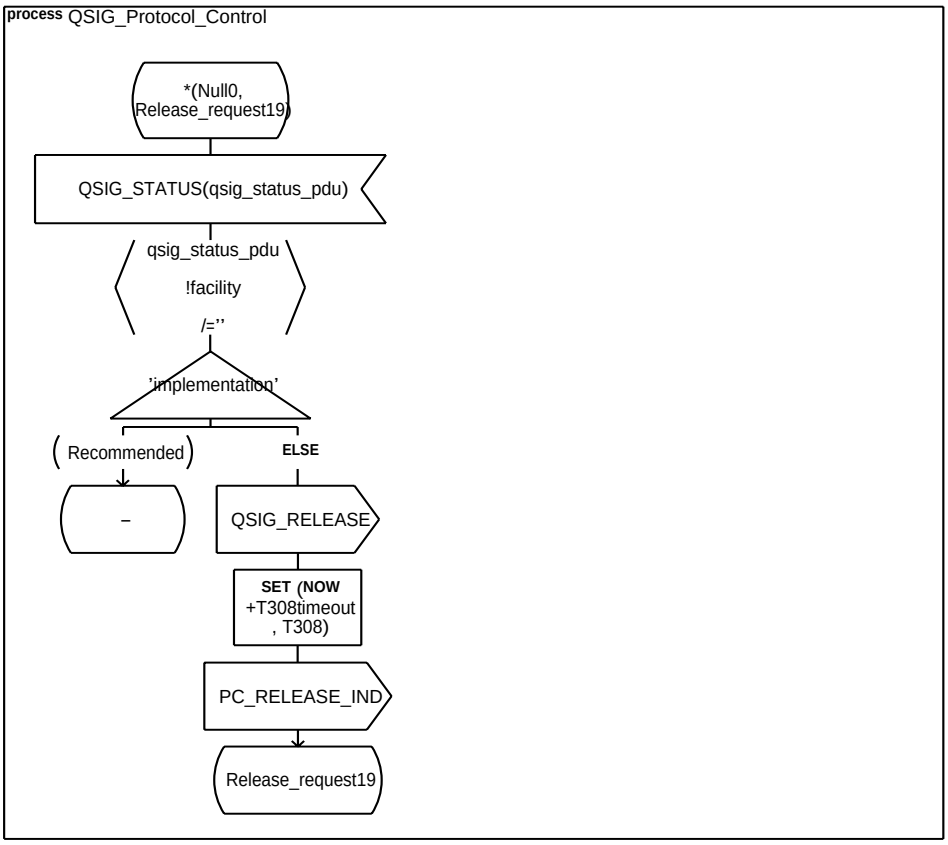




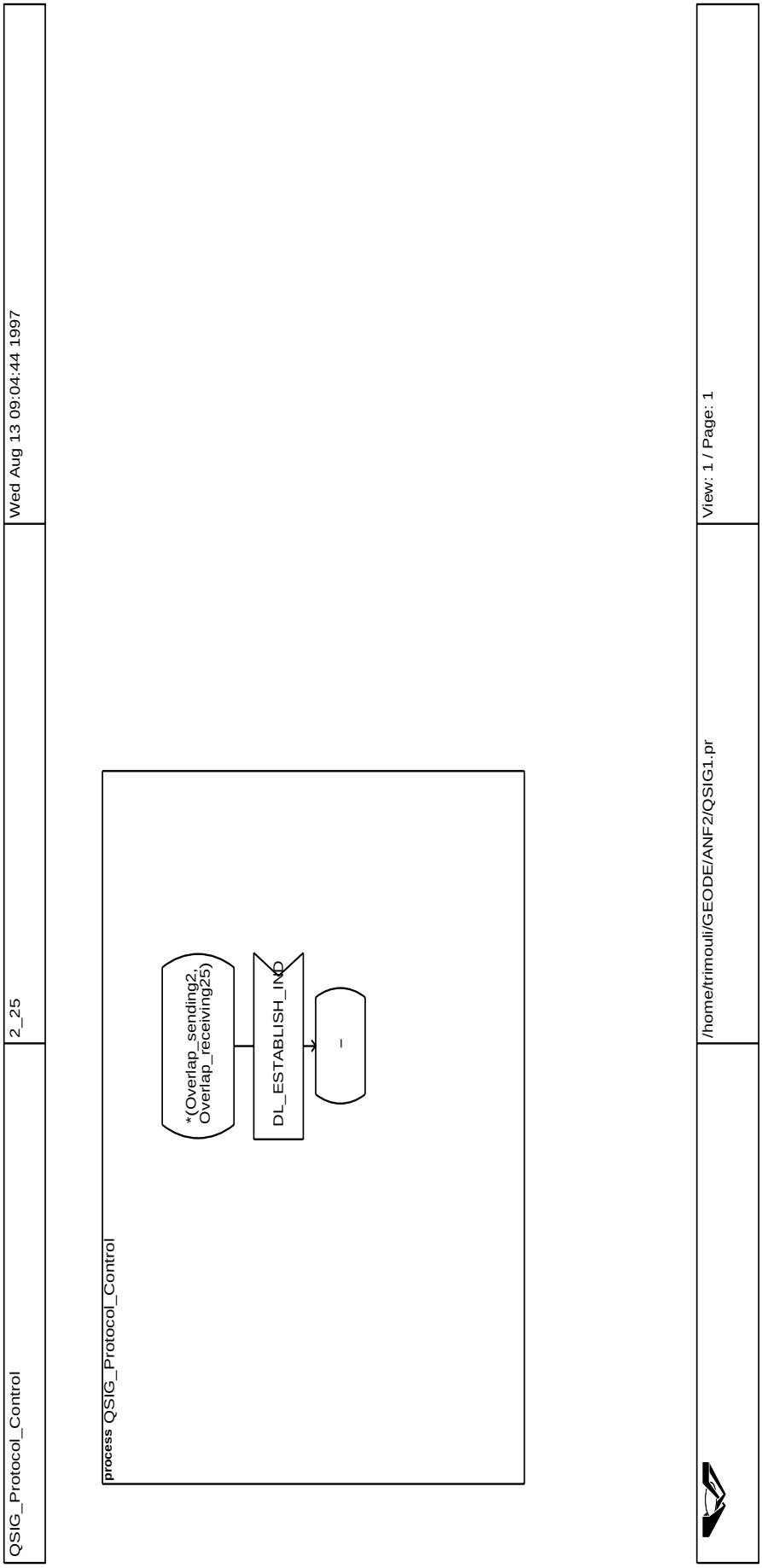


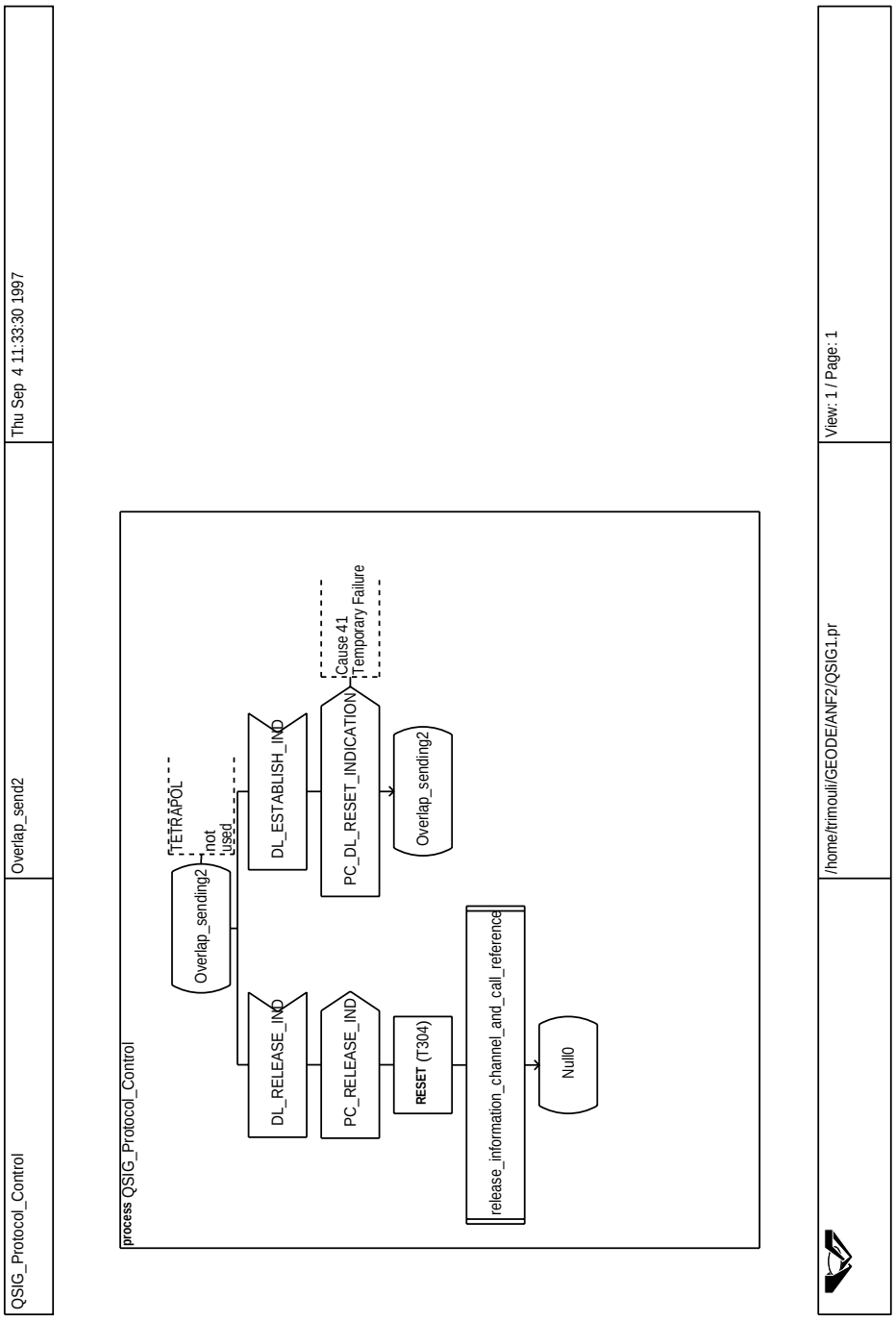


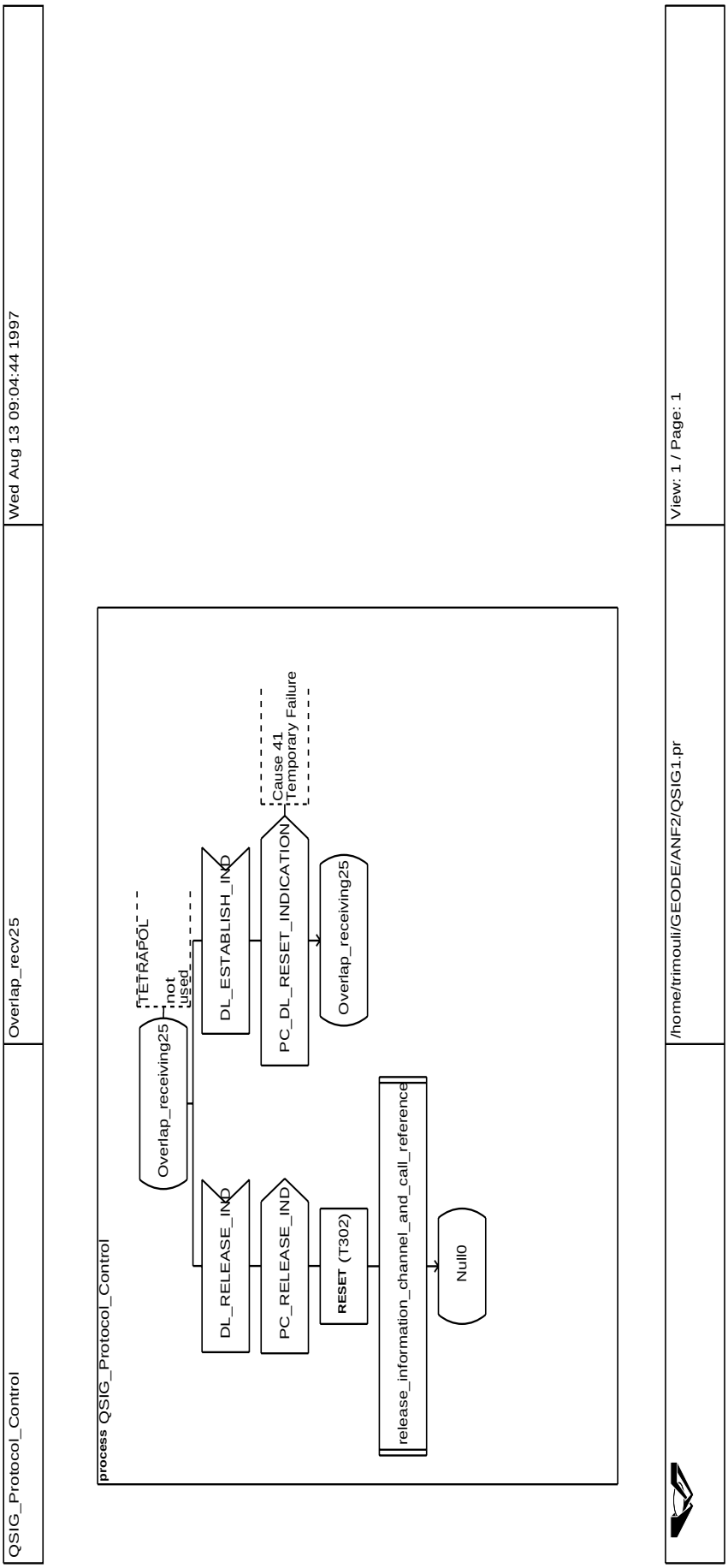
QSIG_Protocol_Control	0_19	Wed Aug 13 09:04:44 1997

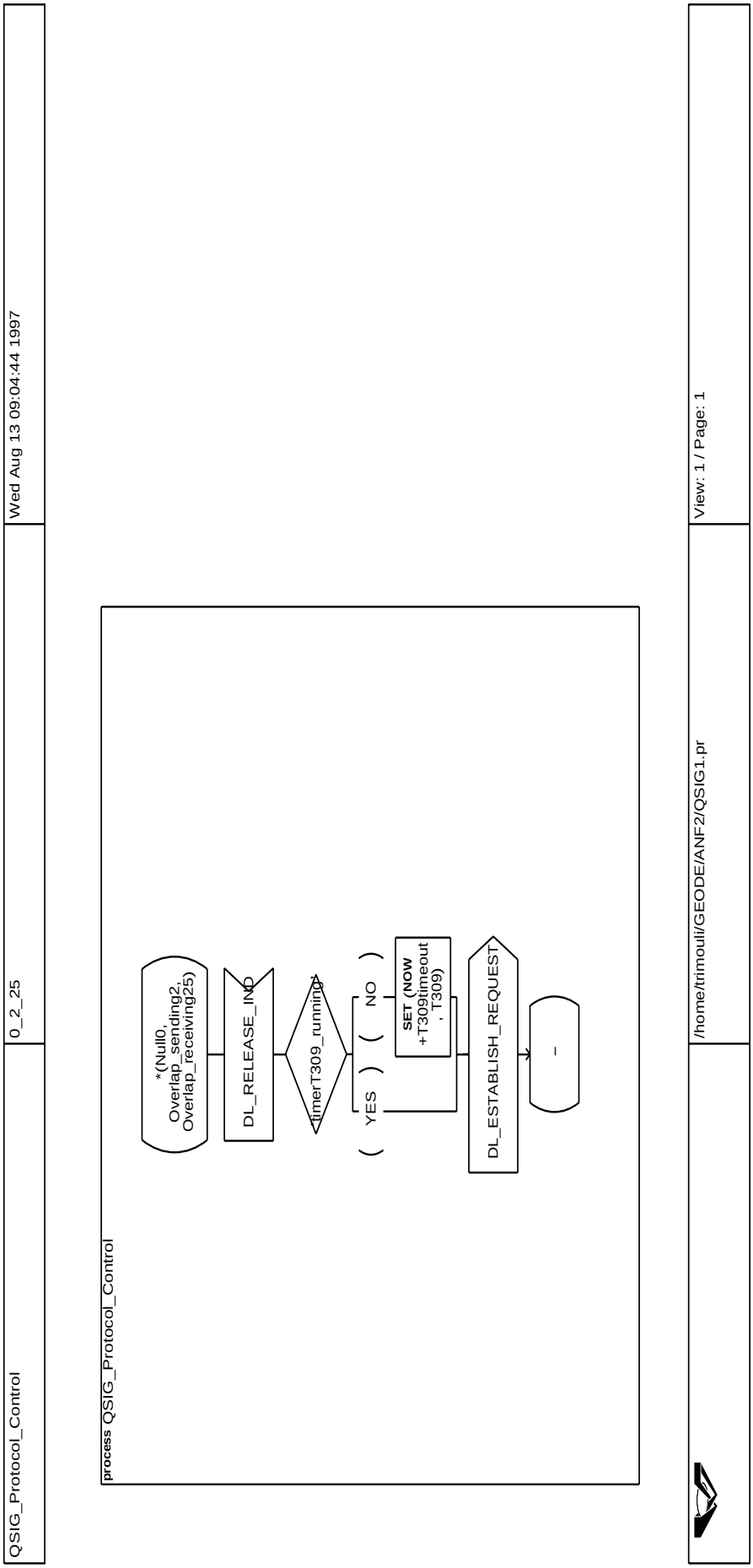


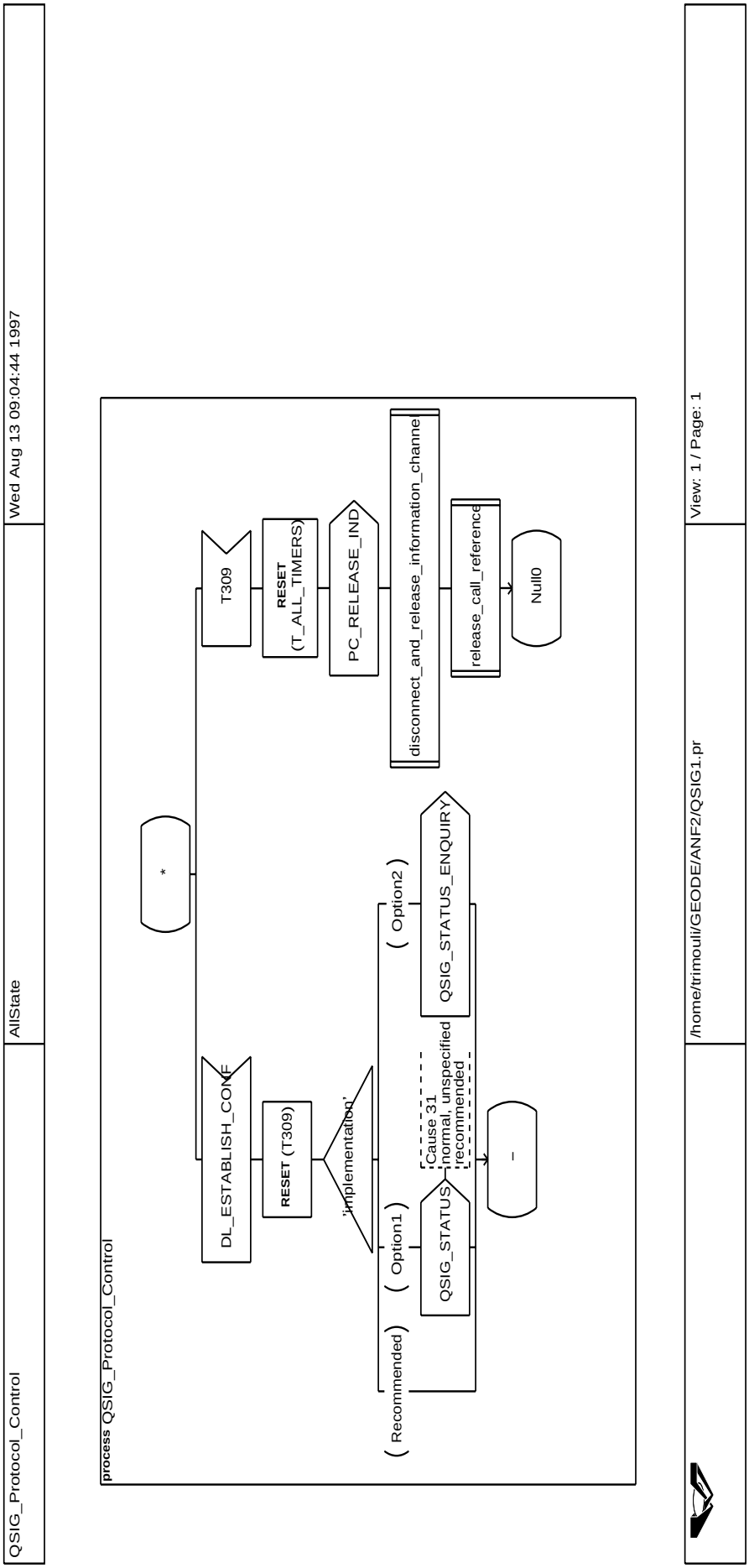
	/home/trimouli/GEODE/ANF2/QSIG1.pr	View: 1 / Page: 1











4.3. ANF model

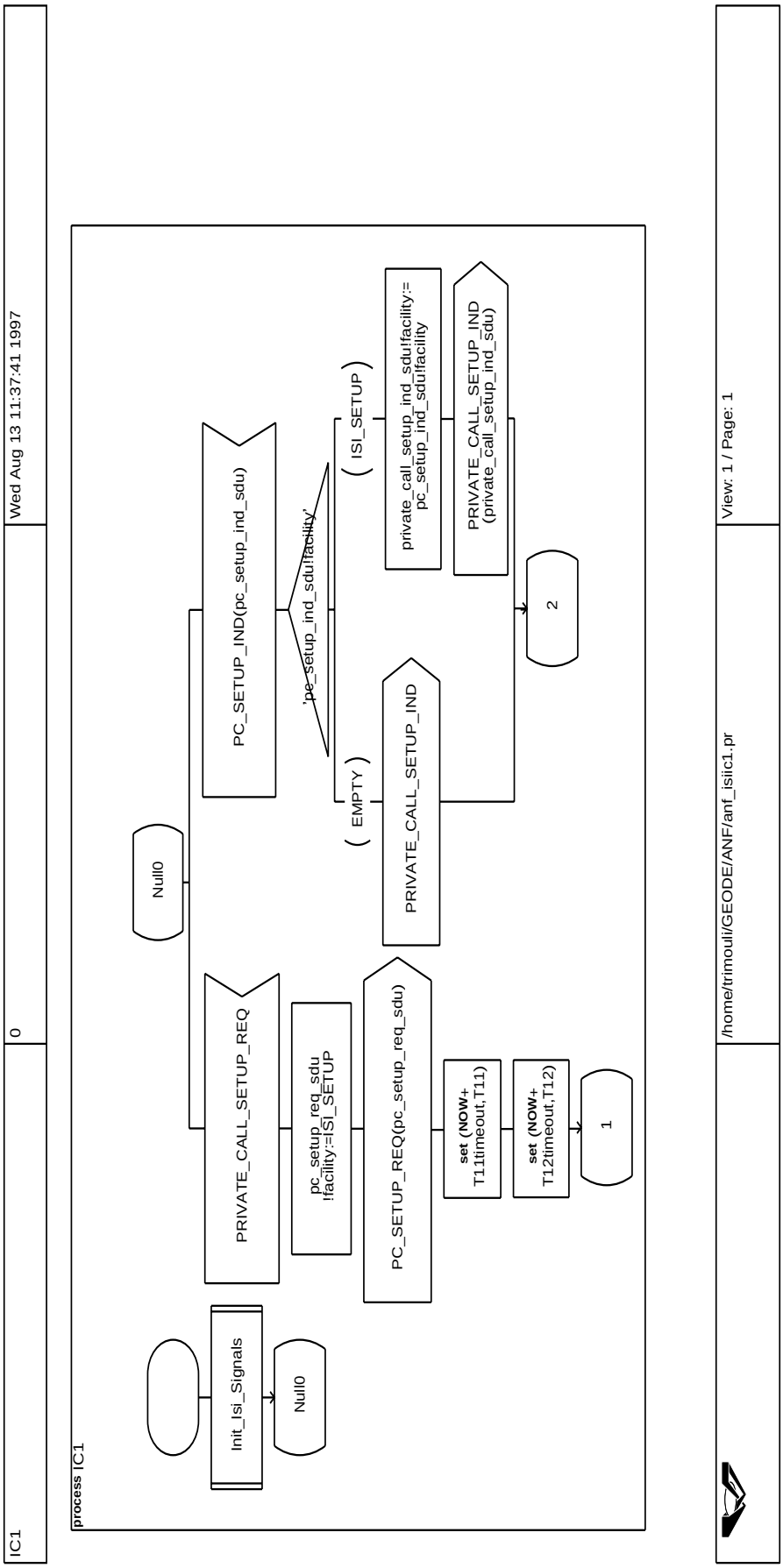
IC1	PR Declaration	Wed Aug 13 11:37:41 1997

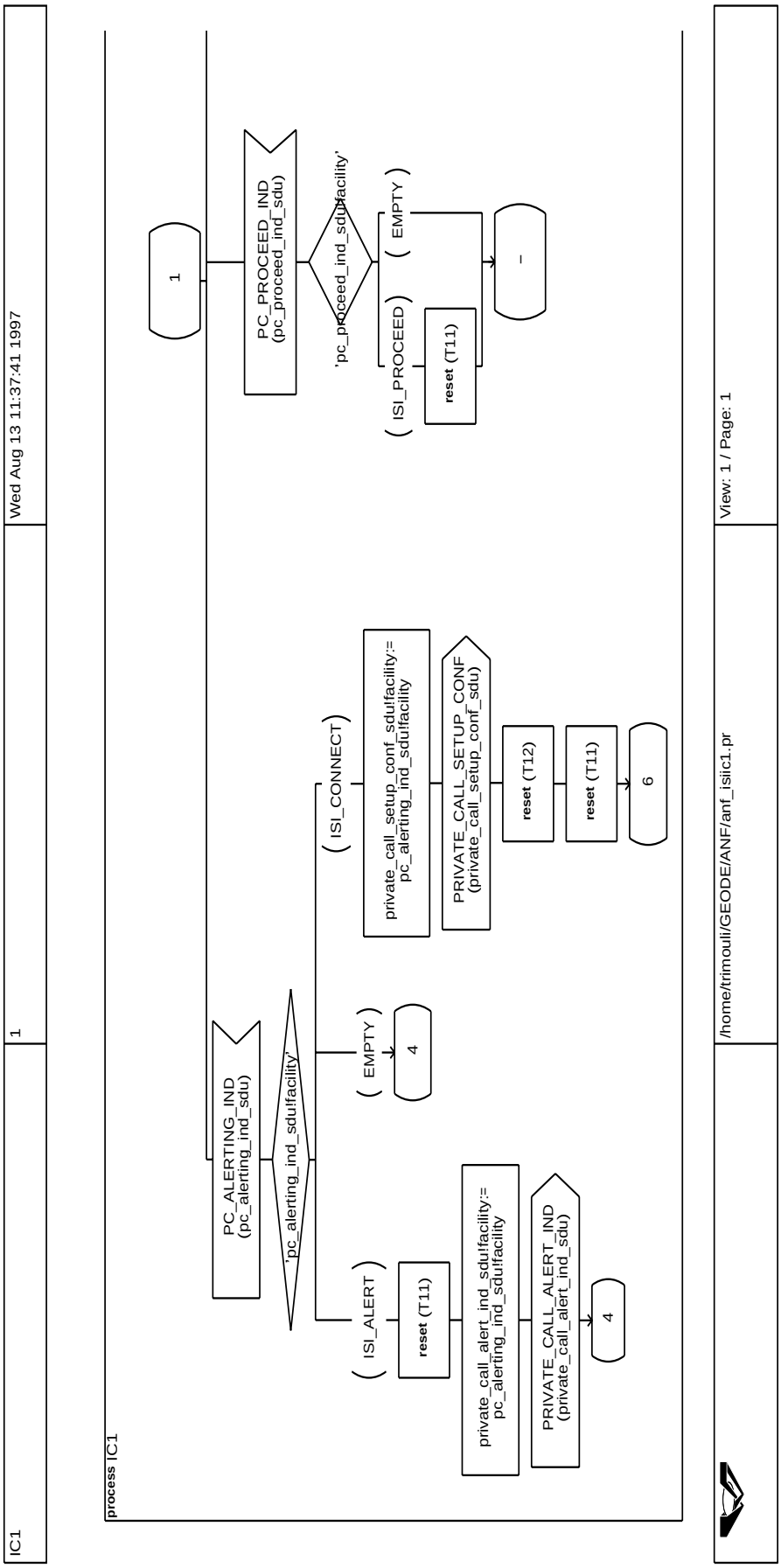
process IC1

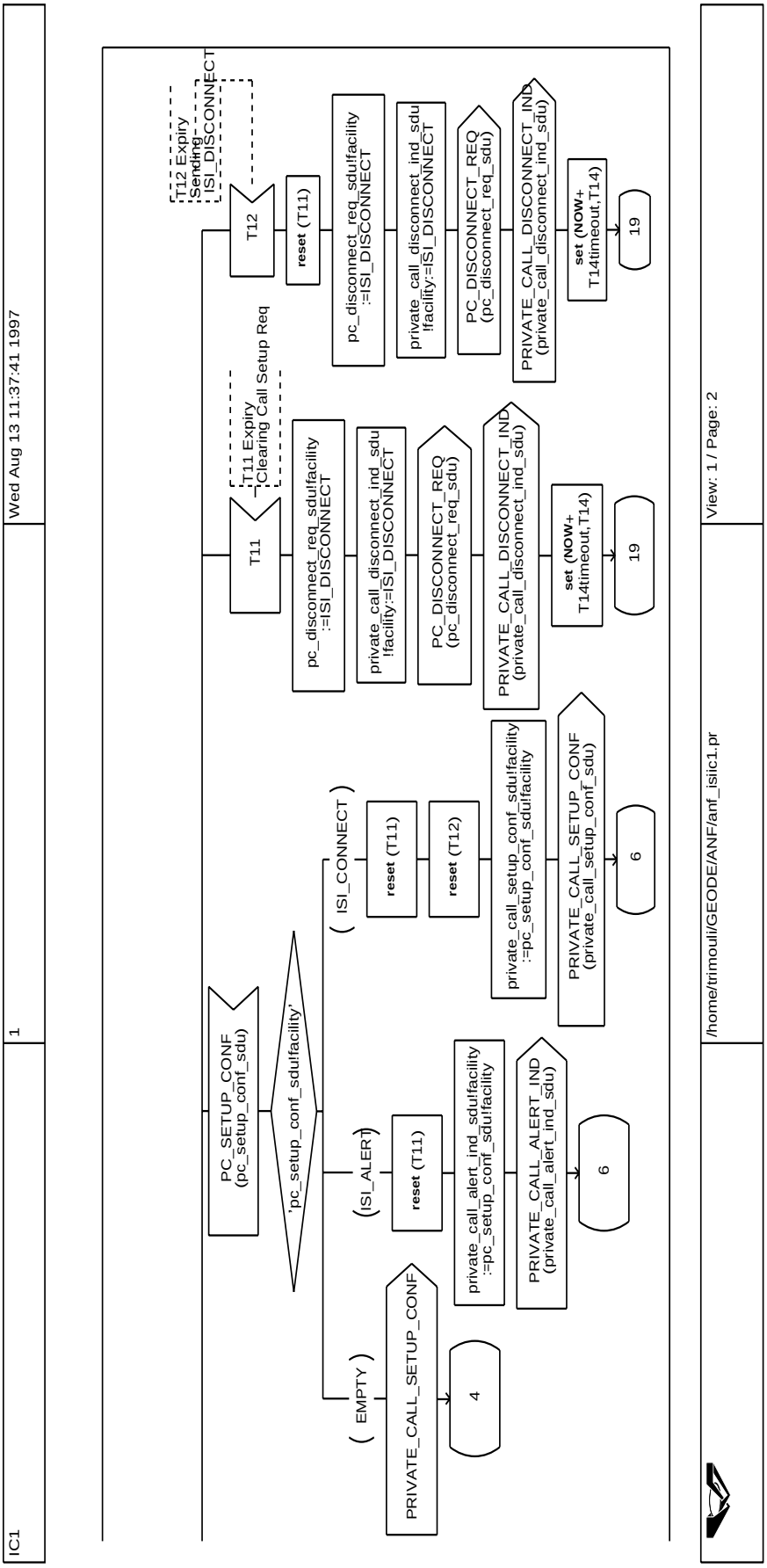
```
TIMER T11,T12,
      T13,T14,
      T15;

DCL T11timeout DURATION :=3;
DCL T12timeout DURATION :=5;
DCL T13timeout,
    T14timeout,
    T15timeout DURATION;


DCL pc_setup_req_sdu PC_SETUP_REQ_SDU;
DCL pc_setup_ind_sdu PC_SETUP_IND_SDU;
DCL pc_setup_resp_sdu PC_SETUP_RESP_SDU;
DCL pc_setup_conf_sdu PC_SETUP_CONF_SDU;
DCL pc_more_information_req_sdu PC_MORE_INFORMATION_REQ_SDU;
DCL pc_more_information_ind_sdu PC_MORE_INFORMATION_IND_SDU;
DCL pc_information_req_sdu PC_INFORMATION_REQ_SDU;
DCL pc_information_ind_sdu PC_INFORMATION_IND_SDU;
DCL pc_proceed_req_sdu PC_PROCEED_REQ_SDU;
DCL pc_proceed_ind_sdu PC_PROCEED_IND_SDU;
DCL pc_alerting_req_sdu PC_ALERTING_REQ_SDU;
DCL pc_alerting_ind_sdu PC_ALERTING_IND_SDU;
DCL pc_progress_req_sdu PC_PROGRESS_REQ_SDU;
DCL pc_progress_ind_sdu PC_PROGRESS_IND_SDU;
DCL pc_reject_req_sdu PC_REJECT_REQ_SDU;
DCL pc_reject_ind_sdu PC_REJECT_IND_SDU;
DCL pc_disconnect_req_sdu PC_DISCONNECT_REQ_SDU;
DCL pc_disconnect_ind_sdu PC_DISCONNECT_IND_SDU;
DCL pc_release_req_sdu PC_RELEASE_REQ_SDU;
DCL pc_release_ind_sdu PC_RELEASE_IND_SDU;
DCL pc_timeout_ind_sdu PC_TIMEOUT_IND_SDU;
DCL pc_dl_reset_ind_sdu PC_DL_RESET_IND_SDU;
DCL gf_facility_req_sdu GF_FACILITY_REQ_SDU;
DCL gf_facility_ind_sdu GF_FACILITY_IND_SDU;

DCL private_call_setup_req_sdu  PRIVATE_CALL_SETUP_REQ_SDU;
DCL private_call_setup_ind_sdu PRIVATE_CALL_SETUP_IND_SDU;
DCL private_call_setup_resp_sdu PRIVATE_CALL_SETUP_RESP_SDU;
DCL private_call_setup_conf_sdu PRIVATE_CALL_SETUP_CONF_SDU;
DCL private_call_proceed_req_sdu PRIVATE_CALL_PROCEED_REQ_SDU;
DCL private_call_alert_req_sdu PRIVATE_CALL_ALERT_REQ_SDU;
DCL private_call_alert_ind_sdu PRIVATE_CALL_ALERT_IND_SDU;
DCL private_call_complete_req_sdu PRIVATE_CALL_COMPLETE_REQ_SDU;
DCL private_call_complete_ind_sdu PRIVATE_CALL_COMPLETE_IND_SDU;
DCL private_call_maintenance_req_sdu PRIVATE_CALL_MAINTENANCE_REQ_SDU;
DCL private_call_maintenance_ind_sdu PRIVATE_CALL_MAINTENANCE_IND_SDU;
DCL private_call_maintenance_resp_sdu PRIVATE_CALL_MAINTENANCE_RESP_SDU;
DCL private_call_maintenance_conf_sdu PRIVATE_CALL_MAINTENANCE_CONF_SDU;
DCL private_call_remote_clearing_req_sdu PRIVATE_CALL_REMOTE_CLEARING_REQ_SDU;
DCL private_call_remote_clearing_ind_sdu PRIVATE_CALL_REMOTE_CLEARING_IND_SDU;
DCL private_call_remote_clearing_resp_sdu PRIVATE_CALL_REMOTE_CLEARING_RESP_SDU;
DCL private_call_remote_clearing_conf_sdu PRIVATE_CALL_REMOTE_CLEARING_CONF_SDU;
DCL private_call_disconnect_req_sdu PRIVATE_CALL_DISCONNECT_REQ_SDU;
DCL private_call_disconnect_ind_sdu PRIVATE_CALL_DISCONNECT_IND_SDU;
DCL private_call_disconnect_resp_sdu PRIVATE_CALL_DISCONNECT_RESP_SDU;
DCL private_call_disconnect_conf_sdu PRIVATE_CALL_DISCONNECT_CONF_SDU;
DCL private_call_transfer_req_sdu PRIVATE_CALL_TRANSFER_REQ_SDU;
DCL private_call_transfer_ind_sdu PRIVATE_CALL_TRANSFER_IND_SDU;
DCL private_call_transfer_resp_sdu PRIVATE_CALL_TRANSFER_RESP_SDU;
DCL private_call_transfer_conf_sdu PRIVATE_CALL_TRANSFER_CONF_SDU;
```

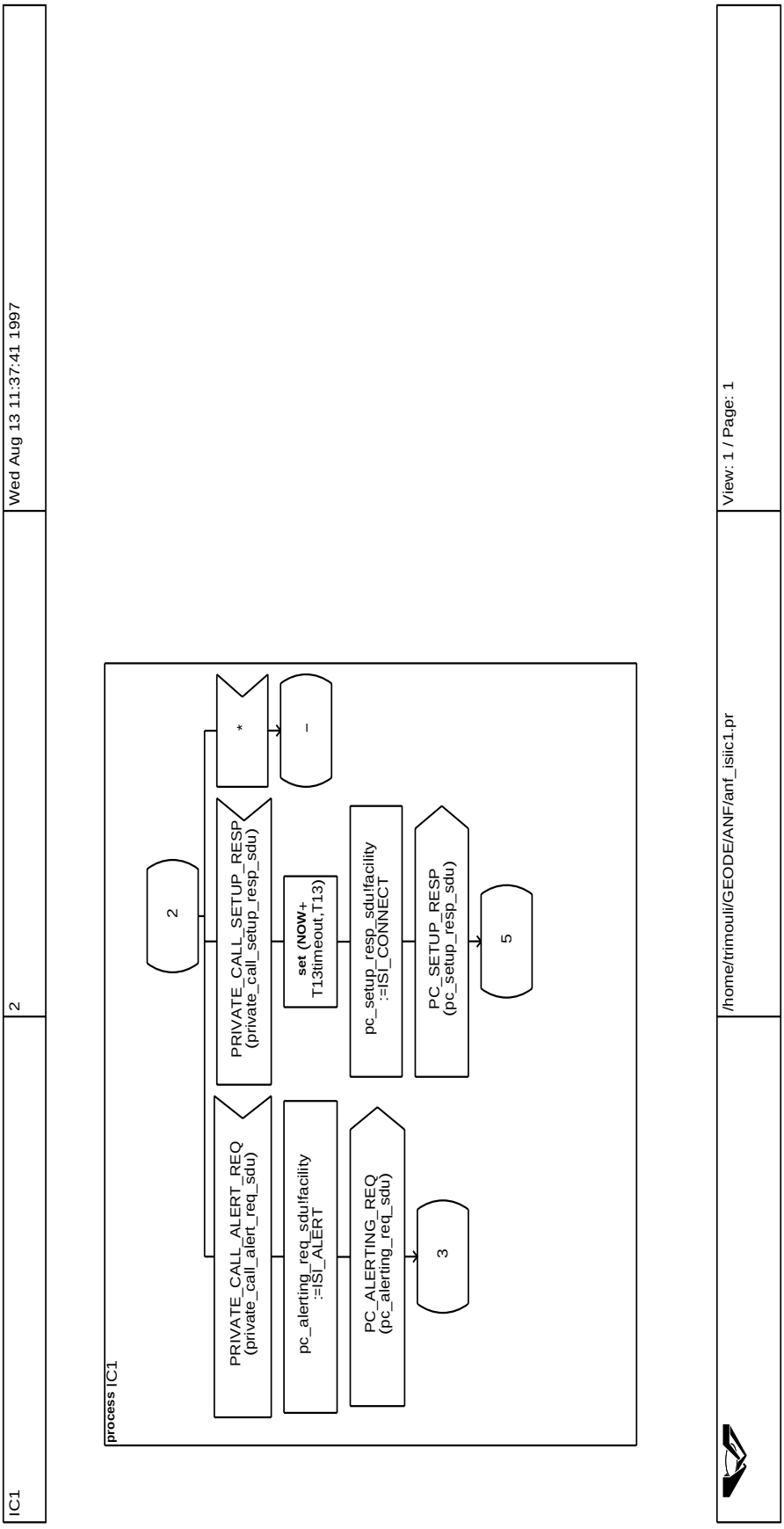
Init_Isi_Signals

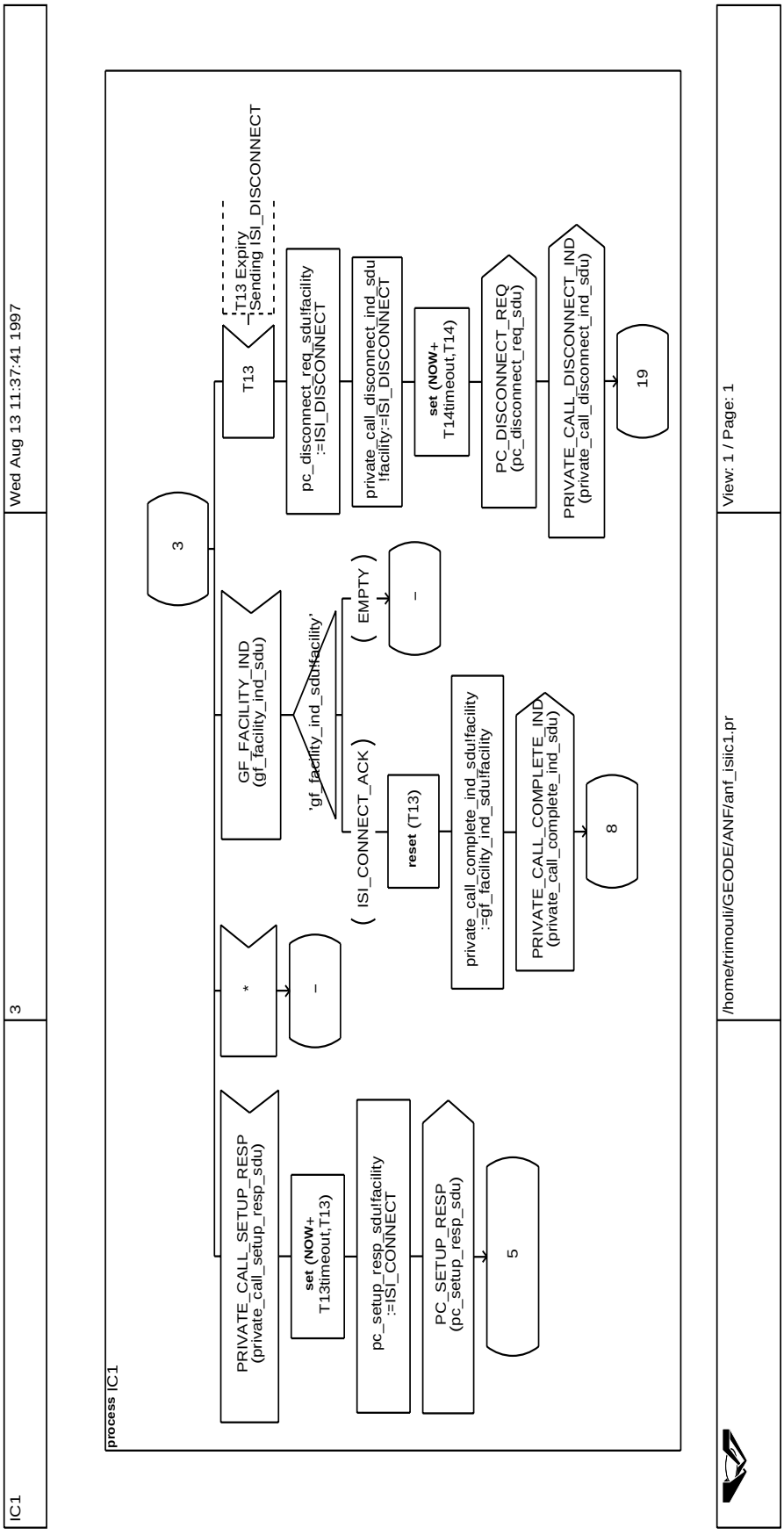
	/home/trimouli/GEODE/ANF/anf_isiic1.pr	View: 1 / Page: 1

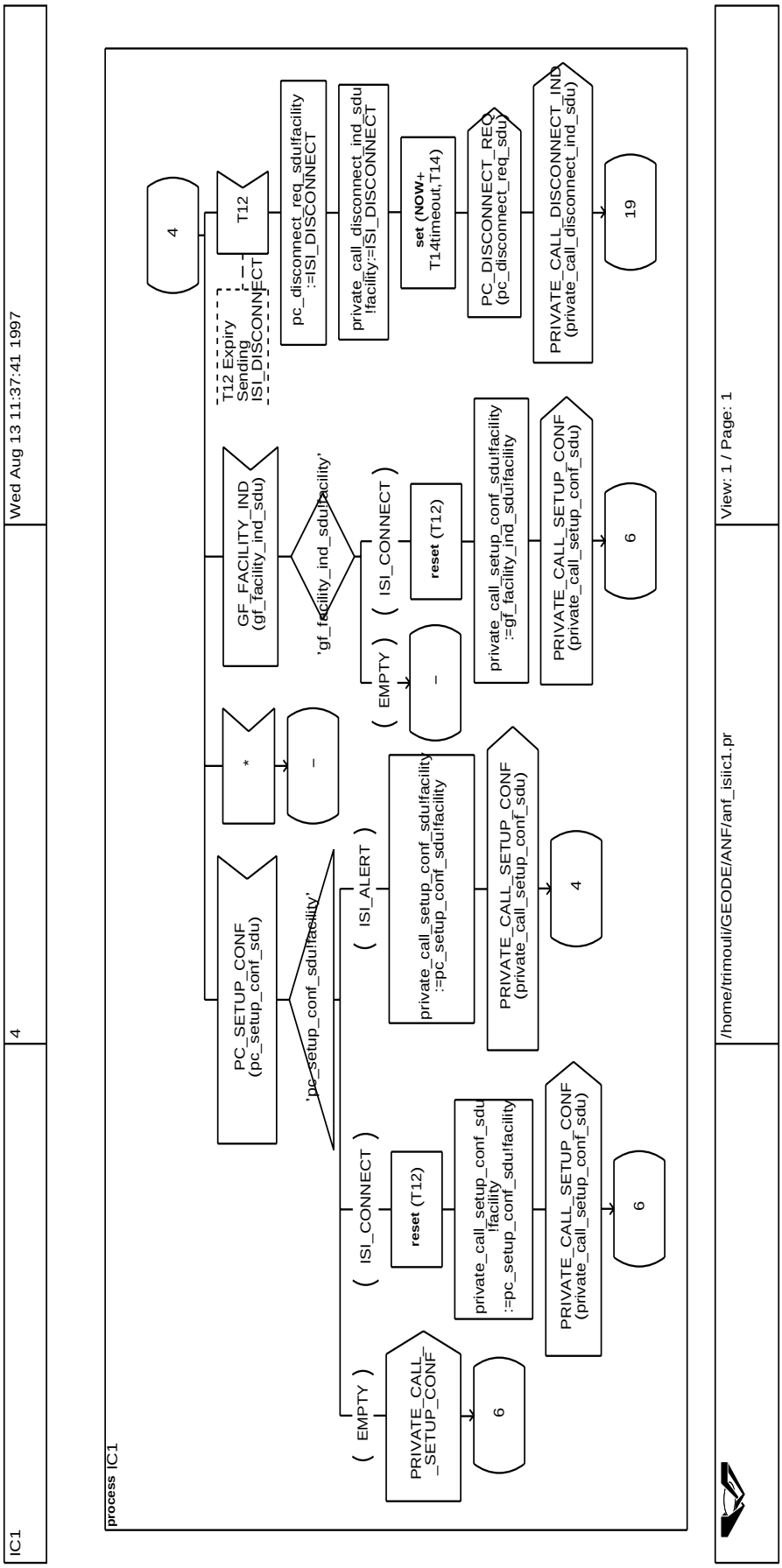


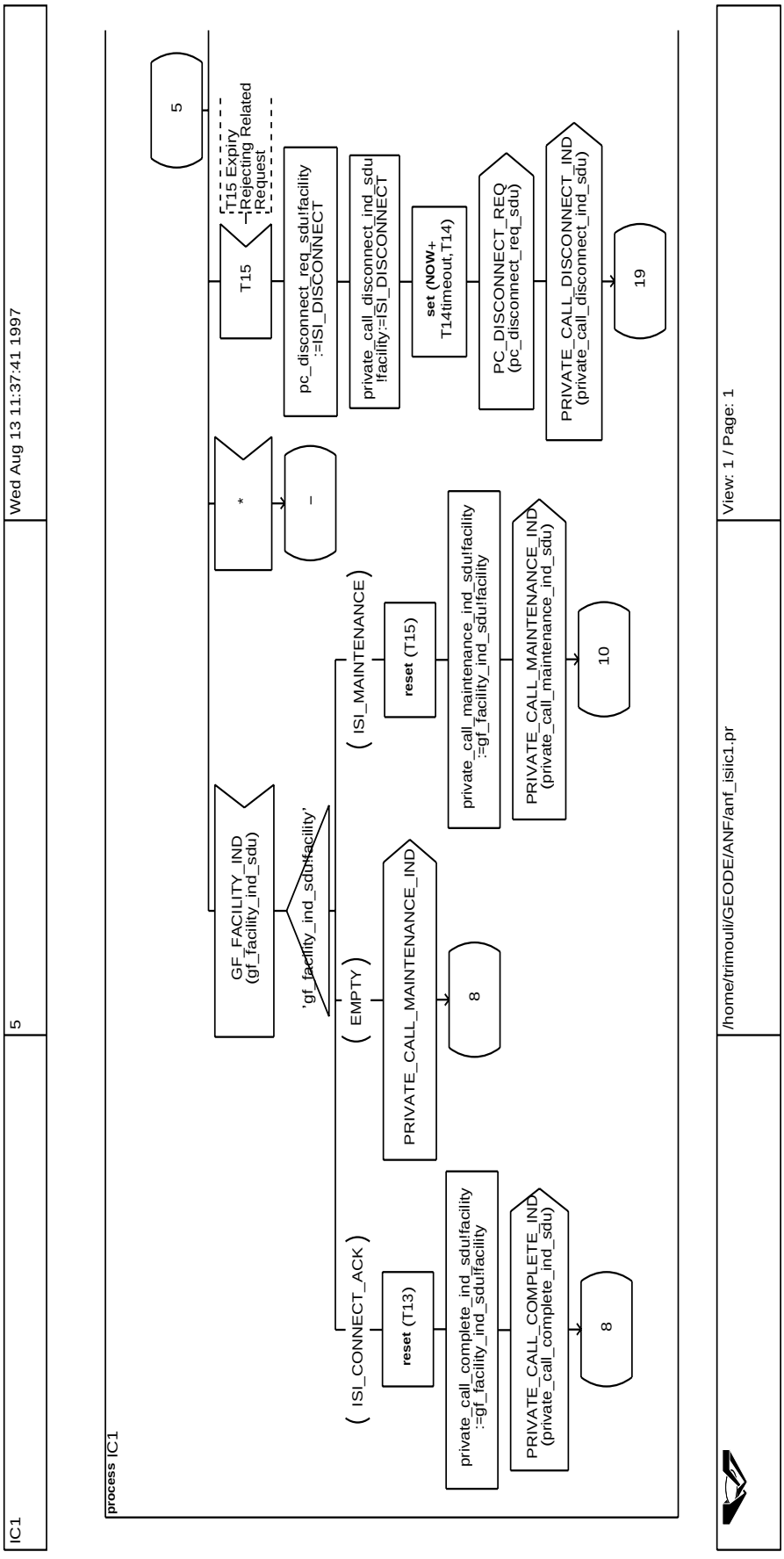


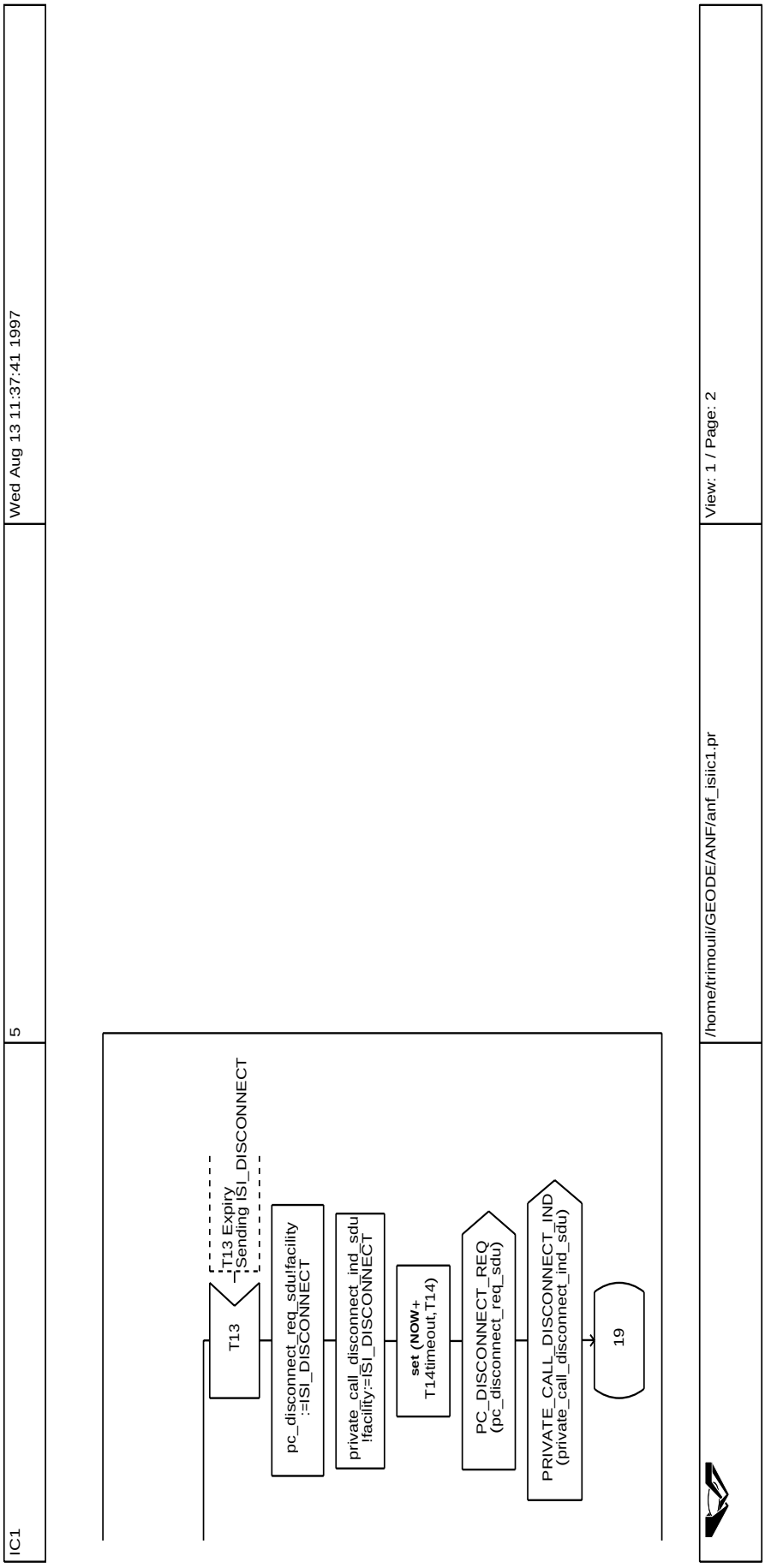


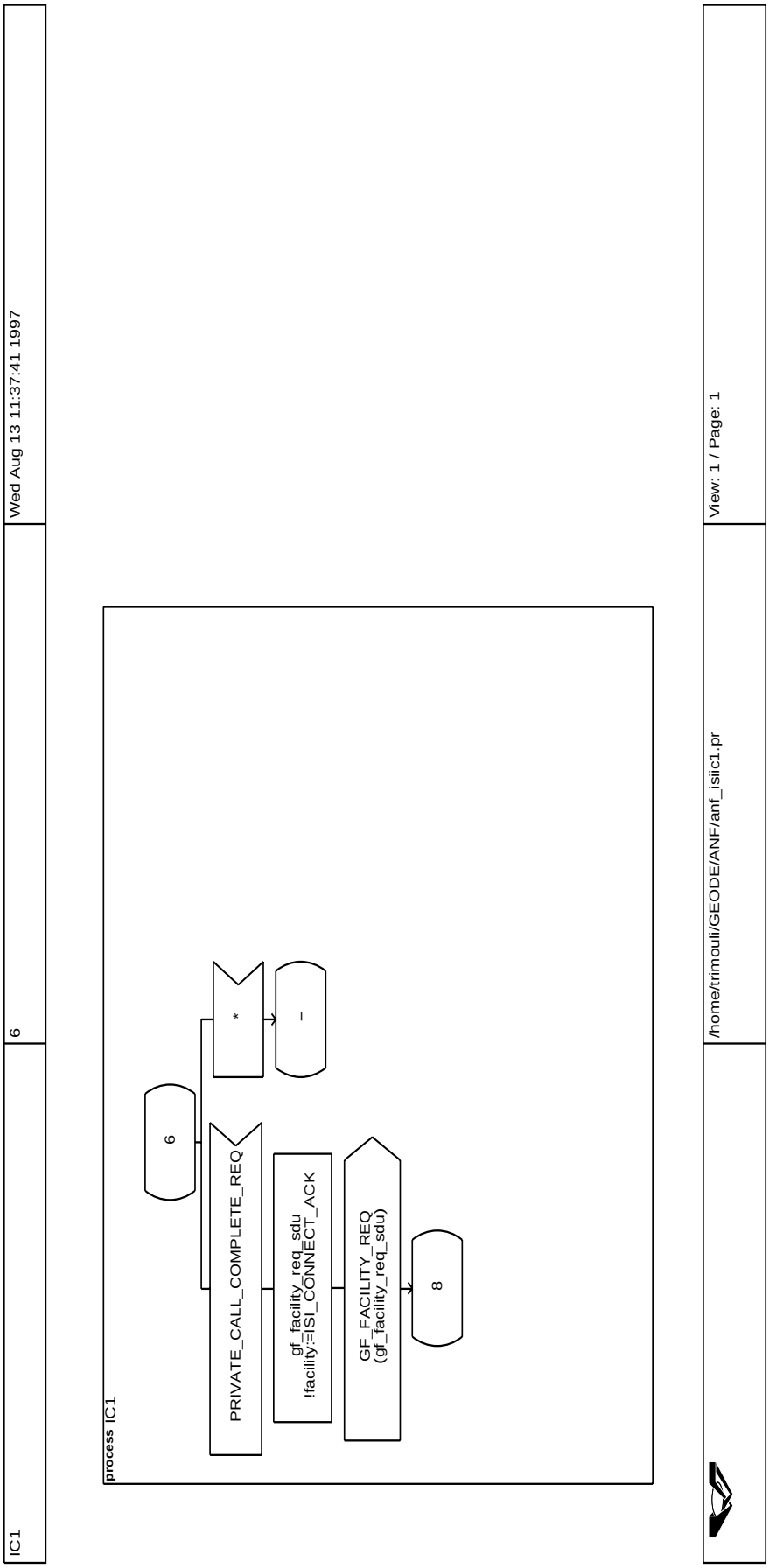


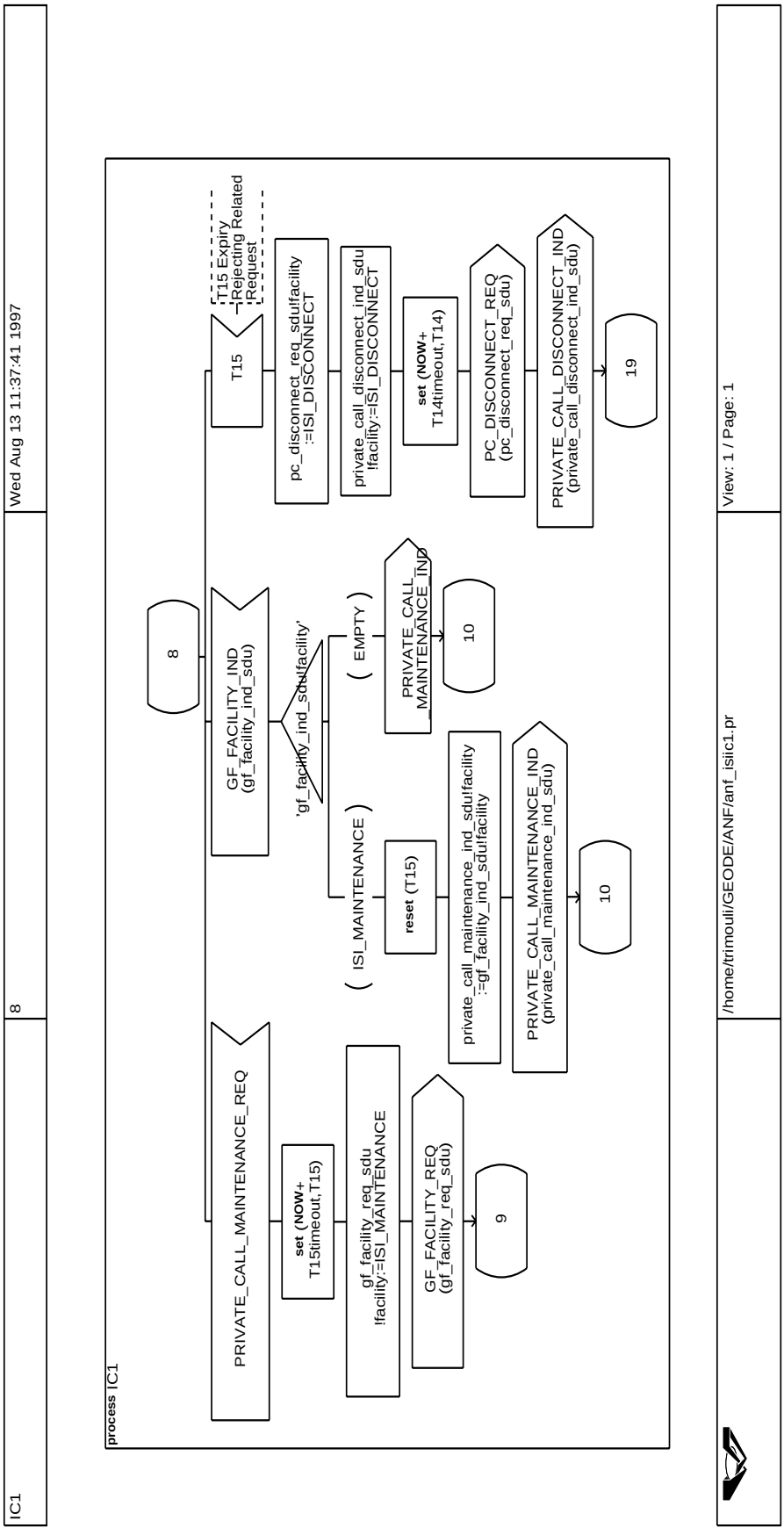


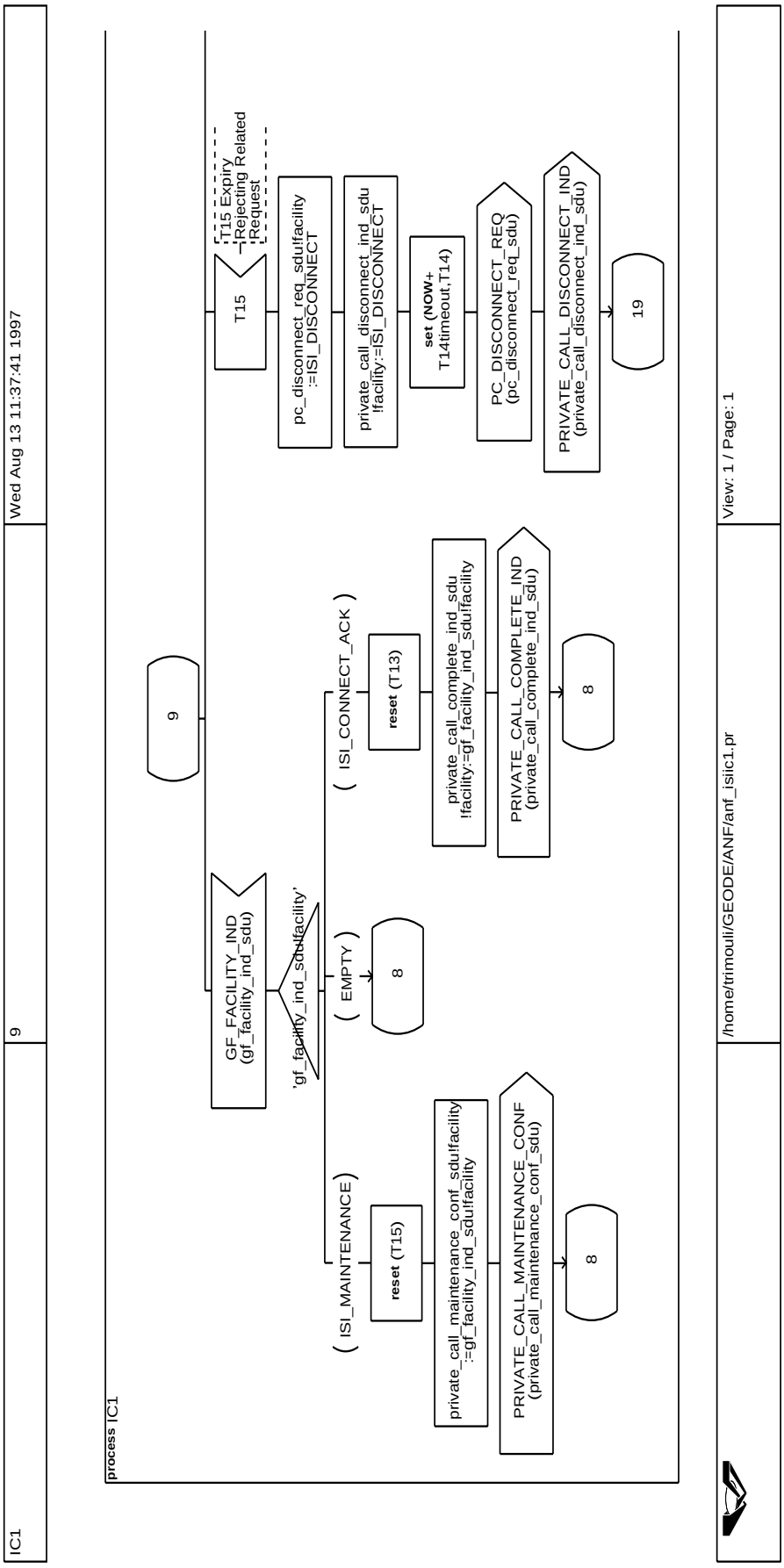


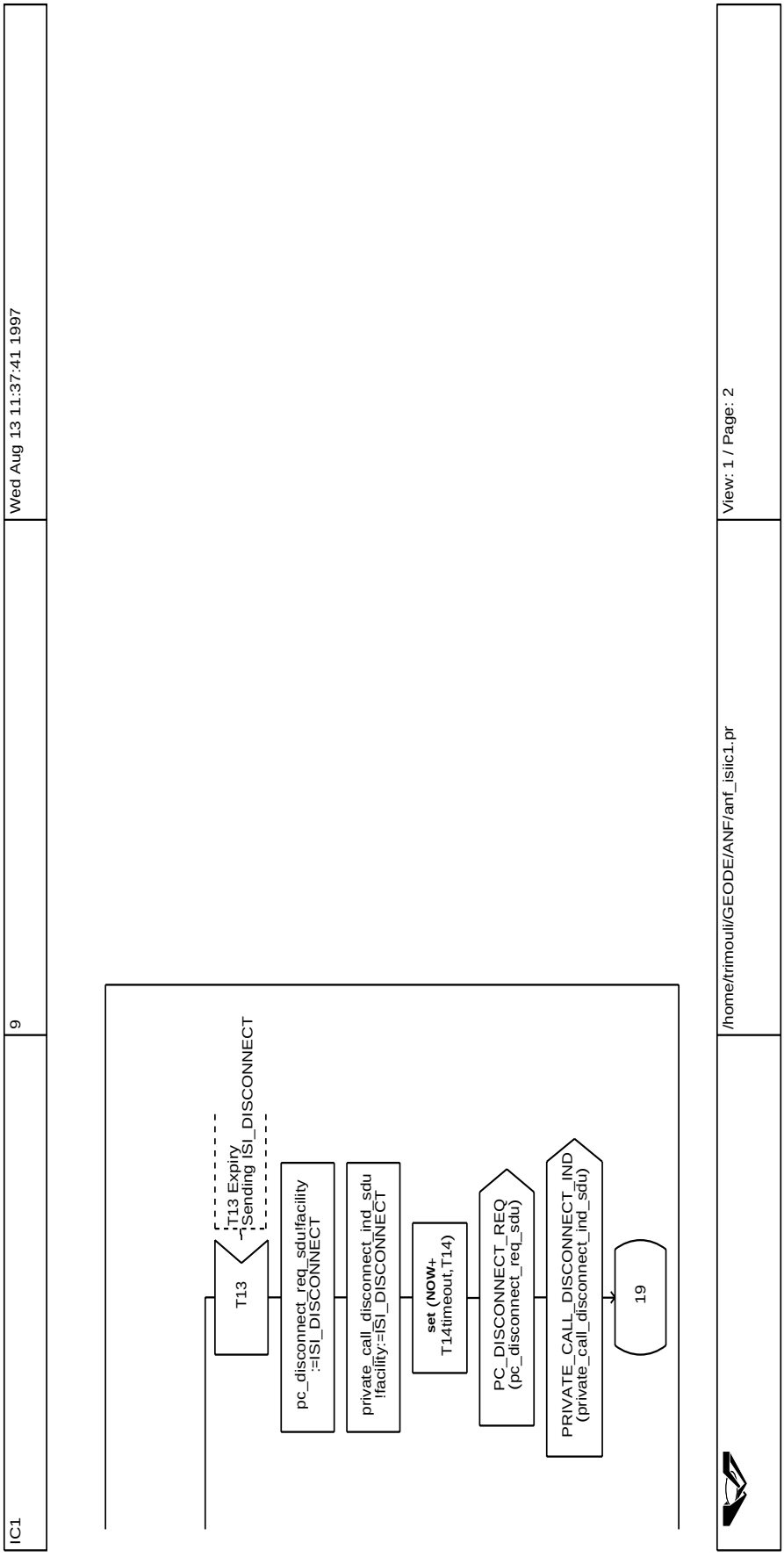


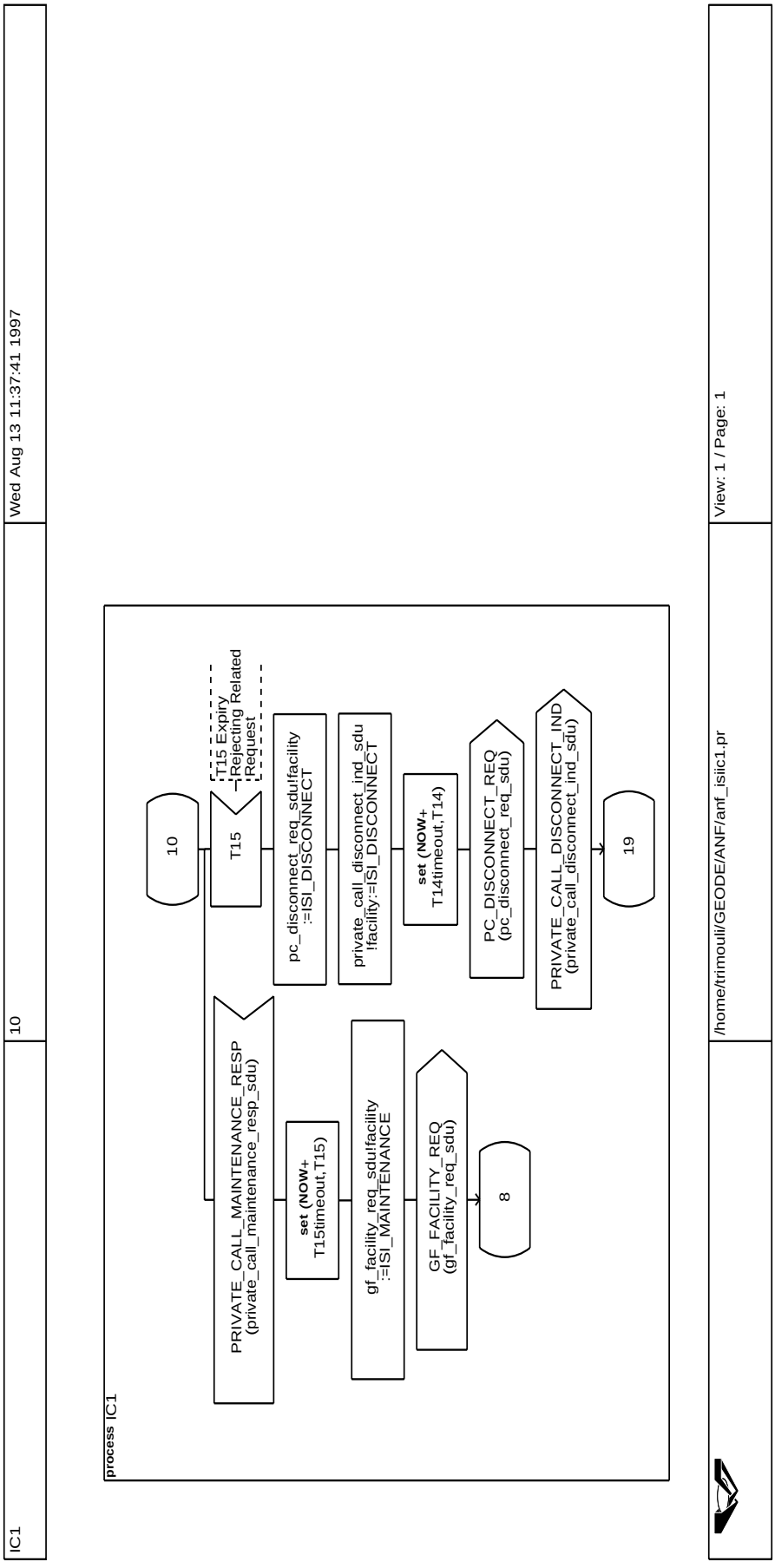




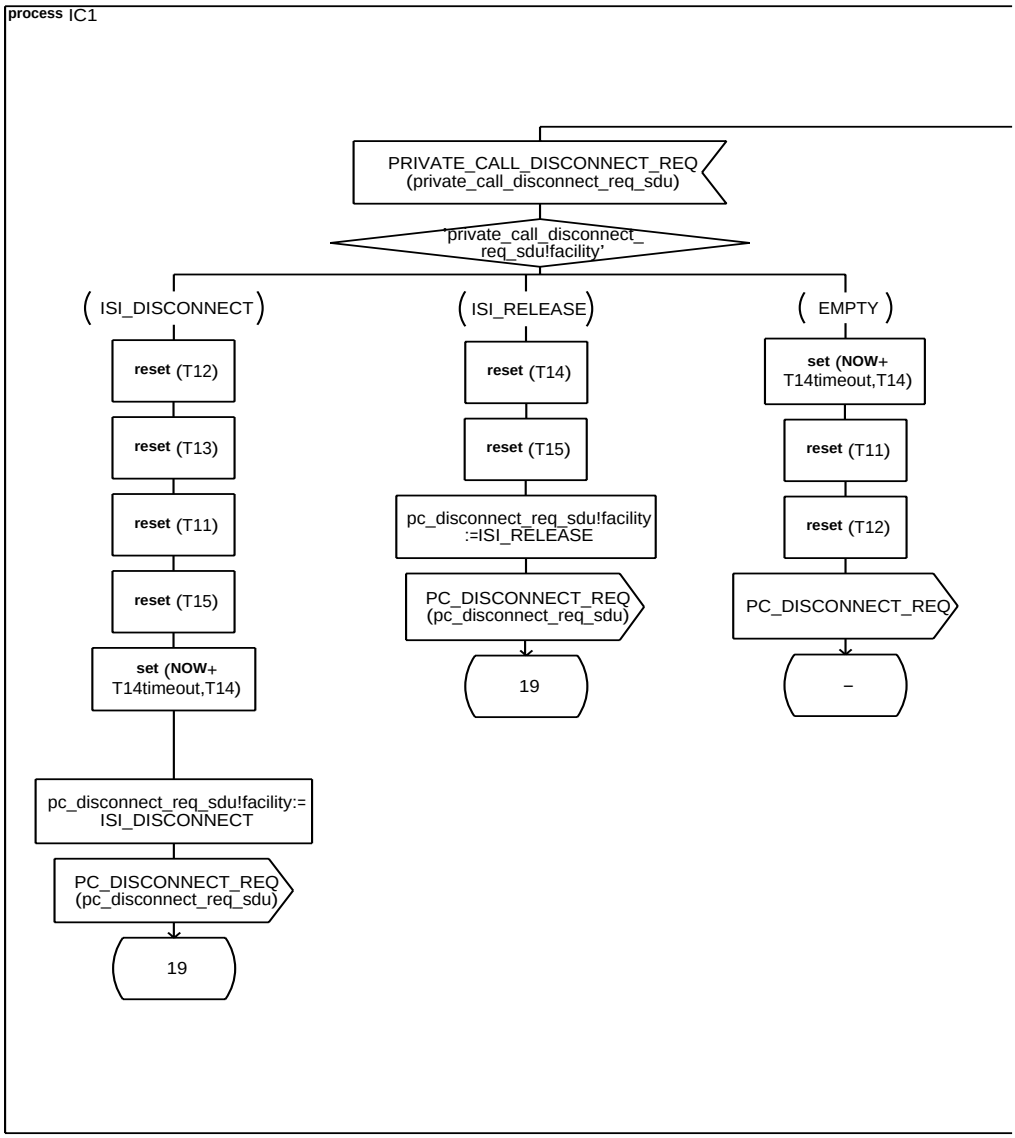






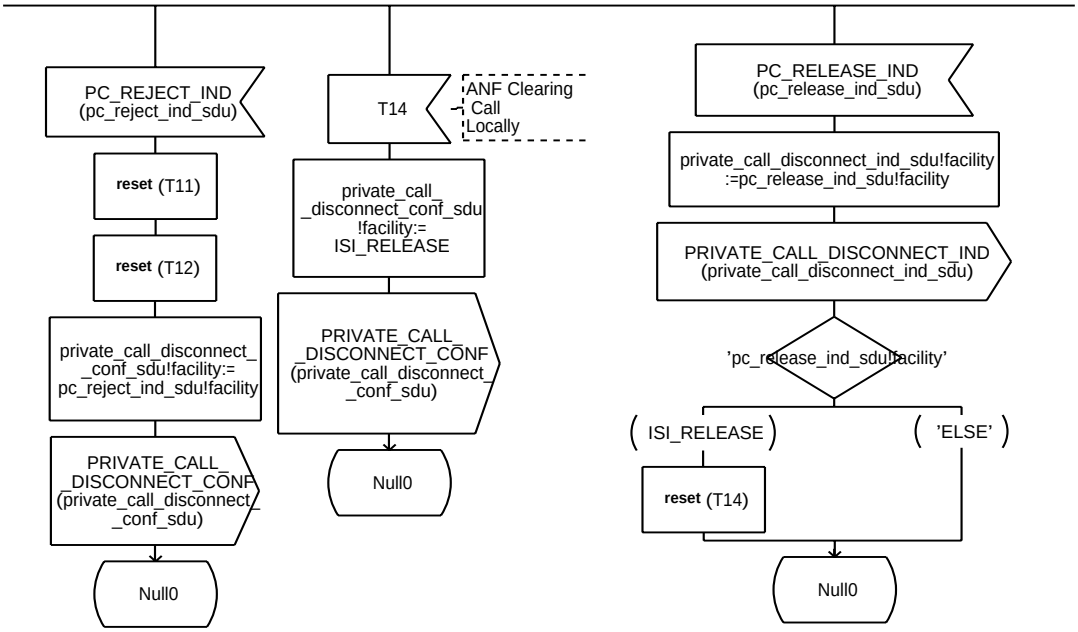


IC1	*0_19	Wed Aug 13 11:37:41 1997



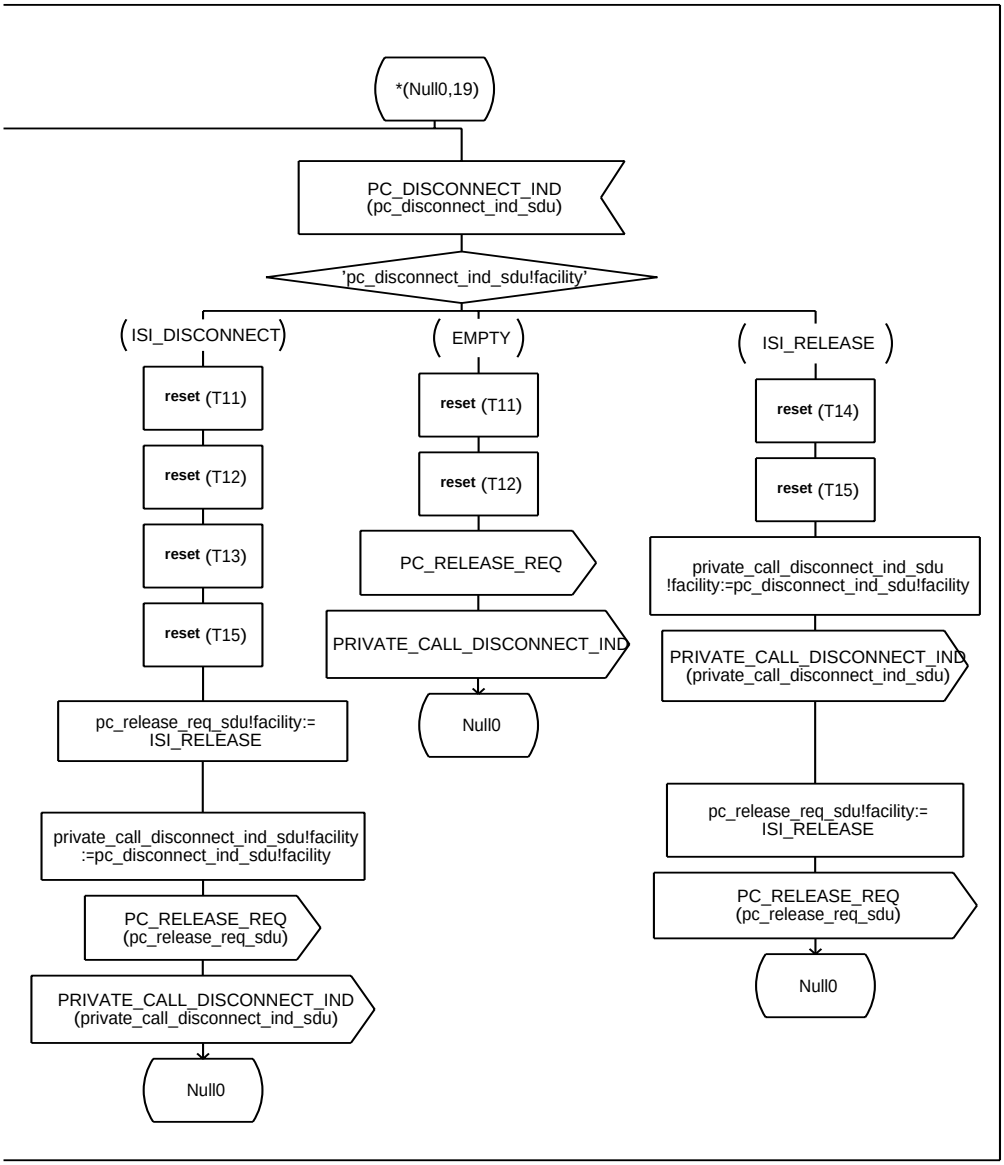
	/home/trimouli/GEODE/ANF/anf_isiic1.pr	View: 1 / Page: 1

IC1	*0_19	Thu Aug 14 14:47:16 1997

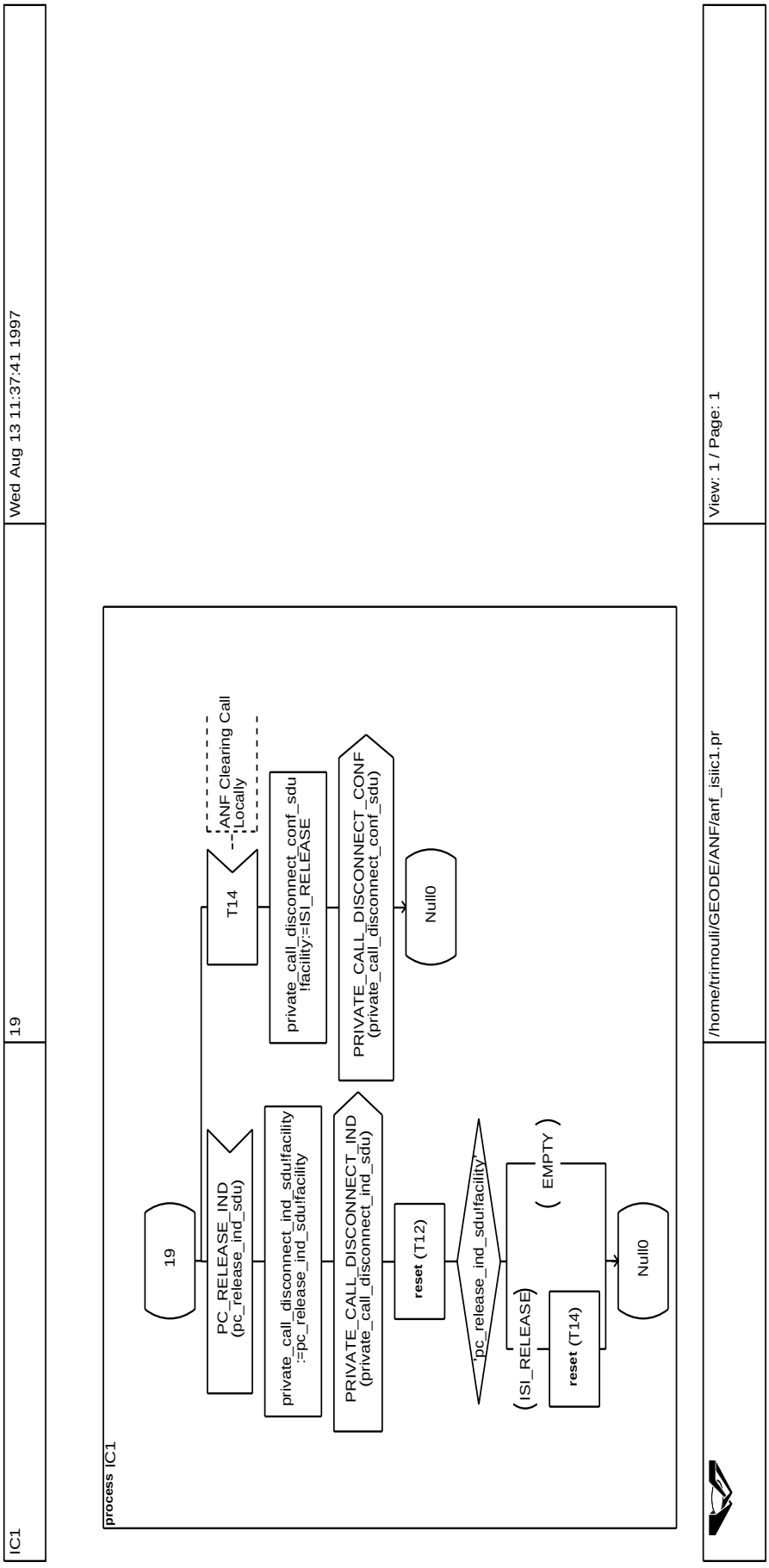


	/home/trimouli/GEODE/ANF/anf_isiic1.pr	View: 1 / Page: 2

IC1	*0_19	Wed Aug 13 11:37:41 1997

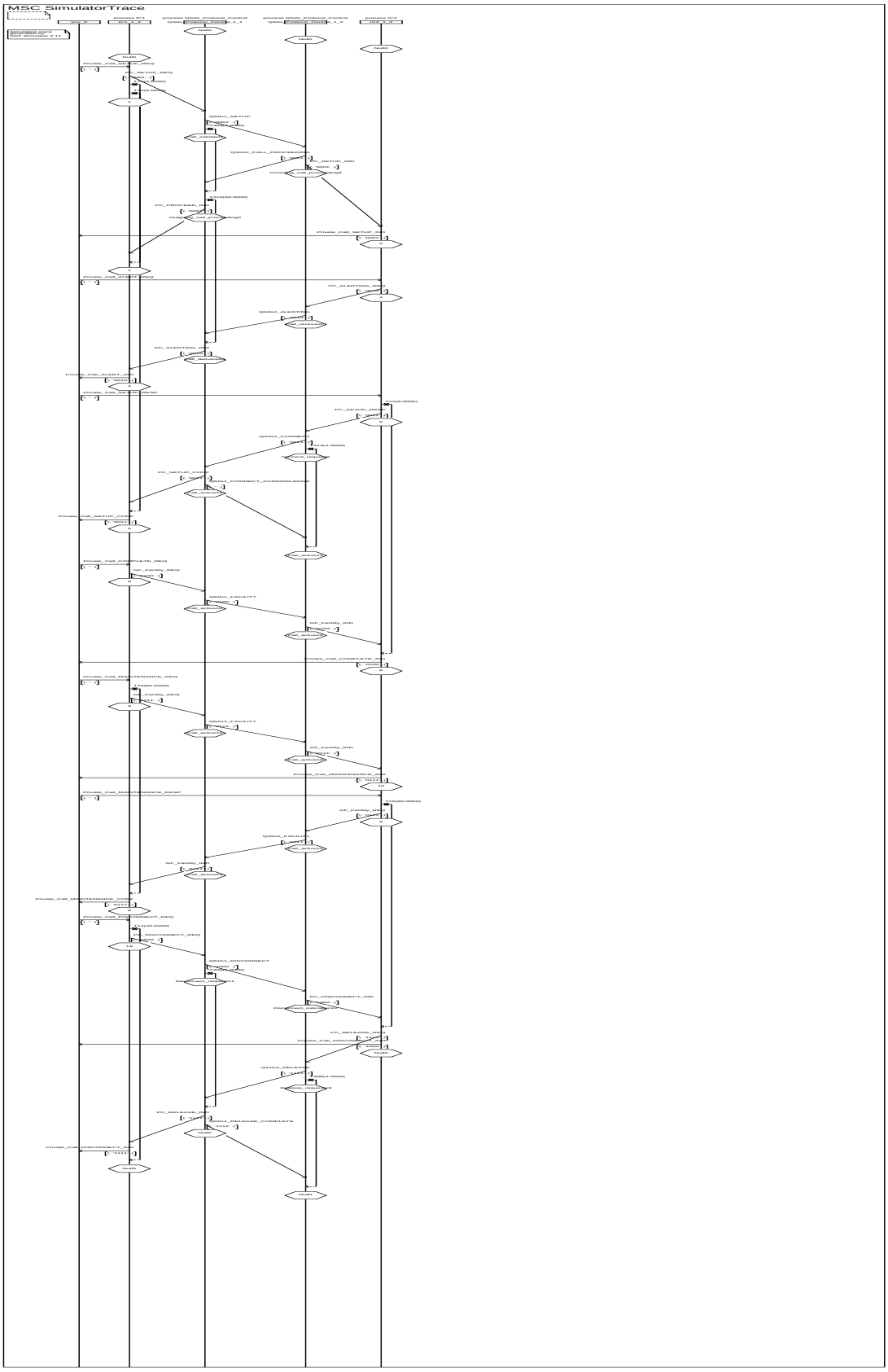


	/home/trimouli/GEODE/ANF/anf_isiic1.pr	View: 1 / Page: 3



5. Test suite structure and test purposes (informative)

The ISI network layer test suite shall follow the same structure as defined in [12] [13], thus testing the correct transportation of ISI service data units. The actual end-to-end communication testing is out of the scope of this suite.



6. Abstract test suite for the inter-system interface overview (informative)

The following part includes the static data declaration of an abstract test suite for the inter-system interface. ANF primitives are indicated as a generic view. ASN.1 data are provided in order to be used together with SDL for the purpose of a TTCN suite implementation.

7. TTCN machine processable form of the abstract test suite (informative)

The TTCN.MP representation corresponding to the TETRAPOL ISI abstract test suite is as follows

$Suite

$SuiteId qsig_anf_sdt

$SuiteOverviewPart

$Begin_SuiteStructure

$SuiteId qsig_anf_sdt

$StandardsRef /* */

$PICSref /* */

$PIXITref /* */

$TestMethods /* */

$Comment /* */

$Structure&Objectives

$End_Structure&Objectives

$Comment /* */

$End_SuiteStructure

$End_SuiteOverviewPart

$DeclarationsPart

$TS_TypeDefs

$ASN1_TypeDefs

$Begin_ASN1_TypeDef

$ASN1_TypeId CharString

$ASN1_TypeDefinition

IA5String

$End_ASN1_TypeDefinition

$Comment /* */

$End_ASN1_TypeDef

$Begin_ASN1_TypeDef

$ASN1_TypeId FCY

$ASN1_TypeDefinition

CharString

$End_ASN1_TypeDefinition

$Comment /* */

$End_ASN1_TypeDef

$Begin_ASN1_TypeDef

$ASN1_TypeId C

$ASN1_TypeDefinition

SEQUENCE { facility FCY }

$End_ASN1_TypeDefinition

$Comment /* */

$End_ASN1_TypeDef

$Begin_ASN1_TypeDef

$ASN1_TypeId PRIVATE_CALL_MAINTENANCE_RESP_SDU

$ASN1_TypeDefinition

C

$End_ASN1_TypeDefinition

$Comment /* */

$End_ASN1_TypeDef

$Begin_ASN1_TypeDef

$ASN1_TypeId PRIVATE_CALL_SETUP_IND_SDU

$ASN1_TypeDefinition

C

$End_ASN1_TypeDefinition

$Comment /* */

$End_ASN1_TypeDef

$Begin_ASN1_TypeDef

$ASN1_TypeId PRIVATE_CALL_SETUP_RESP_SDU

$ASN1_TypeDefinition

C

$End_ASN1_TypeDefinition

$Comment /* */

$End_ASN1_TypeDef

$Begin_ASN1_TypeDef

$ASN1_TypeId PRIVATE_CALL_SETUP_CONF_SDU

$ASN1_TypeDefinition

C

$End_ASN1_TypeDefinition

$Comment /* */

$End_ASN1_TypeDef

$Begin_ASN1_TypeDef

$ASN1_TypeId PRIVATE_CALL_PROCEED_REQ_SDU

$ASN1_TypeDefinition

C

$End_ASN1_TypeDefinition

$Comment /* */

$End_ASN1_TypeDef

$Begin_ASN1_TypeDef

$ASN1_TypeId PRIVATE_CALL_ALERT_REQ_SDU

$ASN1_TypeDefinition

C

$End_ASN1_TypeDefinition

$Comment /* */

$End_ASN1_TypeDef

$Begin_ASN1_TypeDef

$ASN1_TypeId PRIVATE_CALL_ALERT_IND_SDU

$ASN1_TypeDefinition

C

$End_ASN1_TypeDefinition

$Comment /* */

$End_ASN1_TypeDef

$Begin_ASN1_TypeDef

$ASN1_TypeId PRIVATE_CALL_COMPLETE_REQ_SDU

$ASN1_TypeDefinition

C

$End_ASN1_TypeDefinition

$Comment /* */

$End_ASN1_TypeDef

$Begin_ASN1_TypeDef

$ASN1_TypeId PRIVATE_CALL_COMPLETE_IND_SDU

$ASN1_TypeDefinition

C

$End_ASN1_TypeDefinition

$Comment /* */

$End_ASN1_TypeDef

$Begin_ASN1_TypeDef

$ASN1_TypeId PRIVATE_CALL_MAINTENANCE_REQ_SDU

$ASN1_TypeDefinition

C

$End_ASN1_TypeDefinition

$Comment /* */

$End_ASN1_TypeDef

$Begin_ASN1_TypeDef

$ASN1_TypeId PRIVATE_CALL_MAINTENANCE_IND_SDU

$ASN1_TypeDefinition

C

$End_ASN1_TypeDefinition

$Comment /* */

$End_ASN1_TypeDef

$Begin_ASN1_TypeDef

$ASN1_TypeId PRIVATE_CALL_SETUP_REQ_SDU

$ASN1_TypeDefinition

C

$End_ASN1_TypeDefinition

$Comment /* */

$End_ASN1_TypeDef

$Begin_ASN1_TypeDef

$ASN1_TypeId PRIVATE_CALL_MAINTENANCE_CONF_SDU

$ASN1_TypeDefinition

C

$End_ASN1_TypeDefinition

$Comment /* */

$End_ASN1_TypeDef

$Begin_ASN1_TypeDef

$ASN1_TypeId PRIVATE_CALL_REMOTE_CLEARING_REQ_SDU

$ASN1_TypeDefinition

C

$End_ASN1_TypeDefinition

$Comment /* */

$End_ASN1_TypeDef

$Begin_ASN1_TypeDef

$ASN1_TypeId PRIVATE_CALL_REMOTE_CLEARING_IND_SDU

$ASN1_TypeDefinition

C

$End_ASN1_TypeDefinition

$Comment /* */

$End_ASN1_TypeDef

$Begin_ASN1_TypeDef

$ASN1_TypeId PRIVATE_CALL_REMOTE_CLEARING_RESP_SDU

$ASN1_TypeDefinition

C

$End_ASN1_TypeDefinition

$Comment /* */

$End_ASN1_TypeDef

$Begin_ASN1_TypeDef

$ASN1_TypeId PRIVATE_CALL_REMOTE_CLEARING_CONF_SDU

$ASN1_TypeDefinition

C

$End_ASN1_TypeDefinition

$Comment /* */

$End_ASN1_TypeDef

$Begin_ASN1_TypeDef

$ASN1_TypeId PRIVATE_CALL_DISCONNECT_REQ_SDU

$ASN1_TypeDefinition

C

$End_ASN1_TypeDefinition

$Comment /* */

$End_ASN1_TypeDef

$Begin_ASN1_TypeDef

$ASN1_TypeId PRIVATE_CALL_DISCONNECT_IND_SDU

$ASN1_TypeDefinition

C

$End_ASN1_TypeDefinition

$Comment /* */

$End_ASN1_TypeDef

$Begin_ASN1_TypeDef

$ASN1_TypeId PRIVATE_CALL_DISCONNECT_RESP_SDU

$ASN1_TypeDefinition

C

$End_ASN1_TypeDefinition

$Comment /* */

$End_ASN1_TypeDef

$Begin_ASN1_TypeDef

$ASN1_TypeId PRIVATE_CALL_DISCONNECT_CONF_SDU

$ASN1_TypeDefinition

C

$End_ASN1_TypeDefinition

$Comment /* */

$End_ASN1_TypeDef

$Begin_ASN1_TypeDef

$ASN1_TypeId PRIVATE_CALL_TRANSFER_REQ_SDU

$ASN1_TypeDefinition

C

$End_ASN1_TypeDefinition

$Comment /* */

$End_ASN1_TypeDef

$Begin_ASN1_TypeDef

$ASN1_TypeId PRIVATE_CALL_TRANSFER_IND_SDU

$ASN1_TypeDefinition

C

$End_ASN1_TypeDefinition

$Comment /* */

$End_ASN1_TypeDef

$Begin_ASN1_TypeDef

$ASN1_TypeId PRIVATE_CALL_TRANSFER_CONF_SDU

$ASN1_TypeDefinition

C

$End_ASN1_TypeDefinition

$Comment /* */

$End_ASN1_TypeDef

$Begin_ASN1_TypeDef

$ASN1_TypeId PRIVATE_CALL_TRANSFER_RESP_SDU

$ASN1_TypeDefinition

C

$End_ASN1_TypeDefinition

$Comment /* */

$End_ASN1_TypeDef

$End_ASN1_TypeDefs

$End_TS_TypeDefs

$Begin_PCO_TypeDcls

$PCO_TypeDcl

$PCO_TypeId PCO_Type

$PCO_Role LT

$End_PCO_TypeDcl

$Comment /* */

$End_PCO_TypeDcls

$Begin_PCO_Dcls

$PCO_Dcl

$PCO_Id QSIG1_DataLink

$PCO_TypeId PCO_Type

$PCO_Role LT

$End_PCO_Dcl

$PCO_Dcl

$PCO_Id QSIG2_DataLink

$PCO_TypeId PCO_Type

$PCO_Role LT

$End_PCO_Dcl

$PCO_Dcl

$PCO_Id ANF_Link2

$PCO_TypeId PCO_Type

$PCO_Role LT

$End_PCO_Dcl

$PCO_Dcl

$PCO_Id ANF_Link1

$PCO_TypeId PCO_Type

$PCO_Role LT

$End_PCO_Dcl

$Comment /* */

$End_PCO_Dcls

$ASP_TypeDefs

$ASN1_ASP_TypeDefs

$Begin_ASN1_ASP_TypeDef

$ASP_Id DL_DATA_REQUEST

$PCO_Type PCO_Type

$ASN1_TypeDefinition

SEQUENCE { }

$End_ASN1_TypeDefinition

$Comment /* */

$End_ASN1_ASP_TypeDef

$Begin_ASN1_ASP_TypeDef

$ASP_Id DL_DATA_INDICATION

$PCO_Type PCO_Type

$ASN1_TypeDefinition

SEQUENCE { }

$End_ASN1_TypeDefinition

$Comment /* */

$End_ASN1_ASP_TypeDef

$Begin_ASN1_ASP_TypeDef

$ASP_Id DL_ESTABLISH_REQUEST

$PCO_Type PCO_Type

$ASN1_TypeDefinition

SEQUENCE { }

$End_ASN1_TypeDefinition

$Comment /* */

$End_ASN1_ASP_TypeDef

$Begin_ASN1_ASP_TypeDef

$ASP_Id DL_ESTABLISH_IND

$PCO_Type PCO_Type

$ASN1_TypeDefinition

SEQUENCE { }

$End_ASN1_TypeDefinition

$Comment /* */

$End_ASN1_ASP_TypeDef

$Begin_ASN1_ASP_TypeDef

$ASP_Id DL_ESTABLISH_CONF

$PCO_Type PCO_Type

$ASN1_TypeDefinition

SEQUENCE { }

$End_ASN1_TypeDefinition

$Comment /* */

$End_ASN1_ASP_TypeDef

$Begin_ASN1_ASP_TypeDef

$ASP_Id DL_RELEASE_REQ

$PCO_Type PCO_Type

$ASN1_TypeDefinition

SEQUENCE { }

$End_ASN1_TypeDefinition

$Comment /* */

$End_ASN1_ASP_TypeDef

$Begin_ASN1_ASP_TypeDef

$ASP_Id DL_RELEASE_IND

$PCO_Type PCO_Type

$ASN1_TypeDefinition

SEQUENCE { }

$End_ASN1_TypeDefinition

$Comment /* */

```
$End_ASN1_ASP_TypeDef

$Begin_ASN1_ASP_TypeDef

$ASP_Id Private_Call_SETUP_REQ

$PCO_Type PCO_Type

$ASN1_TypeDefinition

SEQUENCE {

  pRIVATE_CALL_SETUP_REQ_SDU1  PRIVATE_CALL_SETUP_REQ_SDU

}

$End_ASN1_TypeDefinition

$Comment /*  */

$End_ASN1_ASP_TypeDef

$Begin_ASN1_ASP_TypeDef

$ASP_Id Private_Call_SETUP_RESP

$PCO_Type PCO_Type

$ASN1_TypeDefinition

SEQUENCE {

  pRIVATE_CALL_SETUP_RESP_SDU1  PRIVATE_CALL_SETUP_RESP_SDU

}

$End_ASN1_TypeDefinition

$Comment /*  */

$End_ASN1_ASP_TypeDef

$Begin_ASN1_ASP_TypeDef

$ASP_Id Private_Call_PROCEED_REQ

$PCO_Type PCO_Type

$ASN1_TypeDefinition

SEQUENCE {

  pRIVATE_CALL_PROCEED_REQ_SDU1  PRIVATE_CALL_PROCEED_REQ_SDU

}

$End_ASN1_TypeDefinition

$Comment /*  */

$End_ASN1_ASP_TypeDef
```

```
$Begin_ASN1_ASP_TypeDef

$ASP_Id Private_Call_ALERT_REQ

$PCO_Type PCO_Type

$ASN1_TypeDefinition

SEQUENCE {

  pRIVATE_CALL_ALERT_REQ_SDU1  PRIVATE_CALL_ALERT_REQ_SDU

}

$End_ASN1_TypeDefinition

$Comment /*  */

$End_ASN1_ASP_TypeDef

$Begin_ASN1_ASP_TypeDef

$ASP_Id Private_Call_COMPLETE_REQ

$PCO_Type PCO_Type

$ASN1_TypeDefinition

SEQUENCE {

  pRIVATE_CALL_COMPLETE_REQ_SDU1  PRIVATE_CALL_COMPLETE_REQ_SDU

}

$End_ASN1_TypeDefinition

$Comment /*  */

$End_ASN1_ASP_TypeDef

$Begin_ASN1_ASP_TypeDef

$ASP_Id Private_Call_MAINTENANCE_REQ

$PCO_Type PCO_Type

$ASN1_TypeDefinition

SEQUENCE {

  pRIVATE_CALL_MAINTENANCE_REQ_SDU1  PRIVATE_CALL_MAINTENANCE_REQ_SDU

}

$End_ASN1_TypeDefinition

$Comment /*  */

$End_ASN1_ASP_TypeDef

$Begin_ASN1_ASP_TypeDef
```

```
$ASP_Id Private_Call_MAINTENANCE_RESP

$PCO_Type PCO_Type

$ASN1_TypeDefinition

SEQUENCE {

  pRIVATE_CALL_MAINTENANCE_RESP_SDU1  PRIVATE_CALL_MAINTENANCE_RESP_SDU

}

$End_ASN1_TypeDefinition

$Comment /*  */

$End_ASN1_ASP_TypeDef

$Begin_ASN1_ASP_TypeDef

$ASP_Id Private_Call_REMOTE_CLEARING_REQ

$PCO_Type PCO_Type

$ASN1_TypeDefinition

SEQUENCE {

  pRIVATE_CALL_REMOTE_CLEARING_REQ_SDU1
PRIVATE_CALL_REMOTE_CLEARING_REQ_SDU

}

$End_ASN1_TypeDefinition

$Comment /*  */

$End_ASN1_ASP_TypeDef

$Begin_ASN1_ASP_TypeDef

$ASP_Id Private_Call_REMOTE_CLEARING_RESP

$PCO_Type PCO_Type

$ASN1_TypeDefinition

SEQUENCE {

  pRIVATE_CALL_REMOTE_CLEARING_RESP_SDU1
PRIVATE_CALL_REMOTE_CLEARING_RESP_SDU

}

$End_ASN1_TypeDefinition

$Comment /*  */

$End_ASN1_ASP_TypeDef

$Begin_ASN1_ASP_TypeDef
```

```
$ASP_Id Private_Call_DISCONNECT_REQ

$PCO_Type PCO_Type

$ASN1_TypeDefinition

SEQUENCE {

  pRIVATE_CALL_DISCONNECT_REQ_SDU1  PRIVATE_CALL_DISCONNECT_REQ_SDU

}

$End_ASN1_TypeDefinition

$Comment /*  */

$End_ASN1_ASP_TypeDef

$Begin_ASN1_ASP_TypeDef

$ASP_Id Private_Call_DISCONNECT_RESP

$PCO_Type PCO_Type

$ASN1_TypeDefinition

SEQUENCE {

  pRIVATE_CALL_DISCONNECT_RESP_SDU1  PRIVATE_CALL_DISCONNECT_RESP_SDU

}

$End_ASN1_TypeDefinition

$Comment /*  */

$End_ASN1_ASP_TypeDef

$Begin_ASN1_ASP_TypeDef

$ASP_Id Private_Call_TRANSFER_REQ

$PCO_Type PCO_Type

$ASN1_TypeDefinition

SEQUENCE {

  pRIVATE_CALL_TRANSFER_REQ_SDU1  PRIVATE_CALL_TRANSFER_REQ_SDU

}

$End_ASN1_TypeDefinition

$Comment /*  */

$End_ASN1_ASP_TypeDef

$Begin_ASN1_ASP_TypeDef

$ASP_Id Private_Call_TRANSFER_RESP
```

```
$PCO_Type PCO_Type
$ASN1_TypeDefinition
SEQUENCE {
  pRIVATE_CALL_TRANSFER_RESP_SDU1  PRIVATE_CALL_TRANSFER_RESP_SDU
}
$End_ASN1_TypeDefinition
$Comment /*  */
$End_ASN1_ASP_TypeDef
$Begin_ASN1_ASP_TypeDef
$ASP_Id Private_Call_SETUP_IND
$PCO_Type PCO_Type
$ASN1_TypeDefinition
SEQUENCE {
  pRIVATE_CALL_SETUP_IND_SDU1  PRIVATE_CALL_SETUP_IND_SDU
}
$End_ASN1_TypeDefinition
$Comment /*  */
$End_ASN1_ASP_TypeDef
$Begin_ASN1_ASP_TypeDef
$ASP_Id Private_Call_SETUP_CONF
$PCO_Type PCO_Type
$ASN1_TypeDefinition
SEQUENCE {
  pRIVATE_CALL_SETUP_CONF_SDU1  PRIVATE_CALL_SETUP_CONF_SDU
}
$End_ASN1_TypeDefinition
$Comment /*  */
$End_ASN1_ASP_TypeDef
$Begin_ASN1_ASP_TypeDef
$ASP_Id Private_Call_ALERT_IND
$PCO_Type PCO_Type
```

$ASN1_TypeDefinition

SEQUENCE {

  pRIVATE_CALL_ALERT_IND_SDU1 PRIVATE_CALL_ALERT_IND_SDU

}

$End_ASN1_TypeDefinition

$Comment /* */

$End_ASN1_ASP_TypeDef

$Begin_ASN1_ASP_TypeDef

$ASP_Id Private_Call_COMPLETE_IND

$PCO_Type PCO_Type

$ASN1_TypeDefinition

SEQUENCE {

  pRIVATE_CALL_COMPLETE_IND_SDU1 PRIVATE_CALL_COMPLETE_IND_SDU

}

$End_ASN1_TypeDefinition

$Comment /* */

$End_ASN1_ASP_TypeDef

$Begin_ASN1_ASP_TypeDef

$ASP_Id Private_Call_MAINTENANCE_IND

$PCO_Type PCO_Type

$ASN1_TypeDefinition

SEQUENCE {

  pRIVATE_CALL_MAINTENANCE_IND_SDU1 PRIVATE_CALL_MAINTENANCE_IND_SDU

}

$End_ASN1_TypeDefinition

$Comment /* */

$End_ASN1_ASP_TypeDef

$Begin_ASN1_ASP_TypeDef

$ASP_Id Private_Call_MAINTENANCE_CONF

$PCO_Type PCO_Type

$ASN1_TypeDefinition

```
SEQUENCE {

  pRIVATE_CALL_MAINTENANCE_CONF_SDU1  PRIVATE_CALL_MAINTENANCE_CONF_SDU

}

$End_ASN1_TypeDefinition

$Comment /*  */

$End_ASN1_ASP_TypeDef

$Begin_ASN1_ASP_TypeDef

$ASP_Id Private_Call_REMOTE_CLEARING_IND

$PCO_Type PCO_Type

$ASN1_TypeDefinition

SEQUENCE {

  pRIVATE_CALL_REMOTE_CLEARING_IND_SDU1
PRIVATE_CALL_REMOTE_CLEARING_IND_SDU

}

$End_ASN1_TypeDefinition

$Comment /*  */

$End_ASN1_ASP_TypeDef

$Begin_ASN1_ASP_TypeDef

$ASP_Id Private_Call_REMOTE_CLEARING_CONF

$PCO_Type PCO_Type

$ASN1_TypeDefinition

SEQUENCE {

  pRIVATE_CALL_REMOTE_CLEARING_CONF_SDU1
PRIVATE_CALL_REMOTE_CLEARING_CONF_SDU

}

$End_ASN1_TypeDefinition

$Comment /*  */

$End_ASN1_ASP_TypeDef

$Begin_ASN1_ASP_TypeDef

$ASP_Id Private_Call_DISCONNECT_IND

$PCO_Type PCO_Type

$ASN1_TypeDefinition
```

```
SEQUENCE {
  pRIVATE_CALL_DISCONNECT_IND_SDU1  PRIVATE_CALL_DISCONNECT_IND_SDU
}
$End_ASN1_TypeDefinition
$Comment /*  */
$End_ASN1_ASP_TypeDef
$Begin_ASN1_ASP_TypeDef
$ASP_Id Private_Call_DISCONNECT_CONF
$PCO_Type PCO_Type
$ASN1_TypeDefinition
SEQUENCE {
  pRIVATE_CALL_DISCONNECT_CONF_SDU1  PRIVATE_CALL_DISCONNECT_CONF_SDU
}
$End_ASN1_TypeDefinition
$Comment /*  */
$End_ASN1_ASP_TypeDef
$Begin_ASN1_ASP_TypeDef
$ASP_Id Private_Call_TRANSFER_IND
$PCO_Type PCO_Type
$ASN1_TypeDefinition
SEQUENCE {
  pRIVATE_CALL_TRANSFER_IND_SDU1  PRIVATE_CALL_TRANSFER_IND_SDU
}
$End_ASN1_TypeDefinition
$Comment /*  */
$End_ASN1_ASP_TypeDef
$Begin_ASN1_ASP_TypeDef
$ASP_Id Private_Call_TRANSFER_CONF
$PCO_Type PCO_Type
$ASN1_TypeDefinition
SEQUENCE {
```

```
  pRIVATE_CALL_TRANSFER_CONF_SDU1  PRIVATE_CALL_TRANSFER_CONF_SDU

}

$End_ASN1_TypeDefinition

$Comment /*  */

$End_ASN1_ASP_TypeDef

$End_ASN1_ASP_TypeDefs

$End_ASP_TypeDefs

$PDU_TypeDefs

$ASN1_PDU_TypeDefs

$Begin_ASN1_PDU_TypeDef

$PDU_Id SECsdu

$PCO_Type

$Comment /* security sdu  */

$ASN1_TypeDefinition


-- begin of SECsdu


EncryptionType ::= ENUMERATED

{

unknown (0),

clearSpeech (1),

networkEncryption (2),

endToEndEncryption (3),

isiEncryption (4)

}


KeyReference ::= OCTET STRING


KeyElement ::= SEQUENCE {

                address Address,

                keyOfCall OCTET STRING
```

```
        }

KeyList ::= SEQUENCE OF KeyElement

Security ::= SEQUENCE {
encryptionType          EncryptionType,
groupPartitioning       BOOLEAN OPTIONAL,
keyList                 [0] IMPLICIT KeyList OPTIONAL,
challenge               [1] IMPLICIT OCTET STRING OPTIONAL,
result                  [2] IMPLICIT OCTET STRING OPTIONAL,
keyReferenceSynchro     [3] IMPLICIT OCTET STRING OPTIONAL,
currentKeyReference     [4] IMPLICIT KeyReference OPTIONAL,
expiredKeyReference     [5] IMPLICIT KeyReference OPTIONAL,
setKeyModifier          [6] IMPLICIT OCTET STRING OPTIONAL,
useKeyModifier          [7] IMPLICIT OCTET STRING OPTIONAL,
other                   [8] IMPLICIT OCTET STRING OPTIONAL
}

SecCodop ::= CHOICE {
data            [1] IMPLICIT --$ implement size < -- OCTET STRING,
parameters      [2] IMPLICIT Security,
ackResult       [3] IMPLICIT AckResult -- confirmed transfer of previous SDU
}


SECsdu ::= SEQUENCE {
connectionId            ConnectionIdentifier,
invokeReference         InvokeReference,
codop                   SecCodop
}
```

```
-- end of SECsdu
```

```
$End_ASN1_TypeDefinition
$Comment /*  */
$End_ASN1_PDU_TypeDef
$Begin_ASN1_PDU_TypeDef
$PDU_Id CCsdu
$PCO_Type
$Comment /*  ISI call control sdu */
$ASN1_TypeDefinition
-- begin of CCsdu

Cause ::= ENUMERATED {
successfull (0),
inconsistentData (1),
reserved2 (2),
noAnswerTimeout (3),
reserved4 (4),
reserved5 (5),
voiceTransmissionAborted (6),
reserved7 (7),
hookOnWhileSetup (8),
voiceTransmissionDisturbance (9),
```

reserved10 (10),

remoteCallClearing (11),

maxNumberVoiceCall (12),

barredService (13),

serviceNotSupportedbySwMI(14),

intrusionOnWrongAddress (15),

reserved16 (16),

reserved17 (17),

reserved21 (21),

reserved22 (22),

reserved23 (23),

reserved24 (24),

voiceInactivity (26),

unreachableParty (41),

calledPartyBusy (42),

reserved43 (43),

callingPartyNotRegistered (44),

reserved45 (45),

inconsistentAddress (46),

reserved47 (47),

withdrawal (48),

preemption (49),

reserved50 (50),

noReply (51),

reserved52 (52),

failedTransfer (53),

reserved54 (54),

hopCreditLimit (61),

failedRouteing (62),

routeingLoop (63),

tooManyHopsForQoS (64),

reserved81 (81),

reserved82 (82),

reserved83 (83),

intrusionOnUnreachableTerminal (84),

intrusionOnCallNotEstablished (85),

unreachableTerminalAttribute (86),

noMorePartyInCall (87),

forwardingChainMax (88),

unknownCalledParty (89),

doubleTransfer (90),

nonCharacterizedTerminal (91),

transferProceeding (92),

listAddressNotExpandable (93),

twoSimultaneousIntrusions (94),

reserved101 (101),

reserved102 (102),

reserved103 (103),

reserved104 (104),

reserved105 (105),

reserved121 (121),

reserved122 (122),

reserved123 (123),

reserved141 (141),

groupAlreadyActive (142),

unknownGroupCommunication (143),

outOfCoverage (144),

reserved145 (145),

partialCoverage (146),

reserved147 (147),

intrusionOnNonSetupGroupCommunication (148),

reserved149 (149),

emergencyCallRelease (150),

wrongEstablishmentOg (151),

inconsistentGroupCommunication (152),

intrusionWithinCoverage (153),

expiredKeyReference (161),

externalPartyNoReply (181),

externalPartyBusy (182),

gatewayAddressInconsistent (183),

externalAddressInconsistent (184),

externalServiceBarredtoCallingParty (185),

externalServiceBarredtoCalledParty (186),

externalEquipmentUnreachable (187),

isdnUnreachable (188)

}

CallId ::= WORD

ConnectionIdentifier ::= --$ implement size < 5 --OCTET STRING

Cell ::= ConnectionIdentifier

RSW ::= ConnectionIdentifier

GRW ::= RSW

GLR ::= RSW

CC ::= OCTET STRING

NC ::= QUARTET

BN ::= SEQUENCE OF QUARTET

SAP ::= CHOICE {

```
grw [1] IMPLICIT GRW,
glr [2] IMPLICIT GLR
}

SwMI ::= SEQUENCE {
countryCode         CC,
networkCode         NC,
baseNetwork         [0] IMPLICIT BN OPTIONAL,
isiGSW              [1] IMPLICIT RSW OPTIONAL,
pisnNumber [2] IMPLICIT PISNPartyNumber OPTIONAL
}

AddressChoice ::= CHOICE
        {
        shortAddress        [1] IMPLICIT SEQUENCE OF QUARTET,
        functionalAddress   [4] IMPLICIT --$ implement size < -- OCTET STRING,
        longAddress         [5] IMPLICIT OCTET STRING(6),
        escapeCode          [7] IMPLICIT --$ implement size < -- OCTET STRING
}


Address ::= SEQUENCE
{
addresschoice       AddressChoice,
subaddress          [3] IMPLICIT -- interconnectAccess
        --$ implement size < -- OCTET STRING OPTIONAL,
countryCode             [8] IMPLICIT CC OPTIONAL, -- may be implicit if longAddress
networkCode         [9] IMPLICIT NC OPTIONAL, -- may be implicit if longAddress
baseNetwork         [10] IMPLICIT BN OPTIONAL -- is usually implicit
}
```

```
GroupChoice ::= CHOICE{
federalGroup        [0] IMPLICIT INTEGER(0..4096),              -- 12 bits
networkGroup        [1] IMPLICIT OCTET STRING(SIZE(6)),     -- 48 bits
group               [2] IMPLICIT INTEGER(0..4096),              -- 12 bits
allGroups     [3] IMPLICIT NULL
}


Group ::= SEQUENCE{
groupChoice         GroupChoice,
countryCode         [8] IMPLICIT CC OPTIONAL,
networkCode         [9] IMPLICIT NC OPTIONAL,
baseNetwork         [10] IMPLICIT BN OPTIONAL
}

CallPriority ::= ENUMERATED { -- relative priority of calls for a user
unknown (0),
routine (2),
urgent (4),
flash (6),
broadcast (8),
emergency (12)
}

SetupPriority ::= CallPriority -- priority of allocating a bearer connection, i.e. virtual resource reservation
      -- default value is routine for quasi-transmission trunked establishment

UserPriority ::= ENUMERATED { -- user rank, delta to SetupPriority or to ActivationPriority
minUserPriority (0),
mediumUserPriority (1),
upperUserPriority(2),
```

```
unused (15)
}

RetentionPriority ::= ENUMERATED {
-- maximum retention priority for bearer connection,
-- i.e. retention priority for depth zero of the associated coverage
unknown (0),
vulnerable (1),
normal (2),
resistant (3)
}

TrunkingPriority ::= OCTET STRING -- SwMI priority allocated at establishment
                                  -- for quasi-transmission trunking
-- used in association with ActivationPriority
--      for actual seizing a physical resource for a bearer connection


Hook ::= ENUMERATED {
automaticHookOffWithoutTone (0),
automaticHookOffWithTone (1),
ring (2),
reserved (3)
}

Routeing ::= SEQUENCE{
originatingSwmi             SwMI,
callMaster                  SwMI,
destinationSwmi             SwMI,
routeingHopCredit           Counter256,
forwardSwitchingForbidden   [0] IMPLICIT NULL OPTIONAL
```

```
}



EndEstablishmentSynchro ::= ENUMERATED {
nil (0),
totalCoverage (1),
partialCoverage (2),
allCalledPartyConnected (3),
oneCalledPartyConnected (4),
allCalledPartyAlerted (5)
}

SpeechService ::= ENUMERATED {
pcmAlaw (0),
tetrapol (1),
tetra (2),
gsmFullRate (3)
}

QuasiTransmissionTrunking ::= SEQUENCE{
deactivationHangTime    Tempo, -- hang time after end of transmission
trunkingPriority        TrunkingPriority
}

MessageTrunking ::= NULL
-- deactivation hangTime=maxCallDuration, i.e. no deactivation until call release
-- maximum TrunkingPriority: a resource allocated for MessageTrunking can
-- not be preempted for a QuasiTransmissionTrunking usage.

Trunking ::= CHOICE{
message [0] IMPLICIT MessageTrunking,
```

```
transmission [1] IMPLICIT QuasiTransmissionTrunking
}

CallQueued ::= NULL

Cad ::= ENUMERATED {
unknownCad (0),
incomingCallToBeAuthorizedByDispatcher (1),
outgoingCallAuthorizedByDispatcher(2),
directDiallingInRequested (3)
}

ServiceAction ::= ENUMERATED {
unknown                 (0), -- reserved
setup                   (1), -- Half duplex call/moch setup with ISISetup
                             -- Coverage bearer setup with ISISetup
                             -- End of setup coverage/moch with ISIDisconnect
release                 (2), -- Call release with ISIDisconnect
                             -- Coverage/moch release with ISISetup
                             -- Remote call clearing with ISISetup
intrusion               (3), -- intrusion into a private call
lateEntryIncludeCall    (4), -- participation, SwMI insertion in call/coverage
withdrawalFromCall      (5), -- SwMI withdrawal from on-going call
withdrawalFromCoverage  (6), -- SwMI withdrawal from Coverage bearer
priorityMode            (7), -- used for scanning on R6 interface only
sequentialMode          (8), -- used for scanning on R6 interface only
resume                  (9), -- used for scanning on R6 interface only
suspend                 (10), -- used for scanning on R6 interface only
reinit                  (11), -- access gate bearer reset on R6 interface only
ambience                (12), -- ambience listening for individual call only
isiFallbackSenderInitiative (13),
```

```
isiFallbackRecipientInitiative     (14)

}


-----------------------------------------------------------------------------------------------------------------------

-- TETRAPOL service types


-- Private communications


PrivateCall ::= SEQUENCE {

callingAddress      Address,

calledAddresses     SET OF Address

}


IndividualCall ::= PrivateCall              -- 1 calledAddress


MultipartyCall ::= PrivateCall              -- up to 4 calledAddresses


-- Group communications


Coverage ::= SEQUENCE { -- defined from OMC

identifier          WORD, -- global reference allocated by initiating SwMI

trunkingDepth       [0] IMPLICIT OCTET STRING OPTIONAL,-- according to trunkingScheme

masterSwMI          [1] IMPLICIT SwMI OPTIONAL,

participantSwMI     [2] IMPLICIT SEQUENCE OF SwMI OPTIONAL

}


GroupCommunication ::= SEQUENCE {

coverage                    Coverage,

participationGroup          [1] IMPLICIT SET OF Group OPTIONAL

}
```

```
-- multisite open open class of services

Moch ::= SEQUENCE{
communication GroupCommunication,
-- coverage identifier is known by user
-- up to 10 participationGroup, provided at establishment
establishmentGroup Group
-- group of users who can establish the moch over the coverage
}
BroadcastMoch ::= Moch -- broadcast SetupPriority and specific air interface service to users

Merging ::= Moch -- selected according to its coverage with air interface service for a set of talkgroups

-- group call class of services
Talkgroup ::= GroupCommunication
-- group over a coverage known by network
-- 1 participationGroup, known by user

GroupCall ::= SEQUENCE {
callingAddress      Address, -- user who requests to establish the group call
calledGroup         Talkgroup
}

-- Broadcast call class of services

BroadcastCall ::= CHOICE{
broadcastmoch [0] IMPLICIT BroadcastMoch,
broadcastgroupcall [1] IMPLICIT NULL
}

-- Emergency call class of services
```

```
Ech ::= SEQUENCE {
moch          Moch, -- emergency SetupPriority and specific air interface service to users
sosAddress  [0] IMPLICIT Address OPTIONAL
}

EmergencyCall ::= CHOICE{
emergencyopenchannel [0] IMPLICIT Ech,
emergencygroupcall [1] IMPLICIT NULL
}

-- Terminal scanning service

GroupComms ::= CHOICE{
openchannel Moch,
broadcast BroadcastCall,
merging Merging,
talkgroup Talkgroup,
groupcall GroupCall
}

Scan ::= SEQUENCE OF GroupComms

--

ServiceType ::= CHOICE {
pc [0] IMPLICIT PrivateCall,
moch [1] IMPLICIT Moch,
ech [2] IMPLICIT EmergencyCall,       -- Not used on ISI
bc [3] IMPLICIT BroadcastCall,
tkg [4] IMPLICIT Talkgroup,
```

```
gc [5] IMPLICIT GroupCall,
scan [6] IMPLICIT Scan,                -- Not used on ISI
merging [7] IMPLICIT Merging,          -- Not used on ISI
cov [8] IMPLICIT Coverage
}

-------------------------------------------------------------------------------------------------------------------------------

-- TETRAPOL call control SDU

ISISetup ::= SEQUENCE {
serviceType              ServiceType,
serviceAction            [9] IMPLICIT ServiceAction DEFAULT setup,
setupPriority            [10] IMPLICIT SetupPriority DEFAULT routine,
requestingUserPriority   [11] IMPLICIT UserPriority DEFAULT minUserPriority,
dispatcherAuthorization  [12] IMPLICIT Cad DEFAULT unknownCad,
setupTimeout             [13] IMPLICIT Tempo DEFAULT 0,
retentionPriority        [14] IMPLICIT RetentionPriority DEFAULT normal,
requestedSpeechService   [15] IMPLICIT SpeechService DEFAULT tetrapol,
hook                     [16] IMPLICIT Hook DEFAULT ring,
routeing                 [17] IMPLICIT Routeing OPTIONAL,
trunkingScheme           [18] IMPLICIT Trunking OPTIONAL,
requestedSecurity        [19] IMPLICIT Security OPTIONAL
}

ISIProceeding ::= NULL -- destinationSwMI-to-originatingSwMI message

ISIAlerting ::= SEQUENCE {
callQueued               [1] IMPLICIT CallQueued OPTIONAL,
proposedSecurity         [2] IMPLICIT Security OPTIONAL,
proposedSpeechService    [3] IMPLICIT SpeechService OPTIONAL
}
```

```
ISIConnect ::= SEQUENCE {
requestedSynchro        EndEstablishmentSynchro,
callId                  [1] IMPLICIT CallId OPTIONAL,
-- mandatory callId upon groupCall setup
-- if callingAddress is not a member of calledGroup
actualDestinationSwMI   [2] IMPLICIT SwMI OPTIONAL,
actualMasterSwMI        [3] IMPLICIT SwMI OPTIONAL,
usedSecurity            [4] IMPLICIT Security OPTIONAL,
usedSpeechService       [5] IMPLICIT SpeechService OPTIONAL
}

ISIConnectAck ::= SEQUENCE {
                usedSynchro             EndEstablishmentSynchro,
usedSecurity            [1] IMPLICIT Security OPTIONAL,
participationGroup              [2] IMPLICIT --$ implement size < 11--
                                        SET OF Group OPTIONAL
}

ISIDisconnect ::= SEQUENCE {
disconnectAction                ServiceAction,
cause                           [1] IMPLICIT Cause OPTIONAL,
transferFromAddress             [2] IMPLICIT Address OPTIONAL,
transferfToAddress              [3] IMPLICIT Address OPTIONAL,
visitedSwMI                     [4] IMPLICIT SwMI OPTIONAL
}

ISIRelease ::= NULL

ISIMaintenance ::= SEQUENCE {
updateSecurity          [1] IMPLICIT Security OPTIONAL,
```

```
suspend                 [5] IMPLICIT NULL, -- scanning
resume                  [6] IMPLICIT NULL, -- scanning
cause                   [7] IMPLICIT Cause OPTIONAL,
transferFromAddress     [8] IMPLICIT Address OPTIONAL,
transferfToAddress      [9] IMPLICIT Address OPTIONAL,
visitedSwMI             [10] IMPLICIT SwMI OPTIONAL, -- when transferred
ackResult               [11] IMPLICIT AckResult --confirmed transfer of previous SDU
}

Cccodopchoice ::= CHOICE {
isiSetup                [11] IMPLICIT ISISetup,
isiProceeding           [12] IMPLICIT ISIProceeding,
isiAlerting             [13] IMPLICIT ISIAlerting,
isiConnect              [14] IMPLICIT ISIConnect,
isiConnectAck           [15] IMPLICIT ISIConnectAck,
isiDisconnect           [16] IMPLICIT ISIDisconnect,
isiReleased             [17] IMPLICIT ISIRelease,
isiMaintenance          [18] IMPLICIT ISIMaintenance
}

CCsdu ::= SEQUENCE {
connectionId    ConnectionIdentifier,
codop           Cccodopchoice
}

-- end of CCsdu



$End_ASN1_TypeDefinition
$Comment /*  */
$End_ASN1_PDU_TypeDef
```

```
$Begin_ASN1_PDU_TypeDef

$PDU_Id Mmsdu

$PCO_Type

$Comment /* mobility management over ISI */

$ASN1_TypeDefinition

-- begin of MMsdu


TerminalState ::= ENUMERATED {

trafficDisabled (1),

forwarded (2),

migrating (16),

powerSaving (32)

}


TerminalType ::= ENUMERATED {

reservedType1 (1),

reservedType2 (2),

reservedType3 (3),

fixedRadioTerminal (4),

mobileRadioTerminal (5),

lineConnectedTerminal (6),

reservedType7 (7),

reservedType8 (8),

reservedType9 (9),

fixedRadioPABXaccess (10),

mobileRadioPABXaccess (11),

lineConnectedPABXaccess (12),

reservedType13 (13),

reservedType14 (14),

reservedType15 (15),

fixedRadioAccessGate (16),
```

```
mobileRadioAccessGate (17),
lineConnectedAccessGate (18)
}

UdtType ::= ENUMERATED {
minitel (1),
stutel (2),
ndis (4),
other (8),
unknown (255)
}

SerialNumber ::=   OCTET STRING(4)


ISILocationUpdateDemand ::= SEQUENCE {
address                 Address,
serialNumber            SerialNumber,
visitedSwMI                     SwMI,
expectedTerminalState   TerminalState
}

RegistrationResult ::= ENUMERATED {
successfullyRegistered (0),
unknownTerminal(1),
serialNumberInconsistentWithAddress (2),
networkNoAccess (3),
terminalAccessDisabled (5),
terminalTrafficDisabled (6),
reserved7 (7),
outOfOrderTerminal (8),
```

vlrSoftwareSaturation (9),

hlrSoftwareSaturation (10),

failedAuthentication (11),

terminalLocalRegistration (12),

olderRegistration (13),

networkCongestion (14),

databaseAccessError (15),

illegalTerminal (16),

databaseReadOnly (17),

vlrSaturation (18),

authenticationKo (50),

migrationNotAllowed (80),

limitedServiceUponMigration (81)

}

StateBitmap ::= BIT STRING

StateBitMask ::= ENUMERATED {

expectingNetworkGroupDownloading (1),

expectingLocalGroupDownloading(2),

homeBnRegistered (16),

locallyRegistered (256),

working (512),

characterized (1024),

reserved2048 (2048),

forwarded (4096),

present (8192),

reserved1638 (16384),

accessDisabled (65536),

trafficDisabled (131072),

tracked (524288),

```
udtEquipped (1048576),
reserved2097152 (2097152),
outOfOrder (8388608)
}

ISILocationUpdateProfile ::= SEQUENCE {
registrationResult RegistrationResult,
address Address,
serialNumber SerialNumber,
terminalType TerminalType,
stateBitmap StateBitmap,
udtType UdtType,
numberOfMigration Counter256,
numberOfFullRegistration Counter256,
homeSwMI [1] IMPLICIT SwMI OPTIONAL,
visitedSwMI [2] IMPLICIT SwMI OPTIONAL,
forwardAddressOnNoReply [3] IMPLICIT Address OPTIONAL,
unconditionnalForwardAddress [4] IMPLICIT Address OPTIONAL,
nominalOperationalGroup Group
}

ISILocationDelete ::= SEQUENCE {
address             Address,
serialNumber        SerialNumber,
numberOfMigration Counter256
}

Scope ::= ENUMERATED {
world (0),
tetrapolDomain (1),
```

```
swmi(2),
baseNetwork (3)
}

ISIEnquiry ::= SEQUENCE {
address Address,
locationScope       [0] IMPLICIT Scope OPTIONAL,
keyReference        [1] IMPLICIT KeyReference OPTIONAL,
requestedSetOfServiceProfile [2] IMPLICIT --$ implement size < 200--
                                        OCTET STRING -- reserved
}

EnquiryResult ::= ENUMERATED {
knownLocalised (0),
vlrNotaccessible (1),
unknownFromHlr (2),
knownNotLocalised (3),
unknownFromVlr (5),
knownLocalisedFromVlrButCellInaccessible (19),
remoteAuthenticationFailed (20),
encryptionFailure (23),
keyReferenceUnknown (24),
randomUnavailable (25)
}

BasicProfile ::= SEQUENCE {
numberOfMigration           Counter256,
numberOfFullRegistration    Counter256,
terminalType                [1] IMPLICIT TerminalType,
stateBitmap                 [2] IMPLICIT StateBitmap,
udtType                     [3] IMPLICIT UdtType OPTIONAL,
```

```
forwardAddressOnNoReply      [4] IMPLICIT Address OPTIONAL,
unconditionnalForwardAddress [5] IMPLICIT Address OPTIONAL,
validityOfKey                [6] IMPLICIT BOOLEAN OPTIONAL,
keyOfCall                    [7] IMPLICIT --$ implement size < --
                                 OCTET STRING OPTIONAL
}

DefaultProfile ::= BasicProfile

Profile ::= CHOICE{
profile [0] IMPLICIT BasicProfile,
bitmap [1] IMPLICIT --$ implement size < --  OCTET STRING -- reserved
} -- DEFAULT DefaultProfile

ISILocationProfile ::= SEQUENCE {
address Address,
enquiryResult EnquiryResult,
visitedSwMI                  [0] IMPLICIT SwMI OPTIONAL,
forwardAddress               [1] IMPLICIT Address OPTIONAL,
setOfEnquiredServiceProfile  [2] IMPLICIT --$ implement size < 200 --
                             SET OF Profile OPTIONAL
}

DisablingType ::= ENUMERATED {
enabled(0),
trafficDisable(1),
accessDisable(2)
}

ISIDisable ::= SEQUENCE {
address                      Address,
```

```
serialNumber              SerialNumber,
expectedDisablingType     [0] IMPLICIT DisablingType OPTIONAL,
actualDisableType         [1] IMPLICIT DisablingType OPTIONAL
}


AckResult ::= ENUMERATED {
confirmedOperation (0),  -- acknowledgement of a successfull SDU transfer
flushInvokeReference (1),        -- flush all buffered SDU with same invokeReference
invokeReferenceFlushed (2),
invokeReferenceNotFound (3),
duplicatedRequest (4)
}

ISIAcknowledge ::= AckResult   -- used over an unreliable bearer service

Mmcodop ::= CHOICE {
isiLocUpdateDemand     [31] IMPLICIT ISILocationUpdateDemand,
                       -- from visited to home SwMI at migration
isiLocUpdateProfile    [32] IMPLICIT ISILocationUpdateProfile,
                       -- from home to visited SwMI at migration
isiLocDelete           [33] IMPLICIT ISILocationDelete,
                       -- from home to previous visited SwMI
                       -- to home SwMI upon hlr recovery
isiEnquiry             [34] IMPLICIT ISIEnquiry,
                       -- location request before routeing
                       -- service profile interrogation
isiLocProfile          [35] IMPLICIT ISILocationProfile,
                       -- from home SwMI as a location indication
                       -- from home SwMI as a service profile delivery
isiDisable             [36] IMPLICIT ISIDisable,
```

```
                        -- terminal disable order
isiAcknowledgment       [37] IMPLICIT ISIAcknowledge
}

MMsdu ::= SEQUENCE {
endConnectionId         ConnectionIdentifier,
invokeReference         InvokeReference,
codop Mmcodop
}

-- end of MMsdu

$End_ASN1_TypeDefinition
$Comment /*  */
$End_ASN1_PDU_TypeDef
$Begin_ASN1_PDU_TypeDef
$PDU_Id UserInfo
$PCO_Type
$ASN1_TypeDefinition
GroupChoice ::= CHOICE{
federalGroup        [0] IMPLICIT INTEGER(0..4096),          -- 12 bits
networkGroup        [1] IMPLICIT OCTET STRING(SIZE(6)),     -- 48 bits
group               [2] IMPLICIT INTEGER(0..4096),          -- 12 bits
allGroups     [3] IMPLICIT NULL
}
$End_ASN1_TypeDefinition
$Comment /*  */
$End_ASN1_PDU_TypeDef
$End_ASN1_PDU_TypeDefs
$Begin_ASN1_PDU_TypeDefsByRef
$ASN1_PDU_TypeDefByRef
```

$PDU_Id

$PCO_Type

$ASN1_TypeReference

$ASN1_ModuleId

$End_ASN1_PDU_TypeDefByRef

$Comment /* */

$End_ASN1_PDU_TypeDefsByRef

$End_PDU_TypeDefs

$End_DeclarationsPart

$ConstraintsPart

$End_ConstraintsPart

$DynamicPart

$DefaultsLibrary

$Begin_Default

$DefaultId OtherwiseFail

$DefaultRef qsig_anf_sdt/

$Objective /* */

$BehaviourDescription

$BehaviourLine

$LabelId

$Line [0] QSIG1_DataLink ?OTHERWISE

$Cref

$VerdictId FAIL

$End_BehaviourLine

$BehaviourLine

$LabelId

$Line [0] QSIG2_DataLink ?OTHERWISE

$Cref

$VerdictId FAIL

$End_BehaviourLine

$BehaviourLine

$LabelId

$Line [0] ANF_Link2 ?OTHERWISE

$Cref

$VerdictId FAIL

$End_BehaviourLine

$BehaviourLine

$LabelId

$Line [0] ANF_Link1 ?OTHERWISE

$Cref

$VerdictId FAIL

$End_BehaviourLine

$BehaviourLine

$LabelId

$Line [0] ?TIMEOUT

$Cref

$VerdictId INCONC

$End_BehaviourLine

$End_BehaviourDescription

$Comment /* */

$End_Default

$End_DefaultsLibrary

$End_DynamicPart

$End_Suite

History

Document history		
Date	**Status**	**Comment**
30/09/96	Version 0.0.1	SDL model, ASN.1 data and TTCN test suite structure First Version