EUROCONTROL

EACI PROGRAMME

ATN End Systems

(FCO.ET3.ST04)

DRAFT Validation Report For CNS/ATM-1 Package SARPs

Abstract

This document describes the Eurocontrol activities towards the validation of the SARPs for CNS/ATM-1, and the results that have been obtained.

File Ref :	TC5/DEL/T03
File Id:	D21v0_e.doc
Version No:	0.E
Date of Issue:	18 September 1996

COPYRIGHT NOTICE

APPROVAL

Authority	Date	Signature
Name Title		
Name Title		
Name Title		

DOCUMENT CONTROL LOG

Section(s)	Date	Amendment Number	Reason for change and/or Change Note Number.
All	13.08.96	0.A	Draft (a) was provided to Steve Van Trees and DVR as an informal 'snapshot' of work in progress, to facilitate discussions between the Agency and the FAA.
Most	16.08.96	0.B	More complete "first draft" with specific material relevant to ADS added.
Most	27.08.96	0.C	To incorporate feedback from Tony Kerr on version 0.B. Queries are marked <<…>>.
Most	9.09.96	0.D	To resolve comments from AJK and DVR, and to add API definition as a validation activity. Also removal of detailed defect report lists. Geode Results added. "Higher Level Requirements" (1.5) and results added.
Most	18.09.96	0.E	Revised after review within Eurocontrol, to remove historical info., and match reported results to actual achieved validation activities.

Version	Originating Company	Author	Contract	Task
0.A	Level-7 Ltd.	I R Valentine	TC5	T03
0.B	Level-7 Ltd.	I R Valentine	TC5	T03
0.C	Level-7 Ltd	I R Valentine	TC5	T03
0.D	Level-7 Ltd	I R Valentine & A J Kerr	TC5	T03
0.E	Level-7 Ltd	I R Valentine	TC5	T03

DISTRIBUTION LIST

Copy Number	Document Holder	Location
1	Ian Valentine	Eurocontrol / Level-7 Bracknell
2	Danny Van Roosbroek	Eurocontrol Headquarters
3	Steve Van Trees	FAA / STEL
4	Ron Jones	FAA
5	Tony Kerr	Eurocontrol / Level-7 Bracknell
6	Tim Maude	Eurocontrol / Level-7 Bracknell
7	Jane Hamelink	FAA / Adsystech
8		
9		
10		
11		
12		
13		
14		
15		
16		

TABLE OF CONTENTS

1. VALIDATION APPROACH AND STRATEGY

1.1 Scope

This document records the Eurocontrol validation activities, and current status related to the Standards and Recommended Practices (SARPs) and Guidance material (GM) for Air-Ground datalink applications and supporting ATN upper layer communications service, as developed by ATNP WG3 subgroups 2 and 3, respectively. The document does not address any validation activities relating to the ATN-internet, or Ground-Ground applications.

1.2 Structure of the Document

This document describes both the validation activities undertaken by Eurocontrol, and the results achieved. It relates these validation activities to the Validation Objectives (VOs) defined by ATNP WG3.

Section 2 re-iterates the High Level VOs identified by ATNP WG3, and adds the Eurocontrol interpretation of these in the validation activities.

Section 3 cross-references the Eurocontrol validation activities to the High Level VOs.

Section 4 describes the validation activities undertaken by Eurocontrol

Section 5 describes the validation results for ADS SARPs and GM

Sections 6, 7, 8, and 9 do similarly for CM, CPDLC, FIS and Upper Layers respectively

1.3 Background

The Agency began work on validation of the CNS/ATM-1 Package SARPs in October 1995, after they began to be baselined at the 4th (Banff) meeting of ATNP WG3. The following validation activities have been undertaken or are planned:

- Inspection and Desk Checking
- Requirements Analysis
- Application Programming Interface (API) Specification
- Simulation and Modelling
- Prototype Implementation
- Interworking Testing

1.4 Objectives

The basic objective of the validation activities is to determine the requirements which are expressed in the draft SARPs and to track these throughout the validation phase to be able to demonstrate which requirements have been satisfactorily validated.

It is also important to evaluate the extent to which the generic validation objectives (VOs) have been met by different types of validation activity. Section 1.6 lists the high level validation objectives, and section 3 illustrates how the validation activities undertaken by the Agency relate to these high level validation objectives.

It must be possible to generate reports to show at any given time which requirements have been validated (i.e. successfully tested) and which VOs have been achieved.

1.5 Approach

The Agency took the view that validation activities should

a) identify defects in the draft SARPs and cause defect reports to be raised to remedy deficiencies, and

b) ensure that the material (particularly the “shall” and “should” statements), is correct and in accordance with the overall purpose of the SARPs.

The defect reports are easy to identify and trace, as each is recorded with a unique number. The disposition of the defect report is recorded in the change records associated with successive versions of the draft SARPs.

Ensuring that the draft SARPs achieve their overall purpose is more difficult, and the approach taken has been to identify the major functionality to be provided by the SARPs, then report on the degree to which this functionality appears to have been successfully provided, as determined through the various validation activities.

The major functionality description for each SARPs area is based on a number of factors:

- logical groupings, such as “all the requirements relating to logon”
- groups convenient for testing, such as the protocol definition and service primitives
- defined subsetting rules within the SARPs themselves.

It became clear that the first two of these groupings do not necessarily need universal agreement, but organisations carrying out validation activities need to select groupings that are convenient for their validation activities. At the end of the validation period, because the mapping of individual SARPs requirements to groupings by different organisations is known, the composite results of many different validation activities can be determined.

1.6 High-Level Requirements

The draft SARPs documents have been constructed such that related requirements at the lowest level are already grouped together in a single “shall” clause. Examples are:

- “shall” clauses which contain lists of actions to be performed in response to a given stimulus when in a given state, usually expressed as multi-level numbered lists.
- “shall” clauses which refer to Profile Requirements List (PRL) tables, where requirements are specified by coded table entries.

In some cases, it is meaningful to divide the SARPs documents into areas of still higher-level functionality which can be validated as a single entity (e.g. “Demand Contract air support” in the ADS SARPs, “Naming and Addressing” in the UL SARPs). In the case of the Air-Ground SARPs, such major areas of functionality are identified in the chapters on Subsetting Requirements.

The identified requirements can then (optionally) be grouped into “Higher-level Requirements”. There then needs to be full traceability between the basic low-level requirements and the higher-level requirements.

Requirements and/or groups of requirements, can then be assigned Validation Means (e.g. visual inspection, protocol simulation, etc.).

Validation testing can then proceed. This can be undertaken by a number of different agencies, whose results may not always be consistent. Thus, for each requirement, a record must be maintained of the validation means, tests performed and results of the testing. These results must be traceable to a particular test run.

2. HIGH LEVEL VOS (FROM ATNP WG3)

The “Validation Objectives” (VOs) listed below were finally agreed by ATNP WG3 at its 7th meeting (Munich, June 1996). These VOs relate to high level generic objectives of the SARPs validation activities, and not to any particular functionality decomposition within individual SARPs for particular validation activities. Three types of VOs are distinguished:

- *System Level Validation Objectives* (SVOs) relate to the system level requirements which are based on operational requirements within the *ICAO Draft Manual of ATS Data Link Applications* or elsewhere.
- *Functional Validation Objectives* (FVOs) relate to the functional characteristics described in the SARPs.
- *Technical Validation Objectives* (TVOs) relate to the technical content of the SARPs.

Table 2-1: Validation Objectives

VO	Description
SVO 1	To determine which System Level Requirements are satisfied by the functional descriptions in combination with the user requirements and recommended practices of the SARPs. *Note: SVO 1 relates to the ICAO Draft Manual of ATS Data Link Applications. It is not part of the VO to check that the ATNP SARPs actually reflect the requirements set down in that manual.*
SVO 2	To determine if the CNS/ATM-1 Package applications specifications are mutually consistent.
FVO 1	To determine if the functional descriptions in the SARPs are compatible with the technical requirements. *Note: “Functional Descriptions” are an explicit section within Air-Ground SARPs, but are not necessarily present in other SARPs*.
FVO 1E	*Eurocontrol interpretation - To determine if the technical specification in the SARPs satisfies any functional descriptions.*
FVO 2	To determine if the user requirements and recommended practices are compatible with the technical requirements. *Note: “User Requirements” have a section in the Air-Ground SARPs, but are not necessarily present in other SARPs*.
FVO 2E	*Eurocontrol interpretation - To determine if the technical specification in the SARPs is in accordance with the specified User Requirements and Recommended Practices.*
FVO 3	To determine if the SARPs are complete.
FVO 4	To determine if the SARPs are unambiguous.
FVO 5	To determine if the SARPs are consistent.

VO	Description
FVO 6	To determine if there are requirements in the SARPs which would have no effect if removed. *Note: This VO should be interpreted to mean that there are no requirements in the SARPs that are not necessary for CNS/ATM-1 package functionality, or to achieve migration to future CNS/ATM functionality. It is not meant to eliminate possible duplicated statements of requirement that are known to exist.*
FVO 7	To determine if provision has been made to ensure that the SARPs are implementation independent.
TVO 1	To determine if the protocol description supports the specified end to end services.
TVO 2	To determine if the protocol description has any unacceptable behaviour
TVO 3	To determine if the abstract service interface parameters are mapped appropriately to PDU fields and/or communication service interface parameters, and vice versa.
TVO 4	To determine if bad protocol generated by the peer application entity is correctly handled.
TVO 5	To determine if the application SARPs are consistent with the upper layer architecture to the extent that this is a requirement, e.g. use of the Dialogue service, application of the control function.
TVO 6	To determine if the APDUs are correctly specified.
TVO 7	To determine if provision for QOS management has been addressed.
TVO 8	To determine if provision for future migration has been addressed.
TVO 9	To determine if efficiency requirements have been addressed, e.g. minimising size of data transfer, appropriate maintenance of dialogue.
TVO 10	To determine that the functionality described in the SARPs is implementable.
TVO 11	To determine that independent implementations built in accordance with the SARPs will be able to interoperate.

Figure 1 shows how the elements of the SARPs relate, and where some of the “relational” validation objectives fit.

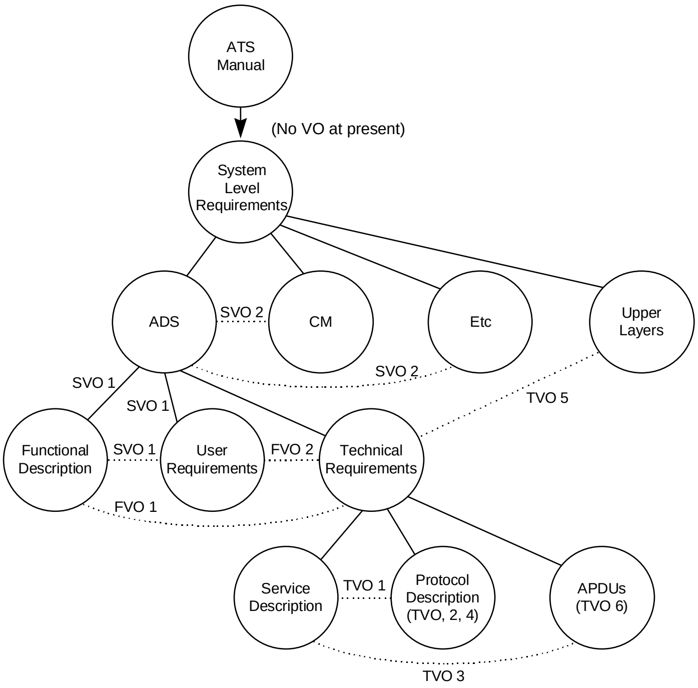

Figure 1: Relationship of SARPs and VOs

3. RELATIONSHIP BETWEEN EUROCONTROL VALIDATION ACTIVITIES AND ICAO ATNP WG3 VALIDATION OBJECTIVES

The following table relates the validation means as identified by ATNP WG3 to the Eurocontrol validation activities.

Table 3-1

Generic Validation Means	Eurocontrol validation activity
Inspection and Analysis	Inspection and desk checking
	Requirements analysis
	API Specification
Simulation	Simulation and modelling
Formal modelling	
Prototype implementation	TES Prototype Implementation
Interworking	planned using above prototypes
Target environment testing	planned
Engineering judgement	Inspection and desk checking

The validation activities are considered in detail in section 4.

The VOs agreed by ATNP WG3, as summarised in section 2, are addressed by the Eurocontrol validation activities as shown below.

Table 3-2

VO	Inspection and Desk Checking	Requirements Analysis	API Specification	Simulation and Modelling	Prototype Implement-ation)	Interworking Testing
SVO 1	√					
SVO 2	√	√	√		√	
FVO 1	√					
FVO 2	√		√			
FVO 3				√	√	√
FVO 4	√		√		√	√
FVO 5	√		√	√	√	
FVO 6		√				
FVO 7	√		√		√	
TVO 1	√			√	√	√
TVO 2				√	√	√
TVO 3	√				√	

TVO 4				√	√	√
TVO 5	√				√	
TVO 6	√				√	√
TVO 7	√		√			√
TVO 8	√					
TVO 9	√					
TVO 10					√	
TVO 11						√

Note: Other intersections are theoretically possible in the above table; the intersections shown are what has actually been done (or is in process of being done) rather than what could be done.

4. EUROCONTROL VALIDATION ACTIVITIES

4.1 Inspection and Desk Checking

The draft SARPs were reviewed at various stages of their development, and in various degrees of thoroughness.

As Eurocontrol has provided the editors for both the ADS and Upper Layers SARPs, these documents have obviously been reviewed in much greater depth than the other application SARPs.

4.2 Requirements Analysis

4.2.1 General

A set of tools was developed to support the analysis of "requirements" in the SARPs documents. These include:

- Macros for extracting requirements from SARPs documents
- Requirements database
- Requirements / validation tables.

The tools were originally developed to assist in tracking requirements and validation results. They include additional features to support requirements analysis.

The SARPs documents contain a series of requirements and recommendations, each expressed using the word ‘shall’ or ‘should’. The series of such statements from a given SARPs can be imported into a requirements database.

The editors of SARPs do not necessarily provide a unique reference for each requirement contained in the SARPs. Thus, the SARPs document is subject to some processing so that each requirement can be identified and then loaded into the database.

Processing the SARPs document involves numbering the ‘shall’ and ‘should’ statements in the document, extracting the numbered statements and then importing the extracted data into a new database, Word document, Excel spreadsheet, etc.

4.2.2 Use for SARPs Validation

The requirements analysis and extraction processes reveal:

- layout / structural errors in the SARPs document
- spurious “shalls” and “shoulds”
- requirements dependent on some condition
- duplicated requirements
- redundant requirements
- badly formed requirements

4.2.3 SARPs Processing Tools

To support the requirements tracing, some tools in the form of macros for Microsoft Word for Windows V6.0a were produced. The macros are used to generate requirement numbers which form a unique reference for each shall/should statement in the source document (SARPs).

The requirements identifier macro was used to automate the process of assigning identifiers to requirements statements. The identifier numbers that the macro generates are stored as hidden text in the SARPs document. The requirement identification number value is based on a number of factors. If identification numbers have not been previously allocated to the SARPs document then the value will be based on the 'initial' number given by the user.

The tools also provide the following functions:

- Checking the SARPs document for sentences that contain more than one requirement. This allows the identification of multiple requirements in sentences (which is considered bad editorial practice); it does not make any changes to the document.
- Detection of strings such as "i.e.", which would present problems in automatically determining the end of sentences, and replacing quotation marks with Word 'smart quotes'. This modifies the document to ensure that requirement statements do not get truncated prematurely and that they can be imported smoothly into the database.
- Allowing user-defined initial identifier values. This allows the requirement numbering process to take account of SARPs which are split into multiple separate documents.

Following the identification and labelling of requirements, a separate tool was used to extract the requirements and any supporting text into a temporary table suitable for import into a database application.

The tools used to process the SARPs documents consist of the following Word macros.

- Numtags1

This is a Word macro for adding a numeric tag to occurrences of the string #### in SARPs documents. This allows "should" statements as well as "shall" statements to be tagged in a single pass of a suitably pre-processed SARPs document. The added tags have the format "([xx])", and are in hidden text.

- NumHeadings3

This is a Word macro which adds "hard" heading and bullet list numbers into SARPs documents where automatic paragraph numbering based on Word Heading levels has been used. This allows the requirements extraction macros to determine the paragraph number of the requirements text, and prevents the format of numbered lists being lost when the document styles are changed.

4.2.4 Requirements Database

A requirements database (RDB) is a means to establish a reference point to the requirements defined in the text of a SARPs.

The RDB was populated with low-level requirements from the post-Brisbane versions of the SARPs, as a means both of picking up initial inconsistencies, and of tracking validation exercises.

The RDB has proved valuable in analysing the style of the draft SARPs addressed and finding typographical errors and inconsistencies in style, as well as instances of bad practice, such as a single clause containing two "shalls".

4.2.5 Requirements / Validation Tables

A table-based approach was pursued for the air-ground SARPs, rather than the more complex record-oriented views supported by the Eurocontrol Requirements Database. For simplicity, lists of requirements and recommendations were produced as Word tables, which could then be imported into other applications as required for spreadsheet or database analysis.

Low-Level Requirements table, containing fields:

Reference number

Requirement identifier

SARPs paragraph number

Requirement text and related information

Validation Objectives table, containing fields:

VO Identifier

VO description

Related validation means

High-Level Requirements table, containing fields:

Functional description

Related requirements

Test results table, containing fields:

Test identification / date & time

Validation means

Target requirements / high-level requirements / VOs

Validation status

Validation results table, for each participating State/Organisation, containing fields:

High level requirement ID

Validation activities performed.

Each of the baseline Air-Ground and UL SARPs (output from 7th WG3 meeting, Munich, June 1996) has been analysed and the low-level requirements recorded in Word tables, together with a unique identifier and the SARPs paragraph number.

4.2.6 Generic Results

A number of structural and editorial improvements were made to the SARPs texts as a direct result of these activities. For example, SARPs clauses containing more than one requirement were identified and split into separate clauses, redundant "shall" statements were identified and eliminated, and context-free shall clauses (e.g. "The ASE shall...") were given context (e.g. "When event X happens, the ASE shall...").

In some cases a "hanging requirement" was identified, indicating that at some level in the hierarchy there is an unstated or implied requirement that needs to be stated. These observations were fed through to the appropriate SARPs editor.

A number of technical and editorial deficiencies were found, and a Defect Report was sent to the appropriate SARPs editor.

4.3 API Specification

4.3.1 Approach

Service boundaries which are defined within the draft application and upper layer SARPs provide convenient points to examine the behaviour of the prototype applications and simulate the actions of the ATN air and ground application users. Such service boundaries are abstract interfaces used in modelling the behaviour of a system; they would not normally be realised as exposed interfaces in a real implementation. However, for the purposes of validation, the TES project specified Application Program Interfaces (APIs) which correspond closely to some selected service boundaries within the CNS/ATM-1 applications.

ATN End System API specifications were developed by Eurocontrol, using the same style as the X/Open XTI transport interface specification, for the following interfaces:

- ADS air and ground application entity to application-user;
- CM air and ground application entity to application-user;
- CPDLC air and ground application entity to application-user;
- Dialogue Service Interface;
- ATN Transport Service Interface.

These APIs are defined in the C language, in a format compatible with the X/Open Transport Service Interface (XTI). They capture the information interchange that needs to take place between an implementation of the SARPs, and the user application.

Although these API specifications are intended for the TES to validate the SARPs, they were made available to other interested parties as a basis for further work. For example, User Interface application developers may find the ATN End System APIs useful in order to develop applications which can be tested on the TES platform.

The ATN application SARPs define Abstract Service Interface (ASI) Definitions and the Abstract Syntax Notation (ASN.1) for messages exchanged between peer applications. These ASIs were used as the basis of APIs within the prototype applications.

The API usage descriptions which follow describe the normal behaviour of the interface users. Abnormal behaviour is also supported, to allow aborts and emergency situations to be modelled.

4.3.2 Generic Results

As part of the specification work, a number of defects in the draft SARPs were detected; these have been notified to the relevant SARPs editors. Thus, the action of specifying the APIs itself has been shown to be a useful validation activity.

The interface definitions were test-compiled. From this exercise, errors were detected in the original SARPs, which resulted in a number of defect reports being generated.

4.4 Simulation and Modelling

ADS and the Upper Layer draft SARPs have been modelled using the GEODE tool from Verilog. This validated the protocol specifications to ensure that:

- the SARPs text allows all acceptable behaviour (as defined by the service definition and the sequence diagrams);
- the SARPs text does not allow unacceptable behaviour.

The CNS/ATM-1 Package protocols are described in the draft SARPs in textual form (the functionality is also represented in general as a set of state tables). Based on the text in the draft SARPs, the functionality was re-written in the formal definition language SDL (Specification and Design Language - defined by ITU-T Z.100), which is then read and processed by the modelling tool GEODE.

Having developed the model, simulation activities were performed. Each of the valid sequences of events was simulated individually, to ensure that they are all possible. Then random simulation was performed to ensure that no problems occurred when the valid sequences of events are mixed. Exhaustive simulation can ensure that all possible scenarios are tested. This proved to be logistically infeasible for ADS since it required too much machine resource. When certain limitations were put on the upper layers model, exhaustive testing was possible.

Modelling the application protocol using SDL and the GEODE tool allows the sections in the draft SARPs that describe the protocol machine to be validated.

The Geode tool runs on the Hewlett-Packard 9000 series platform, under HP-UX 9.0

4.5 Prototype Implementation

4.5.1 Approach

The prototyping activity involves implementing selected subsets of the ADS, CM and CPDLC applications (both air and ground components), together with supporting upper layers. A major goal is to identify any problems in the draft SARPs during the analysis, design and implementation of the prototypes. The applications will then be available for interoperability testing with other, independent implementations.

Eurocontrol is developing the Trials End System (TES) prototype applications to assist in the validation of draft SARPs for the CNS/ATM-1 Package. The objectives of the overall TES project are:

- the validation of the ATN draft SARPs for air-ground applications and supporting upper layers,
- the production of corresponding prototypes and simulation models,
- the free issue of the software to Eurocontrol member Administrations.

The prototype system was commissioned by the TES project of the Eurocontrol ATN End Systems task (FCO.ET3.STO4).

4.5.2 Architecture

The TES Prototype System comprises hardware platforms, base software and custom software, which will be used initially primarily for the validation of the ICAO draft SARPs for the ATN Upper Layers and Air-Ground ATM applications. The TES environment consists of two major components, the air-based end system and the ground-based end system. The ATN Upper Layers rely on the services provided by the ATN Internet, and provide communication services to the ATM applications. The ATN Upper Layers ensure the end-to-end communication between the two end systems over a number of ATN routers connected via ATN compatible subnetworks.

For the TES Prototype System, the air and ground end systems will communicate using a lower layer protocol stack which can be used in a variety of configurations, in place of the ATN Internet. This point is important, since the objective of the TES project is to validate the SARPs for ATM applications and ATN upper layers, and not the ATN Internet. The TES prototype software will use the X/Open transport service interface. Different communications infrastructure configurations can be "plugged in" beneath the transport interface.

4.5.3 Use for SARPs Validation

The validation procedure comprises the following stages:

- analysis of the draft SARPs requirements;
- production of functional specifications;
- production of design specifications;
- implementation;
- stand-alone tests;
- interoperability tests (using defined simulation scenarios).

Each of these stages may identify different types of errors or omissions in the draft SARPs, and will provide documented evidence in the form of reports on the completeness and accuracy of the draft SARPs, including any assumptions and interpretations which it was necessary to make.

The TES prototyping project will produce prototype software implementations of the following CNS/ATM-1 Package SARPs:

- Automatic Dependent Surveillance (ADS), excluding report forwarding;
- Context Management (CM) Application;
- Controller-Pilot datalink communication (CPDLC), excluding ground forwarding;
- Upper Layer Communications Service (ULCS).

Each of the implementations will include both air and ground based end system components. (The TES project does not currently include the validation of the Flight Information Services SARPs in the CNS/ATM-1 Package).

The TES prototypes are being developed by a European industry consortium led by Thomson, who will independently analyse the draft SARPs, produce functional and design specifications based on the draft SARPs and implement the software realisations. The TES prototype will then be used to test the functionality, interoperability and performance resulting from the draft SARPs

The TES prototypes implement application programming interfaces (APIs) which correspond closely to the upper abstract service interfaces (ASIs) specified in the draft air-ground application SARPs. These APIs provide a common interface which will allow simulation and test tools to be developed separately from the TES prototypes.

The TES prototypes have a table-driven test interface. Where possible, the test data will be based on samples of real data. The test interface will be used to introduce both normal and abnormal events into the TES prototype. Test scenarios are based upon real-life situations, including time based events, single instance of a flight and summation of all flights. These will be used to check the behaviour of the TES prototype and the draft SARPs.

The TES prototype system software is aimed at the validation of the SARPs and would not necessarily be used in an operational environment.

The TES and its components will support a number of configurations on the user side or Human Computer Interface (HCI), which will allow it to be used beyond the initial SARPs validation. These user configurations will include:

- the validation environment;
- a demonstration environment, with user interfaces possibly based on Eurocontrol Bretigny HCIs;
- future experiments based on CNS/ATM-1 Package SARPs;

It is intended that the TES prototype system and its hosted applications will evolve into an ATN Application Reference System, providing a stable implementation of the CNS/ATM-1 Package SARPs once validation is complete, against which other implementations can be tested.

The TES prototypes are implemented to run on Hewlett-Packard 9000-series platforms, under HP-UX 9.0.

4.6 Interworking Testing

When the TES prototypes are completed, Eurocontrol plan to use them for interoperability testing, to achieve further levels of validation.

5. ADS VALIDATION

This section reports the results of the Eurocontrol validation activities for the Automatic Dependent Surveillance Air-Ground application SARPs, based on the tools described in 4.

5.1 Grouping of Requirements

For the ADS SARPs validation activities undertaken by the Agency, the following functional groups of requirements have been identified:

- Periodic contract establishment and execution
- Event contract reporting
- Demand contract request and respond
- Emergency mode operation
- Ground forwarding of reports

In addition, other useful groupings are:

- User Requirements (Chapter 7, and some of 2)
- Technical Requirements (Chapters 2, 5, + 6)
- Protocol (Chapters 3 + 5)
- APDUs (Chapter 4 + 6)
- Subsetting (Chapter 8)
- Other (any requirement not considered in any other grouping)

Each of these groupings ("high-level requirements") is made up of an identified set of low-level requirements ("shall" clauses) and recommendations ("should" clauses).

Table 5-1

	Validation Activity					
Group	**Inspection and Desk Check**	**Require-ments Analysis**	**API Spec-ification**	**Simulation and Modelling**	**Prototype Implement-ation**	**Inter-working Testing)**
Periodic contract establishment and execution	√		√	√	√	
Event contract reporting	√		√	√	√	
Demand contract request and respond	√		√	√	√	
Emergency mode operation	√		√	√	√	
Ground forwarding of reports	√					

User Requirements	√	√				
Technical Requirements	√	√				
Protocol	√			√		√
APDUs	√				√	√
Subsetting	√	√			√	
Other	√					

5.2 Inspection

Eurocontrol has provided the editor for this Sub-Volume, so has performed a considerable amount of inspection and analysis as part of the normal editorial activities.

5.2.1 Application

Inspection began on the SARPs at version 1.0. Inspection is also used to determine non-regression when changes are incorporated as a result of defect reports arising from other validation activities.

5.2.2 Validation Results

As identified in section 3, the “Inspection” activity addresses the VOs indicated in the following table. A “final inspection” was carried out on version 3.0 of the SARPs in August and September 1996. All of the groupings identified above were inspected, with the following conclusions:

Table 5-2: ADS Inspection Results

VO	Result
SVO 1	As far as can practicably be determined, all the system level requirements relevant to ADS are satisfied by version 3.0 of the draft SARPs. This is achieved by the “User Requirements” grouping.
SVO 2	All of the technical requirements arising from other draft SARPs have been checked for inclusion in these draft SARPs. This is achieved by the “Technical Requirements” grouping. Items of common text have been identified and checked for divergences.
FVO 1	The technical requirements have been examined to ensure that they provide the intended functionality. (The functional descriptions are mostly in non-normative notes).
FVO 2	All of the User requirements and recommendations in chapters 2.2.1.7 and 2.2.2.7 have been examined and have been determined to be compatible with the technical requirements.
FVO 4	A number of ambiguities were detected in earlier inspections and have been rectified. No further ambiguities were detected in the final inspection.
FVO 5	A number of inconsistencies were detected in earlier inspections and have been rectified. No further inconsistencies were detected in the final inspection.

VO	Result
FVO 7	The SARPs are independent of any particular implementation constraints as far as can be determined. The abstract nature of the service “interfaces” is not always clear.
TVO 3	The abstract service interface parameters (sections 2.2.1.3, 2.2.2.3) are mapped appropriately to PDU fields and/or Dialogue Service primitives.
TVO 5	The draft SARPs appear to cross refer to, and invoke the ULCS in a manner correct and consistent with the ULCS SARPs. An outstanding defect report relates to AE Qualifier syntax.
TVO 6	The APDU definitions have been inspected and appear correct. An ASN.1 compiler is required to verify the syntax.
TVO 7	QOS management is not a function of the ADS SARPs. Priority and requested RER have fixed values. Routing class (which maps to CLNP security label) is specified by the ADS-User, and can take any of the permitted ATSC values - it is not dynamically managed.
TVO 8	A version number and ASN.1 extensibility markers have been included as an aid to future migration. The ADS Report Forwarding function has been specified as a separate ASE. This appears to be sufficient to meet the requirement for future migration.
TVO 9	PER is invoked, and PER-visible constraints have been specified for optimal encoding efficiency. Some further optimisations are possible.

5.3 Requirements Analysis

5.3.1 Application

The “shall” extraction macros were run on version 2.0 of the SARPs in April 1996 and with version 3.0 of the SARPs in July 1996 .

For version 3.0 of the ADS SARPs, there are:

- 319 low-level requirements (“shall” clauses)
- 13 low-level recommendations (“should” clauses).

The RDB was used in the first step of validation; a paper analysis for consistency and completeness.

The ADS SARPs V3.0 was analysed for low-level requirements (“shall” statements) and recommendations (“should” statements), which were identified and extracted into a Word table.

5.3.2 Validation Results

Table 5-3: ADS Requirements Analysis Results

VO	Result
SVO 2	Comparison of the tabulated requirements of ADS with the other A-G applications reveals that a consistent approach has been adopted.

VO	Result
FVO 6	The tabulated requirements indicate that all stated requirements are necessary.

5.4 API Specification

5.4.1 Application

The initial API specification for ADS was produced in July 1995 (V0.B). It was updated in October 1995 (V0.D) to reflect the V1.0 (Banff) version of stable SARPs, in March 1996 (V0.E) to reflect ADS SARPs V2.0, and again in June 1996 (V0.H) to reflect ADS SARPs V2.3, including the split into two ASEs.

For the ADS API, the ADS-ground-user uses the function ADS_snd<contract>() to request a demand, event or periodic ADS contract, and the ADS-air-user uses ADS_listen() to listen for the receipt of contract requests. The contract requests will be responded to by the ADS-air-user using the function ADS_sndResult(), which is used to deliver all response types to the ADS-ground-user. The ADS-ground-user will listen for responses with ADS_rcvResult(). When events occur that are not part of the normal flow, the function ADS_look() can be used to identify the event and allow the ADS user to invoke a function to handle the event.

5.4.2 Validation Results

Table 5-4

VO	Result
SVO 2	Study of the ASIs in each of the application SARPs ensured that they were specified in a consistent manner.
FVO 2	The “User Requirements” correspond to the requirements at the ASI boundary, therefore specification of the API ensured that such requirements can be conveyed.
FVO 4	Specification of the API ensured that the ASI parts of the SARPs are specified unambiguously.
FVO 5	Specification of the API ensured that the various ASI primitives and their parameters are specified consistently in the SARPs.
FVO 7	Specification of the C language API verified that nothing in the ASI specification required a particular implementation platform.
TVO 7	The provision for QOS management was reflected in the “pass-through” Class of Communication parameter.

5.5 Simulation and Modelling

5.5.1 Application

The “GEODE” protocol simulation and modelling tool was used with version 1.0 of the SARPs during December 1995 - May 1996. The model provided for:

- periodic contract establishment and execution
- demand contract establishment and execution
- event contract establishment and execution
- emergency mode establishment and execution

This validation activity addressed the Protocol groupings of “shalls”.

The ground forwarding of reports (ARF - ADS Report Forwarding) was NOT modelled.

5.5.2 Validation Results

The Protocol groups of shalls were successfully modelled based on version 1.0 of the SARPs, together with proposed changes to rectify the defects that were identified. The state machine was not exhaustively exercised, but a number of scenarios were successfully simulated.

During the modelling and simulation, a number of defect reports were raised and reported to the editor of the ADS draft SARPs. The conclusions listed in the table below can be drawn under the assumption that these defect reports are addressed.

It should also be noted that this work was performed on version 1.0 of the ADS draft SARPs. Changes have been made to the draft SARPs since then, partly because of the defects identified due to this work, and partly for other reasons. Strictly speaking, the conclusions can only be said to apply to version 1.0, but there is a high probability that they can be applied to the later versions of the draft SARPs, since the defects detected have been corrected in these later versions.

As identified in section 3, the Simulation and Modelling activity addresses the VOs indicated in the following table.

Table 5-5

VO	Result
FVO 3	All statements in the section on protocols were modelled, and care was taken not to make any assumptions where there were no “shall” statements. Having built the model, it achieved the functions that were intended - there were no parts of the protocol that were “missing”. It can be concluded, therefore, that the “shall” statements describing the protocol are complete.
FVO 5	The ADS model was built, taking care that all statements were modelled. No part of the model had to be removed in order to be replaced by other statements. Thus it can be concluded that the statements on protocol are consistent.
TVO 1	All end-to-end services were exercised within the model. It was not possible to run exhaustive testing, due to memory limitations in the simulation machine, therefore end-to-end service were not exercised under all possible conditions. It can therefore be concluded that the protocol description meets the end-to-end services in all normal cases.
TVO 2	The protocol was modelled completely. No unacceptable behaviour was detected, although it was not possible to run exhaustive testing, due to memory limitations in the simulation machine. It can be concluded that there is a high probability that there is no unacceptable behaviour in the protocol.

VO	Result
TVO 4	All aspects of the protocol were implemented in the model, including error handling. Error handling was not tested against a model which produced incorrect protocol, and therefore cannot claim that this objective has been fully met. It can be concluded that it is probable that sequence errors in the peer ADS application are correctly handled.
TVO 5	Not applicable. Note: The GEODE model of ADS includes a simple model of the dialogue service. Thus, the only method for the ADS model to use the communications services is through the use of the dialogue service model primitives. It can therefore be implied that the ADS protocol is consistent with the definition of the dialogue service at a high level.
TVO 10	Not Applicable. Note: Use of the GEODE tool in principle allows the automatic generation of C code which implements the system that has been modelled. Since the protocol has been modelled it is possible to automatically generate C code. It can therefore be implied that the protocol can be implemented, in terms of event sequencing. PDU structures are not validated by this exercise.

5.6 Prototype Implementation

5.6.1 Application

Prototype Implementation is based on version 3.0 of the SARPs. The prototype implementation includes:

- periodic contract establishment and execution
- demand contract
- event contract
- emergency mode

It does NOT cover ground forwarding of ADS Reports

5.6.2 Validation Results

To be supplied.

5.7 Interoperability Testing

5.7.1 Application

Interoperability testing, using prototype implementations, is scheduled to start early in 1997

5.7.2 Validation Results

To be supplied.

5.8 Overall Conclusion for ADS SARPs

As far as can be determined from the validation results obtained to date, the ADS SARPs can be considered to have passed the first stages of validation. This is subject to satisfactory corrigenda being produced by the SARPs editor for outstanding defect reports.

Greater confidence will be obtained as the remaining validation activities proceed.

6. CM VALIDATION

This section reports the results of the Eurocontrol validation activities for the Context Management Air-Ground application SARPs.

6.1 Grouping of Requirements

For the CM SARPs validation activities undertaken by the Agency, the following functional groups of requirements have been identified:

- Logon function
- Update function
- Contact function
- Maintain Dialogue function
- Abort function
- Forward User
- Forward Initiator
- Forward Response

In addition, other useful groupings are:

- User Requirements (Chapter 7, and some of 2)
- Technical Requirements (Chapters 2, 5, + 6)
- Protocol (Chapters 3 + 5)
- APDUs (Chapter 4 + 6)
- Subsetting (Chapter 8)
- Other (any requirement not considered in any other grouping)

Each of these groupings ("high-level requirements") is made up of an identified set of low-level requirements ("shall" clauses) and recommendations ("should" clauses).

Table 6-6 <<to be completed>>

	Validation Activity			
Group	**Require-ments Analysis**	**API Specific-ation**	**Prototype Implementation**	**Interworking Testing**
Logon				
Update				
Contact				
Maintain Dialogue				
Abort				
Forward User				
Forward Initiator				

Forward Response				
User Requirements				
Technical Requirements				
Protocol				
APDUs				
Subsetting				
Other				

6.2 Requirements Analysis

6.2.1 Application

The “shall” extraction macros were run on version 2.0 of the SARPs in April 1996 and with version 3.0 of the SARPs in July 1996 .

For version 3.0 of the CM SARPs, there are:

- 129 low-level requirements (“shall” clauses)
- 5 low-level recommendations (“should” clauses).

The RDB was used in the first step of validation; a paper analysis for consistency and completeness.

The CM SARPs V3.0 was analysed for low-level requirements (“shall” statements) and recommendations (“should” statements), which were identified and extracted into a Word table.

6.2.2 Validation Results

Table 6-7: CM Requirements Analysis Results

VO	Result
SVO 2	Comparison of the tabulated requirements of CM with the other A-G applications reveals that a consistent approach has been adopted.
FVO 6	The tabulated requirements indicate that all stated requirements are necessary.

6.3 API Specification

6.3.1 Application

The initial API specification for CM was produced in August 1995 (V0.C). It was updated in October 1995 (V0.D) to reflect the V1.0 (Banff) version of stable SARPs, and in March 1996 (V0.E) to reflect CM SARPs V2.0, including ground forwarding.

For the CM API, both the CM-ground-user and the CM-air-user use the CM_listen() function to detect events. The detected events are then used to call the relevant

CM_rcv<function>(). The CM-ground-user can respond to CM_rcvLogonReq() with CM_sndLogonRsp() to indicate the applications' names and addresses available. The CM-air-user can respond to the CM_rcvContactReq() with a CMsndContactRsp() when the CM_rcvLogonRsp() has been returned by the new flight information region (FIR) air traffic control system. The CM-ground-user can send CM_sndUpdate() to the CM-air-user to update the application names and addresses held when aircraft information is transferred by ground links.

6.3.2 Validation Results

Table 6-8

VO	Result
SVO 2	Study of the ASIs in each of the application SARPs ensured that they were specified in a consistent manner.
FVO 2	The "User Requirements" correspond to the requirements at the ASI boundary, therefore specification of the API ensured that such requirements can be conveyed.
FVO 4	Specification of the API ensured that the ASI parts of the SARPs are specified unambiguously.
FVO 5	Specification of the API ensured that the various ASI primitives and their parameters are specified consistently in the SARPs.
FVO 7	Specification of the C language API verified that nothing in the ASI specification required a particular implementation platform.
TVO 7	The provision for QOS management was reflected in the "pass-through" Class of Communication parameter.

6.4 Prototype Implementation

6.4.1 Application

Prototype Implementation began in June 1996, based on version 3.0 of the SARPs. The prototype implementation includes:

- Logon function
- Contact function
- Update function
- Ground forwarding of CM Reports

6.4.2 Validation Results

To be supplied.

6.5 Interoperability Testing

6.5.1 Application

Interoperability testing, using prototype implementations, is scheduled to start early in 1997.

6.5.2 Validation Results

To be supplied.

6.6 Overall Conclusion for CM SARPs

As far as can be determined from the validation results obtained to date, the CM SARPs can be considered to have passed the first stages of validation. This is subject to satisfactory corrigenda being produced by the SARPs editor for outstanding defect reports.

Greater confidence will be obtained as the remaining validation activities proceed.

7. CPDLC VALIDATION

This section reports the results of the Eurocontrol validation activities for the Controller-Pilot Data Link Communications Air-Ground application SARPs.

7.1 Grouping of Requirements

For the CPDLC SARPs validation activities undertaken by the Agency, the following functional groups of requirements have been identified:

<< to be completed >>

Each of these groupings ("high-level requirements") is made up of an identified set of low-level requirements ("shall" clauses) and recommendations ("should" clauses).

Table 7-9 <<to be completed>>

	Validation Activity			
Group	**Requirements Database**	**API Specif-ication**	**Prototype Implement-ation**	**Interworking Testing**

7.2 Requirements Analysis

7.2.1 Application

The "shall" extraction macros were run on version 2.0 of the SARPs in April 1996 and with version 3.0 of the SARPs in July 1996 .

For version 3.0 of the CPDLC SARPs, there are:

- 255 low-level requirements ("shall" clauses)
- 27 low-level recommendations ("should" clauses).

The RDB was used in the first step of validation; a paper analysis for consistency and completeness.

The CPDLC SARPs V3.0 was analysed for low-level requirements ("shall" statements) and recommendations ("should" statements), which were identified and extracted into a Word table.

7.2.2 Validation Results

Table 7-10: CPDLC Requirements Analysis Results

VO	Result
SVO 2	Comparison of the tabulated requirements of CPDLC with the other A-G applications reveals that a consistent approach has been adopted.
FVO 6	The tabulated requirements indicate that all stated requirements are necessary.

7.3 API Specification

7.3.1 Application

The initial API specification for CPDLC was produced in August 1995 (V0.C). It was updated in October 1995 (V0.D) to reflect the V1.0 (Banff) version of stable SARPs, and in April 1996 (V0.F) to reflect CPDLC SARPs V2.0.

For the CPDLC API, the function CPDLC_sndMessage() is used to construct the message for communication with the remote party, either the CPDLC-air-user or CPDLC-ground-user. This includes the definition of data in structures to convey the message parameters to the intended recipient. There are two receive functions which allow the CPDLC user to receive messages from the message queue held below the API. The CPDLC_rcvMessage() function returns the next message from the message queue. The CPDLC_Alert() function can be made to poll or monitor the message queue for the arrival of messages conforming to the selection criteria. Both functions assume that messages are placed in urgency and then time order within the message queue.

7.3.2 Validation Results

Table 7-11

VO	Result
SVO 2	Study of the ASIs in each of the application SARPs ensured that they were specified in a consistent manner.
FVO 2	The "User Requirements" correspond to the requirements at the ASI boundary, therefore specification of the API ensured that such requirements can be conveyed.
FVO 4	Specification of the API ensured that the ASI parts of the SARPs are specified unambiguously.
FVO 5	Specification of the API ensured that the various ASI primitives and their parameters are specified consistently in the SARPs.

VO	Result
FVO 7	Specification of the C language API verified that nothing in the ASI specification required a particular implementation platform.
TVO 7	The provision for QOS management was reflected in the “pass-through” Class of Communication parameter.

7.4 Prototype Implementation

7.4.1 Application

Prototype Implementation began in June 1996, based on version 3.0 of the SARPs. The prototype implementation includes:

- CPDLC-start, message, end and abort services
- Downstream clearance.

It does NOT cover ground forwarding of CPDLC Reports

The work is being undertaken by an industry consortium led by Thomson in France. Implementation is expected to be completed by February 1997.

7.4.2 Validation Results

To be supplied.

7.5 Interoperability Testing

7.5.1 Application

Interoperability testing, using prototype implementations, is scheduled to start early in 1997

7.5.2 Validation Results

To be supplied.

7.6 Overall Conclusion for CPDLC SARPs

As far as can be determined from the validation results obtained to date, the CPDLC SARPs can be considered to have passed the first stages of validation. This is subject to satisfactory corrigenda being produced by the SARPs editor for outstanding defect reports.

Greater confidence will be obtained as the remaining validation activities proceed.

8. FIS VALIDATION

This section reports the results of the Eurocontrol (ATN End Systems TES project) validation activities for the Flight Information Services Air-Ground application SARPs.

8.1 Requirements Analysis

8.1.1 Application

The “shall” extraction macros were run on version 2.0 of the SARPs in April 1996 and with version 3.0 of the SARPs in July 1996 .

For version 3.0 of the FIS SARPs, there are:

- 155 low-level requirements (“shall” clauses)
- 5 low-level recommendations (“should” clauses).

The RDB was used in the first step of validation; a paper analysis for consistency and completeness.

The FIS SARPs V3.0 was analysed for low-level requirements (“shall” statements) and recommendations (“should” statements), which were identified and extracted into a Word table.

8.1.2 Validation Results

Table 8-12: FIS Requirements Analysis Results

VO	Result
SVO 2	Comparison of the tabulated requirements of FIS with the other A-G applications reveals that a consistent approach has been adopted.
FVO 6	The tabulated requirements indicate that all stated requirements are necessary.

8.2 Overall Conclusion for FIS SARPs

Further validation activities by other States or Organisations are required before the FIS SARPs can confidently be declared validated.

9. UPPER LAYERS VALIDATION

This section reports the results of the Eurocontrol validation activities for the ATN Upper Layer Communications Service (ULCS) SARPs.

9.1 Grouping of Requirements

For the ULCS SARPs validation activities undertaken by the Agency, the following functional groups of requirements have been identified:

- D-START service and supporting protocols
- D-DATA service and supporting protocols
- D-END service and supporting protocols
- D-ABORT service and supporting protocols
- D-P-ABORT service and supporting protocols

In addition, other useful groupings are:

- Application layer structure and naming requirements
- Session Layer Requirements and use of Transport service (Chapter 4)
- Presentation Layer Requirements (Chapter 5)
- ACSE Requirements (Chapter 6)
- Protocol (Chapter 3)
- Other (any requirement not considered in any other grouping)

Each of these groupings ("high-level requirements") is made up of an identified set of low-level requirements ("shall" clauses) and recommendations ("should" clauses).

Table 9-13

	Validation Activity				
Group	**Inspection / Desk Check**	**Requirements Analysis**	**Simulation and Modelling**	**Prototype Implementation**	**Interworking Testing**
D-START service and protocols	√		√	√	√
D-DATA service and protocols	√		√	√	√
D-END service and protocols	√		√	√	√
D-ABORT service and protocols	√		√	√	√
D-P-ABORT service and protocols	√		√	√	√
ALS and naming requirements	√	√		√	√
Session Layer	√	√		√	√
Presentation Layer	√	√		√	√
ACSE	√	√	√	√	√
Protocol	√	√	√	√	√
Other	√	√			

9.2 Inspection

Eurocontrol has provided the editor for this Sub-Volume, so has performed a considerable amount of inspection and analysis as part of the normal editorial activities.

9.2.1 Application

Inspection began on the SARPs at version 1.0. Inspection is also used to determine non-regression when changes are incorporated as a result of defect reports arising from other validation activities.

9.2.2 Validation Results

As identified in section 3, the “Inspection” activity addresses the VOs indicated in the following table. A “final inspection” was carried out on version 4.0 of the SARPs in August and September 1996. All of the groupings identified above were inspected, with the following conclusions:

Table 9-14: ULCS Inspection Results

VO	Result
SVO 1	As far as can practicably be determined, all the system level requirements relevant to ULCS are satisfied by version 4.0 of the draft SARPs.
SVO 2	All of the technical requirements arising from other draft SARPs have been checked for inclusion in these draft SARPs.
FVO 1	The technical requirements have been examined to ensure that they provide the intended functionality. (The functional descriptions are mostly in non-normative notes).
FVO 2	After extensive revision of the Dialogue Service specification at WG3-7, there are no normative requirements placed on the user of the ULCS SARPs. All non-normative user requirements have been examined and have been determined to be compatible with the technical requirements.
FVO 4	A number of ambiguities were detected in earlier inspections and have been rectified. No further ambiguities were detected in the final inspection.
FVO 5	A number of inconsistencies were detected in earlier inspections and have been rectified. No further inconsistencies were detected in the final inspection.
FVO 7	The SARPs are independent of any particular implementation constraints as far as can be determined.
TVO 3	Dialogue Service abstract parameters are mapped appropriately to PDU fields and/or ACSE or Presentation Service primitives. Received Presentation Service primitives are mapped appropriately to ACSE and/or Dialogue Service abstract parameters
TVO 5	Not applicable to ULCS SARPs.
TVO 6	The APDU definitions have been inspected and appear correct. An ASN.1 compiler is required to verify the syntax of application layer PDUs. The ISO specifications of Session and Presentation “short” PDUs have not been fully validated.
TVO 7	QOS management is provided on a pass-through basis. Mapping to Transport and CLNP QOS parameters is satisfactorily defined.
TVO 8	ASN.1 extensibility markers have been included as an aid to future migration. Presentation Context identifier values are open-ended. The naming hierarchy is extensible. This appears to be sufficient to meet the

VO	Result
	requirement for future migration.
TVO 9	PER is invoked, and PER-visible constraints have been specified for optimal encoding efficiency. Some further optimisations are possible.

9.3 Requirements Analysis

9.3.1 Application

The low-level requirements (“shall” statements) have been continuously monitored since ATNP/WG3-4 (Banff) in October 1995.

For version 4.0 of the ULCS SARPs, there are:

- 170 low-level requirements (“shall” clauses)
- 0 low-level recommendations (“should” clauses).

The RDB was used in the first step of validation; a paper analysis for consistency and completeness.

The UL SARPs V4.0 was analysed for low-level requirements (“shall” statements) and recommendations (“should” statements), which were identified and extracted into a Word table. A number of structural deficiencies had been corrected as a result of similar exercise on previous versions of the document, and no new defects were detected.

9.3.2 Validation Results

Table 9-15: ULCS Requirements Analysis Results

VO	Result
SVO 2	Comparison of the tabulated requirements of ULCS with the A-G applications reveals that a consistent approach has mostly been adopted.
FVO 6	The tabulated requirements indicate that all stated requirements are necessary.

9.4 Simulation and Modelling

9.4.1 Application

The “GEODE” protocol simulation and modelling tool was used with version 3.0 of the ULCS SARPs during February 1996 - July 1996. The model provided for:

- the Control Function specification (excluding the “pass-through” between the upper AE service boundary and the upper application ASE boundary)
- the ACSE protocol
- D-START, D-END, D-DATA, D-ABORT and D-P-ABORT services

This validation activity addressed the Dialogue Service groupings of “shalls”.

That part of the control function between the AE user and the application ASE upper interface was not modelled. Since this is defined as a simple pass through (a one-to-one mapping of primitives with no change in parameters), there is no added value in producing a model of this part of the protocol.

The session and presentation protocols were NOT modelled.

9.4.2 Validation Results

The Dialogue service / protocol requirements groupings were successfully modelled, based on version 3.0 of the SARPs together with proposed changes to rectify the defects that were identified. The CF state machine was exhaustively exercised within certain limits, and a large number of scenarios were simulated.

During the modelling and simulation, a number of defect reports were raised and reported to the editor of the ULCS draft SARPs. The conclusions listed in the table below can be drawn under the assumption that these defect reports are addressed.

This work was performed on version 3.0 of the ULCS draft SARPs. Changes have been made to the draft SARPs since then, partly because of the defect raised due to this work, and partly for other reasons. Strictly speaking, the conclusions can only be said to apply to version 3.0, but there is a high probability that they can be applied to the later versions of the draft SARPs, since the defects detected have been corrected in these later versions.

Table 9-16

VO	Result
FVO 3	All statements in the section on protocols were modelled, and care was taken not to make any assumptions where there were no “shall” statements. Having built the model, it achieved the functions that were intended - there were no parts of the protocol that were “missing”. It can be concluded, therefore, that the “shall” statements describing the protocol are complete.
FVO 5	The ULCS model was built, taking care that all statements were modelled. No part of the model had to be removed in order to be replaced by other statements. Thus it can be concluded that the statements on protocol are consistent.
TVO 1	All end-to-end services were exercised within the model. End-to-end services were exercised under all possible conditions. It can therefore be concluded that the protocol description meets the end-to-end services in all cases.
TVO 2	The protocol was modelled completely. During exhaustive testing, no unacceptable behaviour was detected. It can be concluded that there is no unacceptable behaviour in the protocol.
TVO 4	All aspects of the protocol were implemented in the model, including error handling. Error handling was not tested against a model which produced incorrect protocol, and therefore it is not claimed that this objective has been fully met. It can be concluded that it is highly likely that sequence errors in the peer ACSE and control function are correctly handled.

VO	Result
TVO 10	Not Applicable. Note: Use of the GEODE tool in principle allows the automatic generation of C code which implements the system that has been modelled. Since the protocol has been modelled it is possible to automatically generate C code. It can therefore be implied that the protocol can be implemented, in terms of event sequencing. PDU structures are not validated by this exercise.

9.5 API Implementation

9.5.1 Application

The ATN Upper Layer architecture for the CNS/ATM-1 Package includes the specification of a Dialogue Service as the boundary between application service element (ASE) specifications and the control function (CF) specification. An API corresponding to this abstract boundary was specified for the TES prototype system, to allow application SARPs to be implemented and validated in a common manner.

The API specification for the Dialogue Service was produced in April 1996 (V0.G).

9.5.2 Validation Results

The interface definitions were successfully test-compiled. No defects in the ULCS SARPs were detected.

Table 9-17

VO	Result
SVO 2	Study of the use of the Dialogue Service in each of the application SARPs ensured that they were specified in a consistent manner.
FVO 2	The “User Requirements” correspond to the requirements at the Dialogue Service boundary, therefore specification of the API ensured that such requirements can be conveyed.
FVO 4	Specification of the API ensured that the Dialogue Service is specified unambiguously.
FVO 5	Specification of the API ensured that the various Dialogue Service primitives and their parameters are specified consistently in the SARPs.
FVO 7	Specification of the C language API verified that nothing in the ASI specification required a particular implementation platform.
TVO 7	The provision for QOS management was reflected in the Version, Security and QOS parameters. The mapping onto the Transport Service was also specified in the APIs document, verifying that such a mapping is possible.

9.6 Prototype Implementation

9.6.1 Application

Prototype Implementation began in June 1996, based on version 4.0 of the SARPs. The prototype implementation includes:

- Exposed Dialogue Service API
- Specified CF mappings and context identification
- ACSE (edition 2) protocol, PER-encoded
- Presentation “short connect” and “Null encoding” protocols
- Session “short connect” and “Null encoding” protocols.

It does NOT include the full Session, Presentation and ACSE protocols, which would support interworking with full OSI stacks, so strictly is non-conformant to ISO standards.

9.6.2 Validation Results

To be supplied

9.7 Interoperability Testing

9.7.1 Application

Interoperability testing, using prototype implementations, is scheduled to start early in 1997. This will include testing between independently produced implementations.

9.7.2 Validation Results

To be supplied.

9.8 Overall Conclusion for ULCS SARPs

As far as can be determined from the validation results obtained to date, the ULCS SARPs can be considered to have passed the first stages of validation. This is subject to satisfactory corrigenda being produced by the SARPs editor for outstanding defect reports.

Greater confidence will be obtained as the remaining validation activities proceed.