EUROCONTROL

ATNP/WG3/IP ___

12 April 1996

AERONAUTICAL TELECOMMUNICATION NETWORK PANEL

WORKING GROUP 3 (APPLICATIONS AND UPPER LAYERS)

Brussels, Belgium, 15 - 26 April, 1996

Current Results of Eurocontrol Application SARPs Validation Activities

INFORMATION PAPER

Prepared by: Tony Kerr

Presented by: Danny Van Roosbroek

SUMMARY

This paper informs the Working Group of the current status of validation activities within the EUROCONTROL TES project. The subjects of this validation effort are the following draft SARPs for the CNS/ATM-1 Package: ADS, CM, CPDLC and Upper Layers. The WG is invited to review these results and to consider them as inputs to the overall Validation Report to be presented to ICAO.

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Scope

The Eurocontrol Trials End System (TES) project is involved in a number of validation activities in support of the validation of the draft ICAO Air-Ground SARPs and supporting ATN Upper Layers. This paper is a short report on the current status of these activities.

A previous paper (WG3/WP4-13 "Approach to Validation of CNA/ATM-1 Package SARPs") outlined a methodical approach to SARPs validation; many of the concepts in that paper are reflected in the structure of the current paper.

The current application SARPs validation activities are considered under the following headings:

- Requirements database (RDB). Low-level requirements statements in the draft air-ground SARPs are analysed and imported into a database where their validation status can be logged.
- Formal modelling. The ADS and Upper Layers protocol entities are being validated through the use of the "GEODE" modelling tool.
- Prototype Implementation. The TES Prototyping contract will produce implementations of the functionality specified in ADS, CM, CPDLC and Upper Layer SARPs.
- API specification. As part of the specification work for the TES Prototyping Contract, a number of strategic end system application programming interfaces (APIs) have been specified, revealing a number of inconsistencies in the draft SARPs.
- Interoperability Test Scenarios. Scenarios are being developed to support validation by means of inter-operating independent implementations of the Air-Ground SARPs.

1.2 Objectives

A SARPs is considered to be "validated" when each requirement has been validated. An individual requirement is considered to be validated when it has been examined and tested to determine that it is a true and accurate requirement, unambiguous and not in conflict with any other requirement.

The objectives of SARPs validation are to ensure that the draft SARPs are:-

- complete and self consistent;
- unambiguous;
- mutually consistent (within a set);
- achieve the declared user requirement.

2. REQUIREMENTS ANALYSIS AND DATABASE

2.1 Purpose

A requirements database (RDB) is a means to establish a reference point to the requirements defined in the text of a SARPs. An RDB modelled on that developed by ATNP/WG2 for the ATN Internet SARPs has been developed within Eurocontrol, and populated with a subset of the CNS/ATM-1 Package SARPs to prove its viability. This is available on request to other ICAO members as a basis for co-ordinating validation exercises.

It is vital that the level of validation is documented as the various validation methods are applied. To this end, the RDB documents the mandatory ("shall" clauses) and recommended ("should" clauses) statements in the SARPs, against which the results of individual validation exercises can be recorded.

2.2 Development

In developing the database, each draft SARPs document was analysed, to extract every statement which contained a "shall" (mandatory requirement) and every statement that represented a recommended practice. A two-way relationship between the SARPs text and the database has been retained, to minimise the risk of failing to incorporate changes required during validation.

All requirements are traceable through the hierarchy to the top of the requirements tree: i.e. "The avionics and ground systems shall comply with the requirements set down in the sub-volume X SARPs".

The database is similarly to be used to ensure that the guidance material relates to existing mandatory requirements, and is not in itself introducing new concepts which should be mandatory.

Following the stabilisation of the draft SARPs as defined by WG3, the full texts will be entered into the database as a means both of picking up initial inconsistencies, and of tracking validation exercises.

2.3 Results

The database has already been used in the first step of validation; a paper analysis for consistency and completeness.

A hierarchical structure has been applied through the database to determine the related functionality requirements. For example, a requirement stating that: "The position report shall be encoded as ..." is dependent on a superior requirement: "The avionics shall provide periodic position reports". 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 have been fed through to the appropriate SARPs editor.

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

3. FORMAL MODELLING

3.1 Purpose

Simulation exercises allow a model of some aspect of the SARPs to be created (at a much lower cost than a prototype implementation). These models are constructed to exhibit some aspects of the behaviour of a SARPs implementation, and allow limited scale experiments which indicate the likely behaviour of full scale implementations.

Protocol simulation models the behaviour of the protocol state machine, the emission and reception of PDUs and the events at the service interface.

3.2 Modelling the ADS Protocol

ATNP/WG3/SG2 (Air subgroup) has produced draft SARPs for a number of air-ground applications for the CNS/ATM-1 Package, one of which is the Automatic Dependent Surveillance (ADS) application. As part of the validation of that SARPs, a model of the protocol machine written in the formal definition language SDL (Specification and Description Language) has been produced on behalf of Eurocontrol by Verilog using the GEODE tool. (SDL is an internationally recognised language (ITU-T Z.100) that is mainly used for describing the behaviour of telecommunications systems).

The ADS protocol is described in chapter 5 of the draft ADS SARPs, and this functionality is also represented as a set of state tables in appendix A of that document. Validation of the protocol aims to ensure that:

- the SARPs text allows all acceptable behaviour (as defined by the service definition in chapter 3, and the sequence diagrams in chapter 5);
- the SARPs text disallows all unacceptable behaviour;
- there are no states that application can get into, but not get out of (deadlocks);
- there are no groups of states that the application can get into, but cannot get out of into the other states (livelocks).

3.2.1 Model Development

In order to validate the ADS protocol, the ADS SARPs was (graphically) modelled using SDL, using tools supplied by Verilog.

Eurocontrol contracted Verilog to create a model of the protocol written in SDL on their modelling tool GEODE. In order to ensure that the model is an exact mirror of the state machine described in the draft SARPs text, the author of the SDL model copied the text of the draft SARPs, and translated it line by line into SDL. This was done without any attempt to understand the protocol as a whole, thus ensuring that the SDL is an exact portrayal of the text, rather than an interpretation of it.

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 work done so far has already validated that the text itself is consistent (this is analogous to having removed compilation errors from a computer program). Ongoing work will validate that the protocol, as described, does what it is intended to do (this is analogous to testing a program once the compilation errors have been removed).

3.2.2 Model Design

The model of the ADS application describes the following components:

- ADS-air ASE;
- ADS-air User;

- ADS-ground ASE;
- ADS-ground User.

In order to allow the model to simulate two ADS ASEs communicating with each other, the model also includes that part of the Dialogue service which is used by the ADS ASEs. This accepts Dialogue service primitives invoked by one ASE and invokes Dialogue service primitives at the other (and vice versa). It can also simulate communications failure and recovery.

Thus, the working model contains two ASE modules, each with its own lower and upper interface processes, as well as a single "Control Function" module, with its Dialogue service process, and an air-user module. The main signals between the different processes map directly onto the service primitives.

3.2.3 Results

The SDL model of the draft ADS SARPs protocol V1.0 has been completed and delivered to Eurocontrol. The ADS model used as the basis for the validation work was developed against the version of the SARPs dated 6th October 1996 with some corrections to take account of defects detected in the SARPs by Verilog when developing the model.

During the development of the model, a number of defect reports were raised, identifying 38 typographical errors in version 1.0 (mostly in the protocol chapter and the state tables). All these have been corrected in version 1.1. It is expected that, during the ongoing simulation activities, other defects will be discovered.

At the current stage of development, problems still exist within certain parts of the model.

3.3 Running ADS Scenarios

Using the GEODE tool, the development of scenarios and associated scripts to exercise the model was based upon the following steps:

- Cross-checking the model against the SARPs,
- Identification of a set of possible scenarios representing the majority of situations,
- Selection of a representative sub-set of these scenarios to test against the model,
- Generation of the scenarios and associated scripts,
- Running a simulation for each scenario,
- Analysis and Interpretation of results,
- Noting alleged defects and reporting these to the ADS SARPs Editor.

Throughout this task the model was cross-checked against the ADS SARPs in order to check the validity of the model.

In order to choose a manageable number of scenarios a simple selection process was used. This basically produced a set of scenarios most likely to occur in everyday use.

3.3.1 Results of Running Selected ADS Scenarios

3.3.1.1 Demand Contract scenarios

1. This scenario simulates a demand contract being accepted (positive acknowledgement) with no prior dialogue existing. The results of this scenario were that the ADS-demand-contract request was correctly followed by an ADS-

demand-contract indication and an ADS-demand-contract response. No ADS-demand-contract confirmation was given.

2. This scenario simulates a demand contract (with no dialogue existing) being rejected (negative acknowledgement). The results of this scenario were that the ADS-demand-contract request was correctly followed by an ADS-demand-contract indication and an ADS-demand-contract response. No ADS-demand-contract confirmation was given.

3. This scenario simulates a demand contract (with no dialogue existing) being accepted (positive acknowledgement). A ground user abort was simulated after the demand contract indication. The results of this scenario were that the ADS-demand-contract request was correctly followed by an ADS-demand-contract indication and an ADS-demand-contract response. The model is deficient as it did not allow a ADS-User-Abort to be issued.

4. This scenario simulates a demand contract (with no dialogue existing) being accepted (positive acknowledgement). A dialogue service provider abort is simulated after the demand contract indication. The results of this scenario were that the ADS-demand-contract request was correctly followed by an ADS-demand-contract indication and an ADS-demand-contract response. The model did allow an ADS-Provider-Abort to be issued but the behaviour beyond this point was dysfunctional.

3.3.1.2 Periodic Contract scenarios

1. This scenario simulates a periodic contract (with no dialogue existing) being accepted (positive acknowledgement). The results of this scenario were that the ADS-periodic-contract request was correctly followed by an ADS-periodic-contract indication. However the model is deficient in that periodic contracts are not handled by the air user.

2. This scenario simulates a periodic contract (with no dialogue existing) being accepted (positive acknowledgement). After x reports were issued another periodic contract (using the existing dialogue) was sent and accepted (positive acknowledgement). After y reports were issued the contract was cancelled. The results of this scenario were that the ADS-periodic-contract request was correctly followed by an ADS-periodic-contract indication. However the model is deficient in that periodic contracts are not handled by the air user.

3. This scenario simulates a periodic contract (with no dialogue existing) being accepted (positive acknowledgement). After the transmission of x reports, an air initiated emergency report was sent. This was followed by y emergency reports before the cancel emergency reports command was issued. z periodic reports should follow and the scenario be terminated by a cancel contract. The results of this scenario were that the ADS-periodic-contract request was correctly followed by an ADS-periodic-contract indication. However the model is deficient in that periodic contracts are not handled by the air user.

3.3.1.3 Event Contract scenarios

1. This scenario simulates an event contract (with no dialogue existing) being accepted (positive acknowledgement). The results of this scenario were that the ADS-event-contract request was correctly followed by an ADS-event-contract indication and an ADS-event-contract response. No ADS-event-contract confirmation was given.

2. This scenario simulates two interlaced event contracts (with no dialogue existing) being accepted (positive acknowledgement). The results of this scenario were that the ADS-event-contract request was correctly followed by an ADS-event-contract indication and an ADS-event-contract response. No ADS-event-contract confirmation was given.

3.3.1.4 Miscellaneous scenarios

1. This scenario simulates a periodic contract (with no dialogue existing) being accepted (positive acknowledgement). While periodic reports are being issued a demand contract shall be established (with a positive acknowledgement).

2. This scenario simulates a periodic contract (with no dialogue existing) being accepted (positive acknowledgement). While periodic contract are being issued an event contract shall be established (with positive acknowledgement). After an event report is issued the periodic contract is cancelled followed by cancellation of the event contract.

3. This scenario simulates a periodic contract (with no dialogue existing) being accepted (positive acknowledgement). While periodic contract are being issued an event contract shall be established (with positive acknowledgement). An event report is issued followed by more periodic reports and a number of emergency reports. Finally a cancel-all-contracts command is issued.

3.3.2 Conclusions

In every case, analysis of the SARPs and the model showed that the model is deficient and the SARPs is correct. The following deficiencies in the GEODE model have so far been located:

1. In the Ground HI module - only requests are handled, there are no indications or confirmations.

2. In the Air User module - periodic contracts are not handled.

3. User Abort - not available from air or ground.

4. Provider Abort - dysfunctional.

The defects in the model have been a severe impediment to the validation work. All that can be concluded is that those parts of the model that correctly implement the draft ADS SARPs show that the message sequences shown in the draft ADS SARPs can be generated by the protocol model.

It is also noted that the current version of the draft ADS SARPs is not the same as that modelled.

3.3.3 Future Work

Having developed the model, the simulation activities are now progressing. Each of the valid sequences of events will be simulated individually, to ensure that they are all possible. Then random, and finally exhaustive simulation will be performed to ensure that no problems occur when the valid sequences of events are mixed. Exhaustive simulation will ensure that all possible scenarios are tested.

The ADS model will be updated:

a) To correct the defects found, and

b) To bring it in line with the latest version of the draft SARPs.

Following this, the generation of the scenarios presented above can be completed.

3.4 Modelling the Upper Layers

ATNP/WG3/SG3 (Architecture subgroup) has produced draft SARPs for ATN Upper Layers for the CNS/ATM-1 Package. As part of the validation of that SARPs, a model of the state machine is being written in SDL using the GEODE tool.

An interim model, containing approximately 50% of the model, has been developed so far.

3.4.1 Model Design

The model of the upper layers contains the following components:

- ACSE (Association Control Service Element) - conforming to edition 2 of the ACSE standard;
- An ATN application ASE - e.g. ADS - this is not modelled here;
- CF (Control Function).

ACSE is modelled directly from the protocol description given in the ISO standard. It accepts ACSE primitives invoked from the control function, and Presentation service primitives, also invoked from the control function. In response it invokes ACSE and Presentation service primitives back to the control function. The ACSE model does not cover the following conditions:

- ACSE protocol machine does not accept the association, since this has little effect on the CF;
- Presentation resynchronisation, since this is not permitted using the fast byte mechanism which used in the upper layers;
- Presentation exception report, since this is not permitted using the fast byte mechanism which used in the upper layers.

The CF will be modelled from the protocol description given in the UL SARPs. The model does not include that part of the CF that handles service primitives between the ATN application ASE (e.g. ADS ASE) and the user, since this is a simple pass-through function mapping primitive invocations directly one-to-one, with no state information. The CF model will accept dialogue service primitive invocations (e.g. D-START request), ACSE service primitives and Presentation service primitives, and will invoke Dialogue service primitives, ACSE service primitives or Presentation service primitives in response.

In order to allow the model to simulate two upper layer stacks communicating with each other, the upper layers model also includes that part of the Presentation service which is used by the AE. This accepts Presentation service primitives invoked by one control function and invokes Presentation service primitives at the other (and vice versa). It can also simulate communications failure and recovery.

Thus, the working model contains two upper layer modules, each with its own ACSE and CF processes, as well as a single Presentation service module, with its Presentation service process. The main signals between the different processes map directly onto the service primitives.

3.4.2 Results

At the current stage of development, little work has been done on the CF process where the majority of defects are expected to be found.

The following defect has been reported to the SARPs editor:

> The state machine allows a D-START response primitive to be invoked immediately after a D-START request. This is because STA1 is overloaded - it is performing two functions: a) being the association pending state for the originator, and b) being the association pending state for the responder.

4. PROTOTYPE IMPLEMENTATION

4.1 Purpose

A prototype implementation is a complete software package, based on the implementable aspects of the SARPs functionality. Prototype implementations will be used for the undertaking of a set of validation activities, including:-

- local functionality testing
- interoperability (with another remote implementation)
- performance testing over “real” or simulated communications links

Prototype implementations of the draft Air-Ground SARPs will be produced under the TES Prototyping project, using a prototyping methodology which ensures that every aspect of the SARPs is addressed by the prototype, and that any ambiguities or omissions detected during implementation are recorded.

Eurocontrol will build prototypes of ADS, CM and CPDLC, with the supporting upper layers embedded.

The TES validation procedure will consist of a number of phases, which will identify different types of errors or omissions from the draft SARPs. These phases include:

- 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 TES phases will include 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.

4.2 Status

The contract for the development of the prototype software implementations is currently at an advanced stage of negotiation, with a start in early May 1996 currently planned.

5. APIS SPECIFICATION

5.1 Purpose

As part of the specification work for the TES Prototyping Contract, a number of strategic end system application programming interfaces (APIs) have been specified. This work involves a close study of the functionality specified in the SARPs and as such plays an important role in validating the SARPs for consistency and functional integrity. The specification work has revealed a number of inconsistencies in the draft SARPs.

5.2 Results

APIs corresponding to strategic abstract interfaces within the end system have been specified (see WG3/WP6-xx "APIs for Application SARPs Validation").

To date, a number of defect reports have been submitted to the relevant SARPs editors, with further defect reports currently being documented. The defect reports submitted so far are as follows:

5.2.1 Defect Reports on draft ADS SARPs

1. The ADS-modify-emergency-contract Service indicates the handling of a positive acknowledgement can circumvent a response/confirmation but it is only on inspection of the state descriptions (Table 5-16) that it becomes clear that the response/confirmation cannot be used to carry a positive acknowledgement.

2. The definition of Positive Acknowledgement in 3.8.4 implies that the parameter is only used for event positive acknowledgements, however this is not the case.

   The definition of Reply in 3.5.5.1 should be consistent with other definition within the section of the SARPs. The Reply definition in 3.5.5.1 is inconsistent compared to 3.6.5.1 and 3.7.5.1

3. Figure 5-5 fails to show a positive acknowledgement which the text implies would have the same effect as the non-compliance.

4. The air component for the demand, event and periodic contract state tables do not provide the correct management of a negative acknowledgement. All the state tables assume that the negative acknowledgement leaves the air component in an ACTIVE state. Whilst this may be true, if a negative acknowledgement allows the resumption of a previous periodic or event contract this should not be the case for a demand contract or initial periodic or event contract.

   Therefore the handling of negative acknowledgements to contracts should be checked and the assumption that a previous event or periodic contract is resumed upon a negative acknowledgement should be documented.

5. The enumerated values used to represent contract events, such as periodic, event and demand contracts, should consistent throughout the ASN.1 definitions. This would allow the software developers to define the value once and use it through out the software implementation reducing the risk of confusion and implementation error. The enumerated types concerned are CancelContract, ContractType and RequestType.

5.2.2 Defect Reports on draft CPDLC SARPs:

1. For some uplink message elements the order of the definition in the text and definitions are inconsistent. This occurs for PositionLevel and TimeSpeed in a number of messages.

2. The CPDLC User Abort indicates that it has a reason which is a CPDLC message. This appears to be inconsistent with the state table descriptions and the User Requirements descriptions.

   For some message elements the definitions do not represent all the data elements in the message, such as in uM184. In Um184 the message suggests that the following data should be portrayed: (Time, ToFrom, Position)

   However the definition only specifies the use of:

   uM184ToFromPosition [184] UM184ToFromPosition

3. For the Uplink message elements a number of the definitions are incorrect :

   uM26AltitudeTime [26] UM26AltitudeTime

   uM27AltitudePosition [27] UM27AltitudePosition

   uM28AltitudeTime [28] UM28AltitudeTime

   uM29AltitudePosition [29] UM29AltitudePosition

   uM33Level [33] UM33Level

5.2.3 Defect Reports on CM draft SARPs

1. In the state table, the event D-START Indication with user data CMLogonRequest in the IDLE state indicates that a CM-update indication should be invoked however the description in the 5.3.2.4.1.2 indicates that this should invoke a CM-contact indication.

2. The CM Logon Request Message Description states that “For each application that can be ground initiated the aircraft must provide the application name, version number and address. For each application which can be air initiated the aircraft must provide the aircraft name and version number.”

   The last statement is inconsistent with the formal definition for a **CMLogonRequest** which indicates that r**equestedGroundNameAddresses** is a sequence of **APName** which does not include the version number.

3. The ASN.1 formal definition of CMLogonResponse fails to specify the ASN.1 definition to be used for the groundNameAddress sequence.

6. INTEROPERABILITY TEST SCENARIOS

6.1 Purpose

A series of tests of varying complexity needs to be defined to be carried out on prototype implementations. Tests on a single prototype may be defined and devised by the organisation responsible for that prototype. Tests requiring interworking between prototypes need to be collaboratively agreed between the partners.

It has already been recognised that there will not be a one-for-one correspondence of tests to SARPs requirements. With several hundred “shall” statements in a typical ATN Application SARPs, such an approach would probably still be in test in 1999! Instead, an approach using a small number of more complex “scenarios” is proposed, each of which, on successful completion, should give a high degree of confidence in a large number of requirements.

Interoperability tests are specified to ensure that the exchange of information between implementations meets SARPs requirements. A test method similar to that used for OSI interoperability testing may be appropriate, either back-to-back with the same implementation, or testing with an independent implementation.

Performance tests are needed to validate performance aspects of the SARPs, e.g. is it feasible to achieve the required round trip time with current network technology? These tests are likely to be carried out over a real ATN Internet and subnetworks, or else over a network simulator.

6.2 Results

The development of interoperability test scenarios is currently in progress within Eurocontrol. A preliminary set of scenarios has been presented in an earlier paper (WG3/WP5-22 “Use of Interoperability Testing as a Validation Tool”, WG3/WP4-16 “Proposed Scenarios for the CNS.ATM-1 Package Draft SARPs Validation”).

In addition, the TES Prototyping project (see section 4) will result in the production of a number of test scripts which will drive the application AEs from the user interface, and will correspond to pre-defined pseudo-operational scenarios.

7. CONCLUSIONS

This paper presents interim validation results from the Eurocontrol TES project, achieved by a variety of different validation methods. The work will continue in the coming months with the aim of achieving a level of validation suitable for acceptance of the selected draft SARPs by ATNP/2.

The WG is invited to review these results and to consider them as inputs to the overall Validation Report to be presented to ICAO.