EUROCONTROL

ATNP/CCB/WP **6-4**

ATNP/WG3/WP **13-08**

22 June 1998

AERONAUTICAL TELECOMMUNICATION NETWORK PANEL

WORKING GROUP 3 (APPLICATIONS AND UPPER LAYERS)

Utrecht, Netherlands, 29 June - 3 July 1998

Version Control Issues

Prepared by: Tony Kerr, Jean-Marc Vacher

Presented by: Danny Van Roosbroek

SUMMARY

This paper discusses issues that will arise as the ATN SARPs evolve from the ICAO V2.2 baseline.

The Working Group is invited to approve the recommendations in this paper, and to resolve the outstanding issues.

1. INTRODUCTION

At the twelfth meeting of WG3, it was noted that a number of issues will arise from the evolution of ATN SARPs after the publication by ICAO of the first edition of the special Manual. It was agreed that a co-ordinated paper on Version Control issues was needed for the next WG3 meeting.

2. BACKGROUND

At the WG3 meeting in March 1998, WP 12-06 made the following recommendations:

It is recommended that the CCB should be advised to implement the following provisions:

1. Define a Baseline consisting of all PDRs closed in the ICAO V2.2 SARPs.

Then for any PDRs which are still open, or which are subsequently submitted:

2. Use best endeavours to find a solution that will not affect interoperability with the Baseline.
3. If such changes are unavoidable, then ensure that the extensibility features inherent in the data definitions are actually utilised (e.g. insert new field AFTER the ASN.1 extensibility marker).
4. If a major change is required which cannot be accommodated using built-in extensibility features, then the protocol version identifier will have to be incremented (this has no relation with the SARPs document version number). In such cases, interoperability with the Baseline application version will not be possible. This should be a rare event.
5. Within each category, distinguish changes that are safety-critical from those that are enhancements or extensions to current application protocols. For the safety-critical cases, all operational implementations will be required to implement the changes, as is the case for current systems.

These recommendations were accepted, but formed part of a wider discussion of version control issues. This paper attempts to identify the different issues and make recommendations where possible on how ATNP should address the issues.

3. CURRENT POSITION

3.1 Overview

ICAO will publish the Manual of Technical Provisions for the Aeronautical Telecommunication Network (ATN) - Doc 9705-AN/956 (first edition - 1998) in August 1998. For convenience in this paper, the combination of the provisions in this special Manual and the core ATN SARPs in Amendment 73 of Annex 10 is referred to as “Package 1 SARPs”.

The Effective Date of the Package 1 SARPs is the publication of Amendment 73 in August 1998.

It must therefore be assumed that there will soon be operational aircraft that are equipped with Package 1 compliant avionics. **These must be accommodated by ground stations now and in the future.**

At the same time, the operational requirements from ADSP continue to evolve, and these, together with new developments in ATNP, will lead to updated technical provisions, referred to here as “Package 2 SARPs” in the December 1999 time frame.

Also, as operational experience with the Package 1 SARPs increases, it is possible that deficiencies will be found in the technical specifications. It is possible that ICAO State letters in the October 1998 time frame could contain corrigenda to the Package 1 SARPs. The CCB will keep a corrigenda file for edition 2-3.

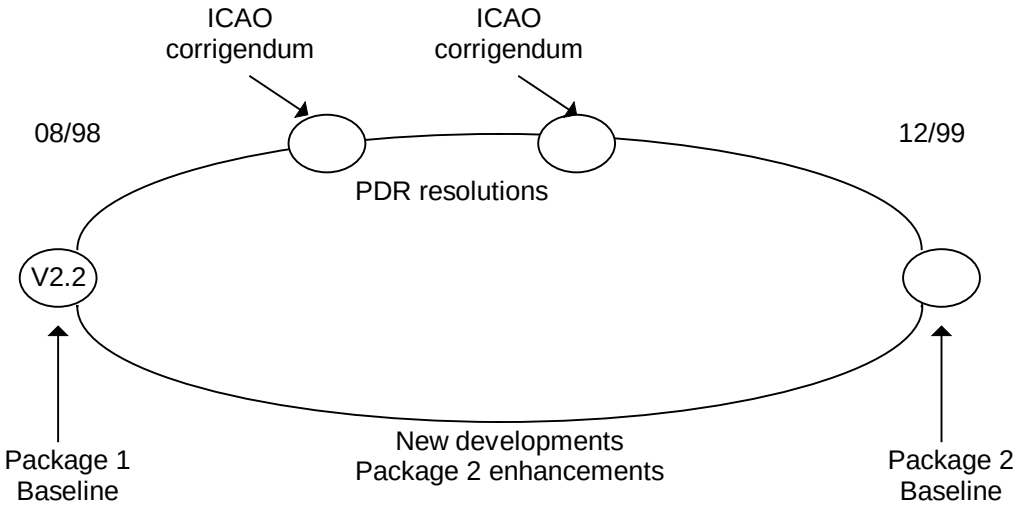

Figure 1. ATN SARPs evolution

3.2 "Package 1" Maintenance

Bugs in Package 1 may be encountered by system implementors or by those defining operational procedures. Errors or ambiguities in specification may result in non-interoperable implementations, or implementations that will not function as required. There may be omissions in the specifications such that operational requirements cannot be supported.

Safety-related problems may be discovered in the underlying operational requirements.

Thus, it can be assumed that it will be necessary to issue essential corrigenda to the Package 1 SARPs, resulting in one or more incremental editions becoming available before the publication of Package 2.

Package 1 work is subject to change control by the CCB, and each modification requires a proposed defect resolution (PDR) to be approved. It is anticipated that critical PDRs will be batched together and distributed by ICAO as corrigenda to the Manual along with regular State Letters.

If a PDR is assessed as safety-critical then more urgent action is required, and procedures need to be defined for alerting ICAO States and Organisations which may have implemented the Package 1 baseline.

For each corrigendum and each safety-critical PDR, a decision has to be taken as to whether:

a) it is necessary to ground aircraft until their software is upgraded,

b) an upgrade is essential at the next routine maintenance opportunity,

c) an upgrade is merely desirable.

3.3 "Package 2" Changes

Package 2 changes could fall into the following categories:

a) Planned functional enhancements - planned developments of Package 1 applications (e.g. enhanced AMHS service) and resolution of Forwarded PDRs (e.g. Upper Layer naming and addressing).

b) New operational requirements - ADSP defines new requirements which impact existing Package 1 applications (e.g. ADS emergency mode)

c) New functional areas - planned developments (e.g. system management, security, multicast) and new applications defined by ADSP or elsewhere (e.g. SAM, METAR) which have no impact on Package 1 implementations.

The baseline is ICAO edition 2.2. Package 2 work is subject to validation by the WGs but is not subject to change control by the CCB until a stable validated baseline is achieved (possibly after approval by ATNP/3).

4. VERSION CONTROL ISSUES

4.1 Requirements baseline

It should be stated formally what version of ADS Manual each version of the ATN SARPs are based on, as there is currently a moving baseline, based on outputs from ADSP WGs. It is understood that all extant changes will be consolidated into a new manual at ADSP/5.

4.2 Document editions

The ATN Panel and its Working Groups are not responsible for documentation management - that is an ICAO responsibility. However, the WG should produce Recommendations to ICAO on how to manage the documentation which it is producing. If these are not accepted, then the consequences need to be considered.

Some of the issues which need to be considered include:

- Maintenance of engineering copies by editors

  Should there be separate streams for edition 2.2 plus PDRs on the one hand and Package 2 enhancements on the other, coming together at ATNP/3, as illustrated in Figure 1?

  Will there be two parallel editions of SARPs at ATNP/3? There has been some disagreement on this point in WG3, one opinion being that Package 1 fixes need to be in place before Package 2 changes are made.

- Publication by ICAO

  It is important for development of the Package 2 material to know what will be the format of Package 2 SARPs. Will this be another special manual?

  Will Package 2 be a delta update to Package 1, or a complete self-contained issue?

  Will ICAO continue to maintain the Package 1 SARPs when the Package 2 SARPs are published, or will Package 2 supersede the current Manual?

- Work-arounds and corrigenda

  The fact that industry progresses faster than ICAO document control needs to be taken into account. For example, if a fatal flaw in a specification is found, developers need a work-around solution quickly, rather than waiting for ICAO to publish a corrigendum.

4.3 Protocol versions

“Protocol version” here refers to the version number associated with each of the application protocol definitions (and conveyed via the CM application for air-ground applications).

The two main issues are:

a) When does the protocol version need to be incremented?

b) What happens if version X of application A in one end system aircraft attempts to communicate with version Y of the same application in another end system, where X may be greater than or less than Y?

4.3.1 Protocol evolution

It is possible for a communication protocol to evolve to a certain extent without needing to be identified as a new version. This has been well-established in the OSI protocols standardised by ISO and ITU-T. In general, it is only necessary to change the protocol version when interworking at a meaningful level is no longer possible between the different versions.

This needs to be considered for each application on a case-by-case basis.

New protocol features can be accommodated by adding new protocol fields or extensions to existing protocol fields in a backwards-compatible manner. **This means that applications are required to ignore (i.e. to be able to detect and skip over) extension fields that they are unable to interpret.**

Extensibility markers in the ASN.1 definitions make it possible to define backward-compatible extensions when PER is used. Any number of extensions can be added incrementally to a baseline definition, but the extensions are not encoded as efficiently as the “root” fields. See Annex B.

The Rio meeting of WG3 established that guidelines for editors on the use and definition of protocol extensions are needed.

4.3.2 Version interworking

Figure 2 illustrates the possible combinations of Version 1 and Version 2 aircraft and ground stations.

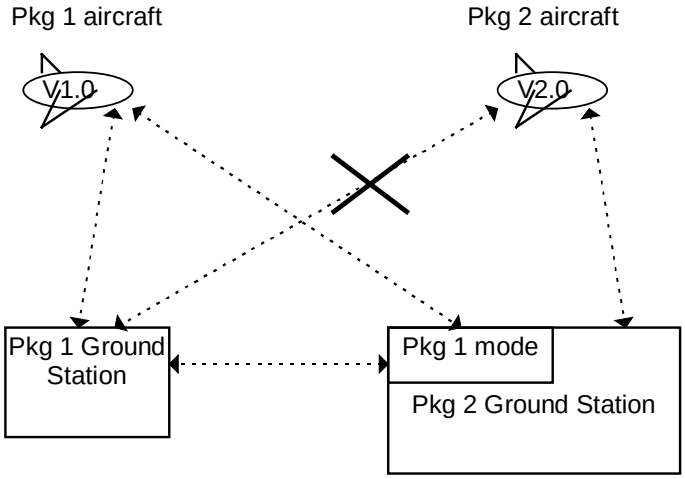

Figure 2. Air and Ground Version Compatibility

This is a simplification, since:

a) intermediate versions are not considered

b) not all applications on a given end system may be at the same version level

c) future versions (3 onwards) are not considered.

The possible combinations are summarised in Table 1.

Table 1. Simplified Interworking Possibilities

	Pkg 1 Aircraft (V1)	Pkg 2 Aircraft (V2)	Pkg 1 Ground (V1)	Pkg 2 Ground (V1 and V2)
Pkg 1 Ground (V1)	Complete compatibility	**Not possible** according to Pkg 1 CM, unless aircraft is dual-stacked.	Complete compatibility	Ground must function in Pkg 1 mode
Pkg 2 Ground (V1 and V2)	Ground must function in Pkg 1 mode	Complete compatibility	Ground must function in Pkg 1 mode	Complete compatibility

4.4 System version

Independently of the SARPs protocol version, each software implementation and each item of equipment will have a version number associated with it. Version control of software is believed to be a certification requirement.

A number of "system" attributes for the ATN management information base (MIB) are under consideration as part of the work on ATN System Management for Package 2. The software and hardware part numbers and version identification are potential system attributes which could be interrogated by a management application.

WG3 in Rio noted that consideration by the JWG may be needed.

4.5 Procedures version

In order to ensure operational interoperability, some consistency of procedures is needed.

This is mostly out of scope for ATNP, but care must be taken that as the SARPs evolve, operational procedures that were possible previously still remain possible.

5. CONCLUSIONS AND RECOMMENDATIONS

The Working Group is invited to take note of the issues raised in this paper, and to agree a common approach to their resolution.

From a practical viewpoint, it is essential that there is a timely resolution of the document edition issues in section 3.2. In particular:

a) The WG should produce Recommendations to ICAO on how to manage the documentation. Recommendations are needed on maintenance (or otherwise) of Package 1 SARPs when Package 2 is published. If not accepted, then the consequences need to be considered.

b) Guidelines are needed for editors producing Package 2 SARPs on the format of the provisions and how to accommodate Package 1 changes.

c) There is a general need for similar discussions about version control in the Core ATN SARPs and the CAMAL.

The other main issues and recommendations are summarised below:

d) Aircraft that are equipped with Package 1 compliant avionics must be accommodated by ground stations now and in the future. A statement is needed about version 1 aircraft operating in version 2 environments and vice versa.

e) The requirement for Package 2 applications to be able to function in Package 1 compatible mode must be borne in mind by the subgroups.

f) If a PDR is assessed as safety-critical then urgent action is required, and procedures need to be defined for alerting ICAO States and Organisations which may have implemented the Package 1 baseline.

g) As changes are made to Package 1 applications, it needs to be considered on a case-by-case basis whether the communication protocol can evolve without needing to be identified as a new version, and if so, what level of interworking is required.

h) Care must be taken that as the SARPs evolve, operational procedures that were supported previously still remain possible.

i) It should be stated formally what version of ADS Manual (including draft material under development in ADSP WGs) each version of the ATN SARPs is based on.

j) Implementations must be required to ignore (i.e. to be able to detect and skip over) extension fields that they are unable to interpret.

k) Guidelines for editors on the use and definition of protocol extensions are needed.

l) The WG should note that version handling is inconsistent between the different Package 1 applications, and attempt to define a common approach in future.

m) The WG should consider whether the current CM provisions, implying that aircraft must carry multiple versions of the application in order to communicate with older ground stations, are acceptable.

n) The Package 1 AIDC protocol has no provisions for version detection and negotiation. The WG should consider the consequences of this.

o) The WG should note that version handling in some cases is implementation dependent (e.g. ARF, Type A Gateway), and should consider the consequences of this.

p) There may be requirements to enable software and hardware part numbers and version identification to be interrogated by a systems management application.

6. ANNEX A - VERSION NUMBER NEGOTIATION IN THE PACKAGE 1 SARPS

The Dialogue Service defined in the ULCS SARPs provides a mechanism for peer applications optionally to exchange version numbers in the range 1..255. This facility is not used by the majority of the Package 1 applications, but could be used in a backwards-compatible way in future Packages. The version number is part of the application context name.

The provisions in the respective Package 1 SARPs can be summarised as follows:

CM: The CM-air-ASE and CM-ground-ASE version numbers are both set to one. The Package 1 airborne specification depends upon the Logon function to support version negotiation (see below). If a Package 1 airborne implementation ever receives a connection request from a V2 ground implementation, the version number is not checked, and it is likely that a protocol error will occur (the Package 1 implementation will not be able to decode V2 PDUs). A Package 1 ground system will check version numbers and reject any connection attempts if the requestor version is greater than the responder. (If the requestor's version is less than the responder's, then it is assumed that the responder can operate in a downgraded mode).

ADS: The ADS-air-ASE and the ADS-ground-ASE version numbers are both set to one. (This is a default value and is not conveyed to the peer). Package 1 has no provisions for version checking; it depends upon CM and the actions of the ADS users to support version negotiation, as described below. If a Package 1 implementation ever receives a connection request from a version 2 implementation, the version number is not checked, and it is likely that a protocol error will occur (the Package 1 implementation will not be able to decode V2 PDUs).

ARF: The ADS-RF-ASE version number is set to one. (This is a default value and is not conveyed to the peer). Being a ground-ground application, it cannot use CM for version negotiation. Package 1 contains provisions that if the version number in a received connection request is "not compatible" then no connection will be established. It is up to implementors to decide what is "compatible".

CPDLC: The CPDLC-air-ASE and CPDLC-ground-ASE version numbers are both set to one. (This is a default value and is not conveyed to the peer). A Package 1 airborne system depends upon CM and the actions of the CPDLC users to support version negotiation, as described below. If a Package 1 air implementation ever receives a connection request from a Version 2 implementation, the version number is not checked, and it is likely that a protocol error will occur (the Package 1 implementation will not be able to decode V2 PDUs). A Package 1 ground system will check version numbers and reject any connection attempts if the requestor's version is not the same as the responder's.

FIS: For this version of the FIS SARPs, the FIS-air-ASE and FIS-ground-ASE version numbers shall both be set to one. There is no version checking in Package 1, as for ADS.

AFTN/ATN Type A Gateway: the version number is explicitly set to one and conveyed to the peer as part of the application context name. Package 1 contains provisions allowing "acceptable" parameter values to be verified in a received connection request. Version handling could be supported, but this is "a local issue". It is up to implementors to decide what is "acceptable".

AIDC: For the first version of the AIDC application, the AIDC-AE version number shall be set to 1 (one). The final arc of the application context name is explicitly set to one. There are no checks for version compatibility in Package 1. If a Package 1 implementation ever receives a connection request from a version 2 implementation, the version number is not checked, and it is likely that a protocol error will occur (the Package 1 implementation will not be able to decode V2 PDUs).

ATSMHS: The ATSMHS specifications are based on international standards, and inherit any version checking capability from these standards. Any future plan introducing new versions of the MHS base standards will take into consideration version checking issues.

The **ULCS** and **ICS** protocols do not carry explicit version numbers.

Version Handling by the CM Application

(The following description is adapted from the CM CAMAL text).

The Package 1 CM SARPs specify the operation of version 1 of the CM application. The version number is a value inherent to the CM and is not provided by the CM users. Since the CM version number is hard coded and changes only with subsequent versions of the SARPs, there will be no confusion due to an entry error or non-standard versions.

When performing a CM-logon service, the air and ground users can discern whether or not their respective CM implementations are compatible. The version negotiation for the CM application is handled by the CM protocol.

Version numbers of other air-ground ATN applications are provided by CM, but it is up to the users of the other air-ground ATN applications to use the correct version numbers in operation. **Other air-ground ATN applications do not perform individual application version negotiation.**

There are three possible cases for CM version number comparisons:

a) Aircraft CM version number equal to the Ground CM version number. In this case, the CM applications in the air and ground are identical versions. There is no need to make the CM version numbers available to either user since the version numbers are identical. Therefore, the exchange of application information commences, and both peer users have the ATN application version numbers.

b) Aircraft CM version number greater than the Ground CM version number. In this case, there is no guarantee of backwards compatibility from the ground user's point of view. Therefore the logon is rejected by the ground's CM-ASE and no application information is exchanged. If the aircraft is capable of supporting an earlier version that is supported by the ground system, another CM logon may be attempted. **This implies that aircraft will need multiple versions of applications, otherwise they will not be able to communicate with older versions on the ground.**

c) Aircraft CM version number less than the Ground CM version number. In this case, the CM application on the ground is assumed to be backwards compatible with the CM application on the aircraft. The application information is accepted by the ground system, the aircraft's CM application version number is made available to the ground user, and the ground's CM information is returned to the aircraft. Since the exchange of CM information was acceptable to the CM-ground-ASE, both peer users are then able to perform individual ATN application version negotiation. The ground user can then decide if any subsequent action needs to be taken based on the aircraft's CM application version number (i.e. whether the ground user wants to perform CM services with an older version of CM).

CM version negotiation for ground-ground exchanges using the CM-forward service are identical to those for air-ground exchanges using a CM-logon service, with the sending ground system taking place of the aircraft.

The CM-update and CM-contact services assume that CM version negotiation has already been performed.

7. ANNEX B - USE OF ASN.1 EXTENSIBILITY

Once extension markers are included in an ASN.1 protocol definition, extensions can be added and some degree of backwards compatibility will be maintained. Specifically, a "V1" implementation which has no knowledge of any extensions, will be able to skip over and ignore any extension fields which are added by a "V2" specification.

Further extensions can be added in the future without the need for any further extension markers in the abstract syntax. If a "V3" implementation, sends to a "V2" implementation, then the V2 implementation will be able to handle only those extensions which it knows about, and will be able to skip over any extensions which are unknown. If a "V3" implementation, sends to a "V1" implementation, then the V1 implementation will be able to skip over any extensions which are present.

The ASN.1 standard gives the following examples:

The first version of a protocol may include a type "A", defined as:

```
A ::= SEQUENCE {        -- extension root type (Version 1)
      a     INTEGER,
      ...           -- extension marker (no extensions defined)
}
```

If the protocol is amended at some future stage, then A could be extended as follows:

```
A ::= SEQUENCE {        -- first extension (Version 2)
      a     INTEGER,
      ...,              -- extension marker
      b     BOOLEAN,    -- b and c are the extension fields
      c     INTEGER
}
```

A further version could then be defined by adding additional elements at the end of the previous extensions, as follows:

```
A ::= SEQUENCE {        -- second extension (Version 3)
      a     INTEGER,
      ...,              -- extension marker
      b     BOOLEAN,
      c     INTEGER,
      d     SEQUENCE {  -- new extension field
            e     INTEGER,
            ...,
            ...,
            f     IA5String
      }
}
```

Note that the new type d is itself an extensible SEQUENCE, and that future extensions can be added either in the middle of the SEQUENCE, between the paired extension markers, or at the end of the previous extensions.

A Version 4 could then be defined, as follows:

```
A ::= SEQUENCE {        -- third extension (Version 4)
      a     INTEGER,
      ...,              -- extension marker
      b     BOOLEAN,
      c     INTEGER,
      d     SEQUENCE {
            e     INTEGER,
            ...,
            g     BOOLEAN OPTIONAL,
            h     BMPString,
            ...,
            f     IA5String
      }
}
```

Future extensions could then be added either after f or after h, and so on.

A Version 1 implementation will still be able to decode a Version 4 PDU, but will ignore the fields b, c, d, e, f, g and h.

Note that, if extension markers had not been used, then a Version 1 implementation would not even be able to recognise a Version 2 PDU, so no interworking at all would be possible.

Effect of extensions on encoding efficiency

Where a SEQUENCE or SET type has an extension marker, but no extension additions (as is the case with Package 1 / Version 1 applications), then there is a one-bit overhead, compared to the same type without an extension marker. Where additions are present in the type, and are actually transmitted in an instance of communication, there is a further overhead of about one octet, plus an additional length field for each extension addition that is transmitted, compared with the same type with the extension marker removed.

It is important to note that either the addition or removal of an extension marker changes the bits on the line, and will in general require a version number change for the protocol.