1. INTRODUCTION

1.1 Purpose

1.1.1 In line with normal ICAO practice, this document was developed as a companion document to the Context Management (CM) SARPs. It may be read alongside the CM Standards and Recommended Practices (SARPs), in order to provide a greater understanding of the specification itself, or it may be read instead of the ATN CM SARPs by readers that simply want to understand the purpose of the CM Application rather than the detail of the specification.

1.1.2 This document also provides some historical information on the development of CM and explanations as to why CM is specified the way it is, including corresponding notes and recommendations, in the SARPs.

1.2 Scope

1.2.1 This document provides guidance material for those implementing the CM Application.

1.2.2 This document does not define any mandatory or optional requirements for CM, neither does it define any recommended practices. This document is not intended to instruct users on how to use CM in a particular operational environment.

1.2.3 The CM application SARPs are dedicated to Air Traffic Services (ATS). Aeronautical Operational Control (AOC) may choose to use the CM SARPs as a model for their own applications.

1.3 History

1.3.1 The CM application allows appropriately equipped aircraft and ground systems or two ground systems to exchange and update data link application information between each other. The Automatic Dependent Surveillance Panel (ADSP) has created new operational requirements necessary to fulfill the initial contact and application information exchange.

1.3.2 The functionality specified by the ADSP in the Manual of ATS Data Link Applications accepted by the fourth meeting of the ADSP (ADSP/4) states that the Data Link Initiation Capability (DLIC) be aircraft initiated, and have the following capabilities:

a) Logon: data link application initiation and, if required, flight plan association,
b) Update: updating of previously coordinated initiation information,
c) Contact: instructions from a ground system to perform data link iniation with another specified ground system,
d) Registration: local dissemination of application information, and
e) Ground Forwarding: ground-ground forwarding of aircraft application information.

1.3.3 The CM SARPs have been developed in order to fulfill the DLIC requirements as set forth by ADSP/4. ICAO has specified that the CM application conform to the ATN protocols for data link operations.

1.3.4 The initial CM concept was a system-level application information (i.e. application addresses, names and version numbers) exchange function between the aircraft and the ground. After an initial log-on message was sent by the aircraft requesting support for specific services, the ground system would reply, passing the necessary address information. Within the same uplinked message, the ground would terminate the dialogue. This left both the ground system and aircraft a list of application information which could be used as necessary. Once the initial application information was received by the ground system, the ground system also had capability to provide updated application information to an aircraft as well as to direct an aircraft to perform a logon to another ground system.

1.3.5 In the course of the development of these SARPs, the concept has become more complex as more operational capability was included. In particular, the ADSP identified a need for the ground system to maintain the dialogue after the initial connection had been made. The requirement for this facility applied particularly, but not exclusively, to aircraft operating in regions where Flight Information Regions (FIRs) can be small, and consequently the exchange of application information with relevant data authorities can be frequent.

1.3.6 There was also an explicit requirement in the ADSP material for the Air Traffic Service Unit (ATSU) which had received information from the CM log-on function to be able to pass aircraft address information to an adjacent ATSU. The ATNP therefore decided to include an element of ground/ground message forwarding in the CM functionality.

1.3.7 States may not be willing to incur the costs of implementing a complete system if they only ever intended to use certain elements of the application. For example, both the maintenance of a dialogue and the ground/ground functionality were seen by some States as having a limited applicability. The SARPs therefore took account of the need to separate out the functionalities to enable partial implementation, while still retaining the interoperability required by the ICAO Standards. This led to the development of subsetting rules, and the identification of conformant configurations.

1.3.8 The ATNP worked very closely with the ADSP to ensure that the development of both the operational concepts and the technical means of achieving them kept in step with each other. However, the ADSP generally looked at a longer timescale than the current ATNP initial implementation programme, and this will inevitably mean that some elements of their work has not been incorporated into the present SARPs.

1.3.9 The ATNP identified a set of packages to accommodate the continuous specification of operational requirements by the ADSP. This document provides guidance material for Version 1 of the CM application. This has been previously known as CNS/ATM-1.

1.3.10 ICAO approved the SARPs and established a configuration control board (CCB) to manage any changes required to the SARPs.

1.4 Structure

1.4.1 Chapter 1 - INTRODUCTION - This chapter contains the reason for providing guidance material as well as the scope. In addition, it provides a brief overview of the CM functionality, CM's relationship with other SARPs, and identifies applicable reference documents.

1.4.2 Chapter 2 - OVERALL GENERAL FUNCTIONALITY - This chapter describes generic concepts that are used throughout the CM SARPs and guidance material. This chapter also covers some implementation issues that are not addressed in the SARPs.

1.4.3 Chapter 3 - CM FUNCTIONAL DESCRIPTION - This chapter gives a functional breakdown of the various services that CM provides. It describes a peer to peer interaction, including reasons for why particular information is used or not used, and what operations on the information are expected.

1.4.4 Chapter 4 - CM CHAPTER DESCRIPTION - This chapter clarifies any functionality that was not addressed in Chapter 3 on a chapter by chapter basis.

1.4.5 Chapter 5 - DIMENSIONS - This chapter gives some sample encoding sizes for guidance on what capacities need to be allowed for in order to meet normal operational expectations.

1.4.6 Chapter 6 - INDEXES / TABLES - TBD

1.4.7 Chapter 7 - EXAMPLE SCENARIOS - This chapter gives some examples as to what typical scenarios one can expect in course of normal CM operation.

1.4.8 Chapter 8 - EXAMPLE ENCODING - This chapter outlines some actual sample PER encoding of typical CM messages.

1.5 CM Overview

1.5.1 This chapter gives a very brief, high level description of the CM as an application allowing an addressing capability for data link applications and contains an outline description of the functions which the CM application provides, namely:

a) The Logon Function - allows the exchange of application information,
b) The Update Function - allows a ground system to modify application data held by an aircraft,
c) The Contact Function - allows a ground system to direct an aircraft to logon to another ground system,
d) The Forward Function - allows a ground system to forward aircraft application information to another ground system, and
e) The Registration Function - allows an aircraft and ground system to make application information available to other applications or communications systems in the aircraft or ground system.

1.5.2 Logon Function

1.5.2.1 The logon function defines a method for the aircraft system to provide its data link application information along with additional flight plan information as required to a ground system. In response, the ground system provides its data link application information to the aircraft. The logon function is not restricted to logons solely with ground systems that provide ATC services; it may also be used to exchange and store addressing information to a centralized location.

1.5.2.2 The logon function is used when the aircraft or ground system does not have a priori knowledge of the application information needed in order to initiate data link operations. If a ground system has the aircraft's CM address, it may use the update function in lieu of using the logon function (as detailed in 1.5.6).

1.5.2.3 The logon function refers to an application level (i.e. user) logon; the communications logon service is provided by the underlying communications protocols. Note that the logon function is not the equivalent of a logon to a computer system. A logon to a computer system provides the initial mechanism that allows all other functions to be performed. The logon as used in CM is strictly for the exchange, correlation and maintenance of data link application information, and is independent of other data link functions such as ADS/FIS contract establishment or CPDLC dialogue initiation.

1.5.2.4 The logon function can be initiated by either the pilot or the avionics system. In order to invoke the logon function, some information must be provided by the avionics or pilot prior to invocation. This information includes:

a) The Initial Ground CM address, which must be known by or input to the avionics, and is the address of the ground system's CM application that is to be logged on to,
b) The Airframe Flight ID, necessary for inclusion so that the logon information is able to be correlated to the proper flight plan,
c) CM Address, which is needed for update and ground forwarding purposes so that other end systems can communicate with the aircraft's CM application,
d) Application name and version number for all air-only initiated services that can be supported. This information is required so the ground system can determine version number compatibility for the specified applications. Addresses are not needed, since the ground cannot initiate the request of an air-only initiated service. Up to 256 applications can be specified. Note that if an aircraft does not want to support a particular service, it need not provide the application name and version number of that service,
e) Application name, version number and address for all ground-initiated services that can be supported. This information is required so the ground system can both determine version number compatiblity and know the addresses for the specified applications. Note that applications that can be both ground and air initiated need to have their addresses provided for the case of ground initiation. Also, if an aircraft does not want to support a particular service, it need not provide the application name, version number and address of that service. However, if this precludes an aircraft from operating in a particular environment that will have to be dealt with by local operational procedures. Up to 256 applications can be specified,
f) Other flight information as required , which is provided to allow unambiguous association of the aircraft with flight plan information stored on the ground. The other flight information as required refers to the facility designation, departure and destination airports, and date/time of departure. All of this information is optional. Examples of usage would include repetetive flights to the same destination (i.e. shuttle flights), where the distinguishing feature of the flight would be the date/time of departure since departure and destination airports would be the same,
g) The Called Facility Designator and the calling 24-bit Aircraft Address. These must be supplied for use by the dialogue service provider,
h) CM version number. This is necessary to ensure that the aircraft and ground system CM applications can interact properly, and
i) Class of Communication Service, which is a user option that specifies if there is a required network performance.

1.5.2.4.1 Upon receipt of the logon request, the ground system checks for CM application version number compatibility. If the version numbers are incompatible, the ground system will not accept the logon. If the version numbers are compatible, the ground system makes the information contained in the logon request available to the other applications within the ground system.

1.5.2.4.2 If the version number of the aircraft's CM application is compatible but different than the version number of the ground system's CM application, then consideration of the different versions must be taken into account when using CM features that are not backwards compatible.

1.5.2.4.3 Using the information contained in the logon request, the ground system establishes a correlation between the aircraft and the appropriate flight plan held on the ground. This ensures that there is an unambiguous mapping of the aircraft to its flight plan and that the ground system is communicating with the intended aircraft. If the information in the logon request is not sufficient for correlation, other means to establish the correlation should be used.

1.5.2.4.4 The addressing information contained in the logon request needs to be made available to the other applications (such as ADS, CPDLC, FIS, etc). The dialogue service provider will also need the application information for use in naming and addressing.

1.5.2.4.5 The ground system then responds to the logon request with a logon response. The logon response contains:

a) The application names, addresses and version numbers for the requested applications that can be air-initiated for all versions that the ground and aircraft can support. Up to 256 applications can be specified,
b) The application names and version numbers of the ground-initiated applications that are supported. The addresses do not need to be relayed to the aircraft since the ground is the only allowed initiator. Up to 256 applications can be specified, and
c) An indication on whether or not a dialogue is to be maintained.

1.5.2.4.6 Upon receipt of a logon response, the aircraft must first determine whether or not the logon was successful by checking to see if the version numbers were compatible. If the logon failed because the version numbers were not compatible, but a compatible version of CM exists on board the aircraft, then another logon using that version number should be attempted. If a compatible version of the CM application is not available, then local procedures must be executed in order to determine the level of service that can be given to the aircraft in its present environment.

1.5.2.4.7 If the logon was successful, the avionics must make the information contained in the logon response available to the other applications (such as ADS, CPDLC, FIS, etc) supported by the aircraft. The dialogue service provider will also need the application information for use in naming and addressing.

1.5.3 Update Function

1.5.3.1 This function provides the capability for the ground system to update application information contained in an aircraft system. This can be used if addressing or application information on the ground has changed (e.g. CPDLC is now unavailable) or in lieu of a logon if the aircraft's application and addressing information has been received by another means (e.g. ground-ground forwarding).

1.5.3.2 Generally the update function is invoked after a logon has been performed, so the ground system is sure to have the correct aircraft address. However, if a ground system has the correct aircraft information and a logon has not been performed, there is nothing to preclude the ground system from using the update function to provide the aircraft with its ground application information.

1.5.3.3 The ground system must provide the following information for the update function:

a) The application names, addresses and version numbers for applications that can be air-initiated. Up to 256 applications can be specified.
b) The application names and version numbers of the ground-initiated applications that are supported. Up to 256 applications can be specified.

c) The called aircraft address and calling facility designation (if a dialogue does not exist), and
d) An indication of the Class of Communication required (if a dialogue does not exist).

1.5.3.3.1 If the aircraft is maintaining information from the sending ground facility of the update information, then the aircraft will update only that information of the sending ground facility (i.e., other application information that is held for other ground systems are not affected).

1.5.3.4 If the aircraft does not have information from the sending ground facility of the update information, then the aircraft will treat information the same as it would for a logon response.

1.5.4 Contact Function

1.5.4.1 This provides the capability for the ground system to request the aircraft to initiate the logon function with another designated ground system. If ground-ground connectivity does not exist between a ground system that has an aircraft's application information and a ground system that does not, the contact function can provide a means to provide the aircraft's data link application information to the ground system in need. If ground-ground connectivity exists between a ground system that has an aircraft's application information and a ground system that does not, the ground-ground forward function may be used to pass the aircraft's application information to the ground system in need, and the contact function does not have to be used.

1.5.4.2 The ground system must provide the following information for the contact function:

a) The called aircraft address and calling facility designation (if a dialogue doesn't already exist) , to be used by the dialogue service provider for addressing purposes,
b) An indication of the Class of Communication (if a dialogue doesn't already exist),
c) The facility designation of the ground system that the aircraft is to contact, and
d) The CM address of the ground system that the aircraft is to contact.

1.5.4.3 Upon receipt of a contact request, the aircraft will perform a logon function (as detailed in 1.5.2) with the ground system identified in the contact request.

1.5.4.4 After the logon function has been completed, the aircraft will send a response to the initiated ground system indicating whether or not the contact (i.e. logon) with the indicated ground system was successful.

1.5.4.5 If the contact was not successful, then another contact may be attempted or another means may be employed in order to exchange the aircraft's application information with the desired ground system.

1.5.5 Forward Function

1.5.5.1 The forward function provides the capability for a ground system to forward information received from a CM logon function to another ground system. This function is initiated by a ground system, which supports ground-ground forwarding, having completed a successful logon that then forwards the aircraft CM logon information to other ground systems.

1.5.5.2 This allows ground systems an alternative to using the contact function for disseminating aircraft data link application information. There may be advantages to supporting this functionality: ground-ground communication may be more cost and time effective for application information dissemination than air-ground communication, the forward function can facilitate a centralised CM server, and the forward function provides a mechanism for the ground to initiate an application information exchange with an aircraft through the update function instead of waiting for the aircraft to make the initial contact. Offsetting these advantages are added implementation costs and procedural development.

1.5.5.3 The forward function is a one-shot exchange of information only. There is no concept of maintaining a dialogue between two ground systems for CM.

1.5.5.4 The initiating ground system must provide the following information for the forward function:

a) The called and calling facility designations, to be used by the dialogue service provider for addressing purposes,
b) The CM application version number, to be used for CM application compatibility purposes,
c) The information that was contained in the logon request as received from the aircraft (Airframe Flight ID, aircraft CM address, application information and any other flight information), and
d) An indication of the Class of Communications if required.

1.5.5.5 All ground systems that support CM must be capable of recognizing a forward request from another ground system. A ground system is not required to implement the full functionality associated with the processing of the received information. That is, a ground system that receives forwarded information may respond that the processing of that information is not supported. Note that receiving of forwarded information is independent from the sending of forwarding information; a particular ground system implementation may not support the processing of received forwarded information while supporting the sending of forward information. This is an implementation subset, which is detailed further in Chapter 8.

1.5.5.6 Upon receipt of forwarded information, if the receiving ground system does not support the processing of the forwarded information, it will reply to the originating ground system with a “service not supported” result.

1.5.5.7 If the receiving ground system does support the processing of forwarded information, it must first check to see that the CM application version numbers are compatible. If they are not, the forwarded information is rejected and the originating ground system is notified that the versions are incompatible.

1.5.5.8 If the version numbers are compatible, then the receiving ground system will make the information contained in the forward request available for flight plan correlation and to the other applications (such as ADS, CPDLC, FIS, etc) within the receiving ground system. The receiving ground system will then indicate to the sending ground system that the forward function was completed successfully.

1.5.6 Registration Function

1.5.6.1 The registration function is a local implementation that provides a method for the CM users to make available application name, address and version number for each application exchanged in the logon, update or forward functions to other applications and communication systems in the aircraft or ground system.

1.5.6.2 In addition to the application information available, any flight information that is exchanged is also made available and is used for correlation purposes.

1.5.6.3 Upon registration and correlation of an aircraft’s or ground system’s application information, it is a local procedure to determine when specific data link services such as ADS, CPDLC or FIS will be initiated.

1.5.6.4 Since the registration function is a local implementation, there are no messages specified by the SARPs for this function.

1.6 Inter-relationships with Other SARPs

1.6.1 There is no direct interaction between then CM application and the other applications specified in the SARPs, but the addressing information exchanged by CM is required in order to make use of the other application SARPs.

1.6.2 The CM SARPs also make use of the Upper Layer Application (ULA) SARPs (Ref 1.8x) to perform dialogue service functions required by the CM application. Since the ULA also handles addressing, the addressing information provided by CM must be made available to the ULA service provider.

1.7 Structure of the SARPs

1.7.1 All the air-ground SARPs are produced to a standard format. This has greatly helped the maintenance of document stability, commonality and presentation. The CM SARPs are no different in basic layout from all other air-ground applications SARPs.

1.7.2 The CM SARPs constitute the first part of Sub-volume II.

1.7.3 Section 2.1.1 - INTRODUCTION - gives a very brief, high level description of CM, as an application enabling the exchange of application information in order to perform data link operations. Since this overview contains no information directly related to the stipulation of specific standards, it is almost entirely written as series of informative notes.

1.7.4 Section 2.1.2 - GENERAL REQUIREMENTS - contains information and high level requirements for the maintenance of backward compatibility and error processing.

1.7.5 Section 2.1.3 - ABSTRACT SERVICE - defines the abstract service interface for the CM Application. The CM Application Service Element (CM-ASE) abstract service is described from the viewpoint of the CM-air-user, the CM-ground-user and the CM-service-provider.

1.7.6 Section 2.1.4 - FORMAL DEFINITION OF MESSAGES - describes the contents of all permissible CM messages through definition of the CM ASN.1 abstract syntax. All possible combinations of message parameters and their range of values are detailed.

1.7.7 Section 2.1.5 - PROTOCOL DEFINITION - splits up the specification of the CM protocol into three parts: sequence diagrams for the services covered by the abstract service, protocol descriptions and error handling for the CM-Air- and Ground-ASEs, and State Tables.

1.7.8 Section 2.1.6 - COMMUNICATION REQUIREMENTS - specifies the use of Packed Encoding Rules (PER) to encode/decode the ASN.1 message structure and stipulates the Dialogue Service requirements, including Quality of Service (QoS).

1.7.9 Section 2.1.7 - CM USER REQUIREMENTS - describes the requirements imposed on the CM-users concerning CM messages and interfacing with the CM-ASEs.

1.7.10 Section 2.1.8 - SUBSETTING RULES - specifies conformance requirements which all implementations of the CM protocol obey. The protocol options are tabulated, and indication is given as to whether mandatory, optional or conditional support is required to ensure conformance to the SARPs. These subsetting rules will permit applications to be tailored to suit individual ground implementations, commensurate with the underlying task, while still maintaining an acceptable level of interoperability.

1.8 Future Considerations

1.8.1 While CNS/ATM is still in operational use, any future version of data link will be capable of expecting CM Version 1. Without this backwards compatibility capability, there may be no way for future data link versions to recognize a logon attempt in order to respond properly. The rules dependent on this may or may not be applicable (i.e., the protocol may not work properly).

1.9 References

1.9.1 [1] Context Management Application, Annex 10, Volume III, Part 1, Chapter 3 (ATN), Appendix A, Sub-volume II - Air-Ground Applications, section 2.1

1.9.2 [2] Upper Layer Communications Service, Annex 10, Volume III, Part 1, Chapter 3 (ATN), Appendix A, Sub-volume IV

1.9.3 [3] ICAO Manual of Air Traffic Service (ATS) Data Link Applications, ICAO ADS Panel, ADSP/4, September 1996

1.9.4 [4] Internet Communications Service, Annex 10, Volume III, Part 1, Chapter 3 (ATN), Appendix A, Sub-volume V

1.9.5 [5] Introduction and System Level Requirements, Annex 10, Volume III, Part 1, Chapter 3 (ATN), Appendix A, Sub-volume I

2. OVERALL GENERAL FUNCTIONALITY

2.1 General

2.1.1 The CM SARPs is defined as a single application handling the air-ground and ground-ground functionalities.

2.2 Topology

2.2.1 The CM application may function in either an end user-to- end user or client/server concept.

2.2.2 An end user-to-end user concept consists of a single aircraft communicating with a single ground system. Both the aircraft and the ground system constitute fully functional, independent end systems and have the necessary application and addressing capabilities. This is depicted in Figure 2.1. Note that the aircraft can communicate with more than one ground system independently. The ground system can also communicate with another ground system independently.

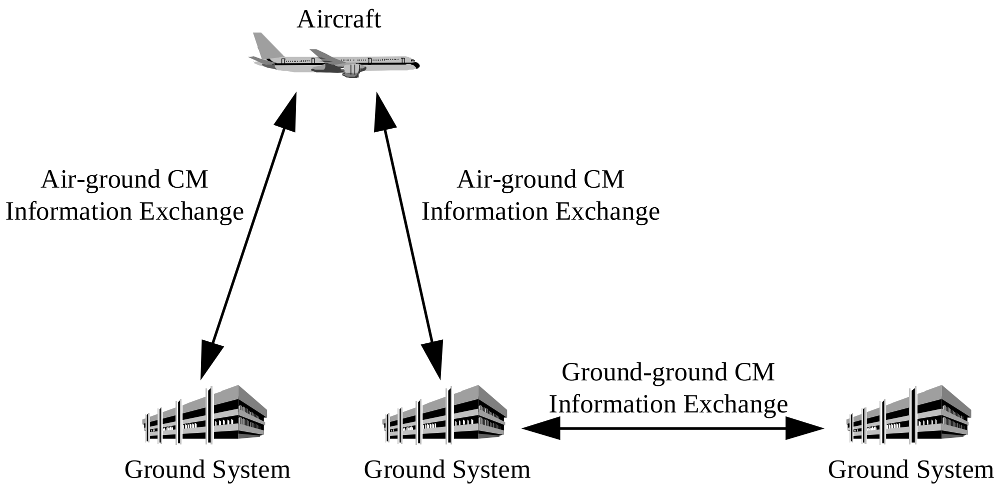

Figure 2.1. End user-to- end user Concept

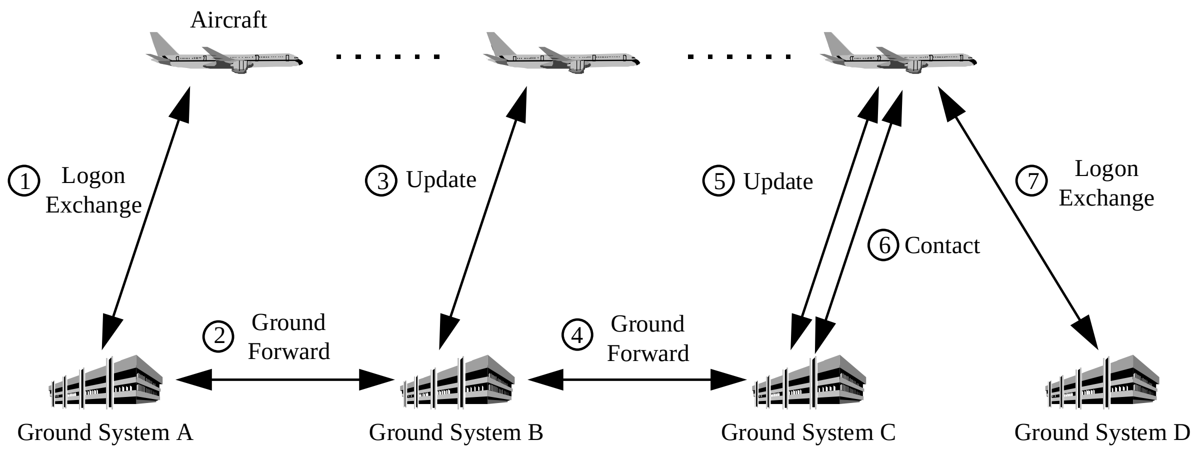

Figure 2.2. CM Application Information Exchanges

2.2.3 In this configuration, addressing information made available by the CM application may be passed, either air-ground or ground-ground, as needed and as supported by the various ground systems. Figure 2.2 illustrates the various methods of application information passing for a single aircraft's flight. The numbers in Figure 2.2 indicate the order in which the various exchanges happen and are explained below:

1) The aircraft performs a logon with Ground System A in order to exchange application information.
2) Since ground-ground connectivity exists between Ground System A and Ground System B, Ground System A forwards the information contained in the initial logon exchange to Ground System B.
3) Ground System B has a need to perform data link applications with the aircraft. Since Ground System B has received the necessary application information for the aircraft from Ground System A, an update can be performed with the aircraft. This provides the aircraft with all of the necessary application information from Ground System B. Data link application services can now take place between Ground System B and the aircraft.
4) Again, since ground-ground connectivity exists between Ground System B and Ground System C, Ground System B forwards the information contained in the initial logon exchange (from step 1) to Ground System C.

5) Ground System C performs an update with the aircraft in order to provide the aircraft with Ground System C's application information.
6) Since no ground-ground connectivity exists between Ground System C and Ground System D, the ground forwarding of application information is not able to take place. Therefore, the contact function is used. Ground System C instructs the aircraft to logon with Ground System D.
7) Upon receipt of the contact instruction, the aircraft performs a logon with Ground System D, so both Ground System D and the aircraft have each other's application information, and data link application services between them can commence.

2.2.4 For the client/server architecture, the concept of a centralized CM server is introduced. In order for addresses of aircraft and ground systems to be readily available for a particular geographic region, a centralized CM server concept may be implemented. This is depicted in Figure 2.3. In this case, an aircraft will logon to a system that will store its application addresses. Other ground systems that wish to contact the aircraft can obtain the proper addressing information from the central server via a ground forwarding or other means (e.g. telephone). In that way, an aircraft is not required to logon to each facility that requires the aircraft's application information; the facility will obtain the proper information and perform a CM-update service in order to let the aircraft know what applications the facility supports. Therefore the centralized CM server may be capable of supporting communications other than Air Traffic Control (ATC).

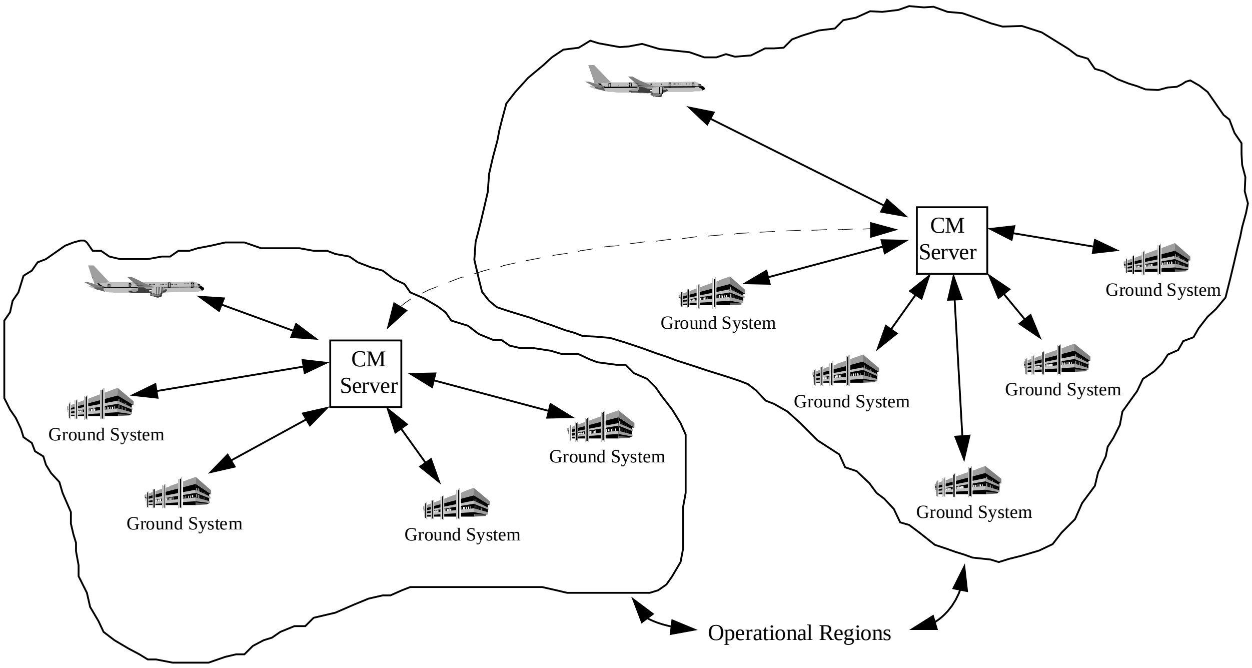

Figure 2.3. Centralized CM Server Concept

2.2.5 The central CM server also provides an easy method for the aircraft to initially log on before pushback. Since the CM server is accessed by all ground systems that need the aircraft's application information, the aircraft need only ever to logon to the same CM server address. This has the advantage of being easily supported by the aircraft's avionics, and makes it very simple for initiation by the pilot (e.g. push a "logon" button) or the automatically by the avionics. If the aircraft is required to contact a facility that does not make use of the central CM server, it can be told the proper CM addressing requirements through a CM-contact service by the central CM server.

2.2.6 Note that the links between the ground systems and the CM server as depicted in Figure 2.3 do not have to be CM connections. Some other communication mechanism can be used by the CM server or ground system(s) to request and/or send the appropriate application information. However, it is also possible for the CM server to relay the aircraft's application information to various ground systems by using the CM forward function. The receiving ground system can then initiate the desired data link services with the aircraft.

2.2.7 It is also possible for two CM servers to exchange application information, as depicted by the dashed line in Figure 2.3. This can be done by using the forward function or by some other means. Connectivity between CM servers of particular operational regions can further simplify some of the addressing concerns for the aircraft; an aircraft may be able to logon to a CM server for the initial part of a flight and have the CM server forward the aircraft's information to an appropriate CM server of a downstream operational region.

2.2.8 Maintaining a CM dialogue with a centralized CM server is described in 2.8.

2.2.9 There are a number of considerations that need to be taken into account for a centralized CM server architecture. While a centralized CM server works well for ground-initiated applications, there are complications for applications that can be air-initiated. Unless an appropriate level of intelligence is built into the CM server, there is no way that the server can know which ground system the aircraft needs to communicate with. The only addresses that the aircraft knows is what the CM server responds with. Therefore, in order to respond with the application information of an appropriate ground system, the CM server must be able to identify from a locally stored flight plan, the flight plan information contained in the logon from the aircraft, or some other means which ground system the aircraft would need to communicate with. The CM server then must also have that ground system's application information stored locally so that it can respond to the aircraft's logon message with the appropriate application information.

2.2.10 The CM server also does not lend itself well if there are multiple FIS servers in an operational region. Unless the CM server has sufficient intelligence to know what FIS address the aircraft will need to use, then the aircraft would have to get the FIS application information by some other means. In the case of a single FIS server for a particular operational region, this problem is alleviated since there is only one possible FIS address to give to the aircraft. For this particular case of FIS infrastructure, a central CM server would work well.

2.2.11 CPDLC would also have the same problem as the multiple FIS server environment. One operational solution would be to not allow aircraft to start a CPDLC dialogue. If this is not acceptable and CM server is going to be used, then the CM server must have sufficient intelligence to know which ground system's CPDLC address to return to the aircraft.

2.2.12 Since ADS is a ground-initiated application, it does not have the same concerns as FIS and CPDLC. The CM server may return all of the ADS application version numbers that the ground systems within the operational region can support to the aircraft.

2.2.13 In order to further illustrate some operational scenarios with a central CM server, two examples are given; one with a “dumb” CM server (the CM server does not know enough of the aircraft’s intent to determine which ground system will require data link services with the aircraft) and a “smart” CM server (the CM can determine the appropriate ground systems that require data link services with the aircraft). These scenarios are not meant to be exhaustive guidance on the implementation of a CM server, but to highlight some of the concerns of a CM server concept.

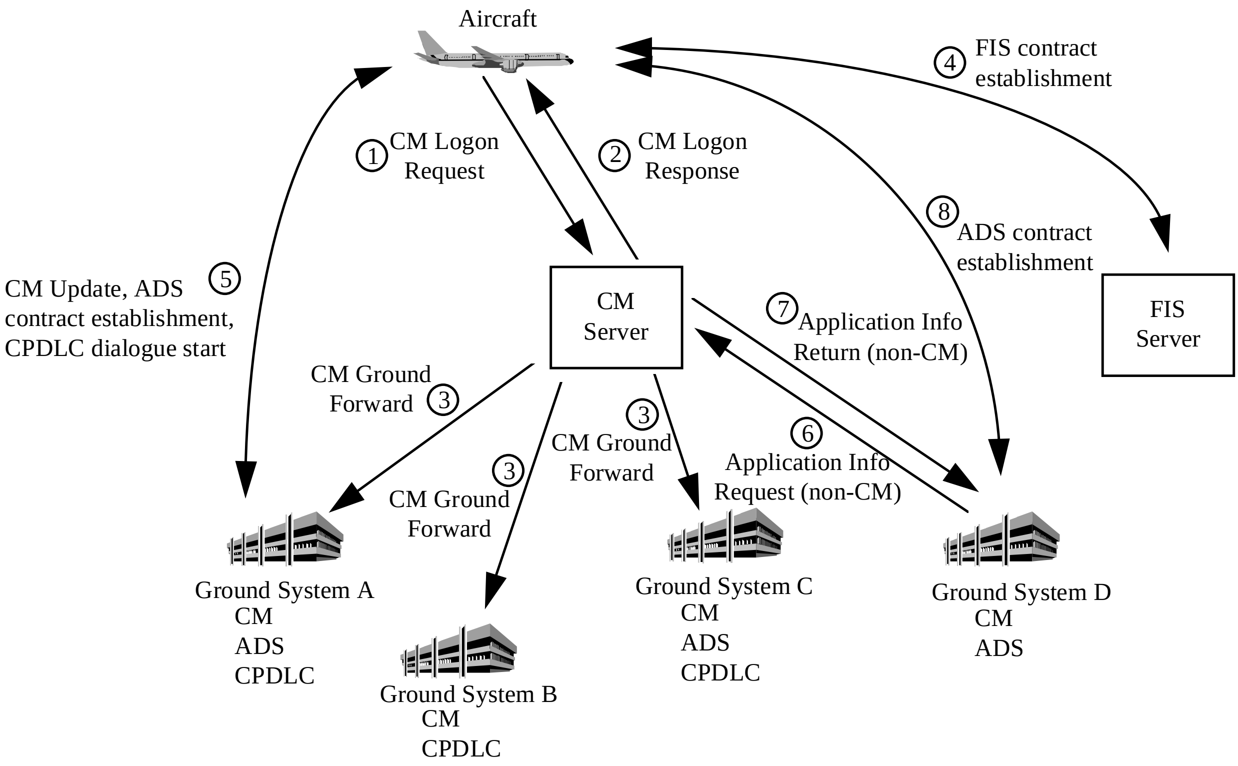

Figure 2.4. “Dumb” CM Server Scenario

2.2.14 For the configuration depicted in Figure 2.4, the CM server does not have any knowledge of the application information of the ground systems except for CM. In this case, the CM server has the CM application address and facility designation of Ground Systems A, B and C. In addition, the CM server has ground-ground CM connectivity with Ground Systems A, B and C. The CM server does not have ground-ground CM connectivity with Ground System D. Additionally, there is a central FIS server (for simplification) which is registered with the CM server. The ground systems each support the applications as indicated. The numbers in Figure 2.4 are explained below:

1) An aircraft sends a CM logon request to the CM server.
2) The CM server responds with either nothing or only application name and version number for the ADS or CPDLC applications, and the FIS server’s application version number and address.
3) Upon sending the logon response, the CM server automatically performs a CM Ground Forward with all ground systems with which it has ground-ground connectivity (Ground Systems A, B and C). The CM server does not know which ground systems need the aircraft’s information, so it sends it to all. Ground Systems B and C have no need for the aircraft’s application information at this time, so they both discard the information.
4) The aircraft desires FIS services from the server, so it establishes an FIS contract with the server. Note that if a central FIS server did not exist and instead there were several FIS locations, a ground

system that performs data link services with the aircraft would be responsible for giving the aircraft the address of the appropriate FIS location by using a CM Update function.

5) Ground System A needs to have data link services with the aircraft. It also wishes to give the aircraft the option to start a CPDLC dialogue. So Ground System A sends a CM update with its CPDLC application address, and starts an ADS contract. The aircraft starts a CPDLC dialogue with Ground System A based on the information contained in the CM Update.
6) Based on locally-held flight plan information, Ground System D is expecting the aircraft to approach its sector and wishes to start ADS services. Since it does not have ground-ground CM connectivity with the CM server, it must use another means to request the aircraft's application information. Ground System D sends a request to the CM server asking for the application information for the aircraft whose Flight ID and 24 bit address is contained in the locally-held flight plan.
7) The CM server responds to Ground System D with the requested application information using a non-CM method.
8) Now that Ground System D has the aircraft's ADS application address, it establishes an ADS contract.

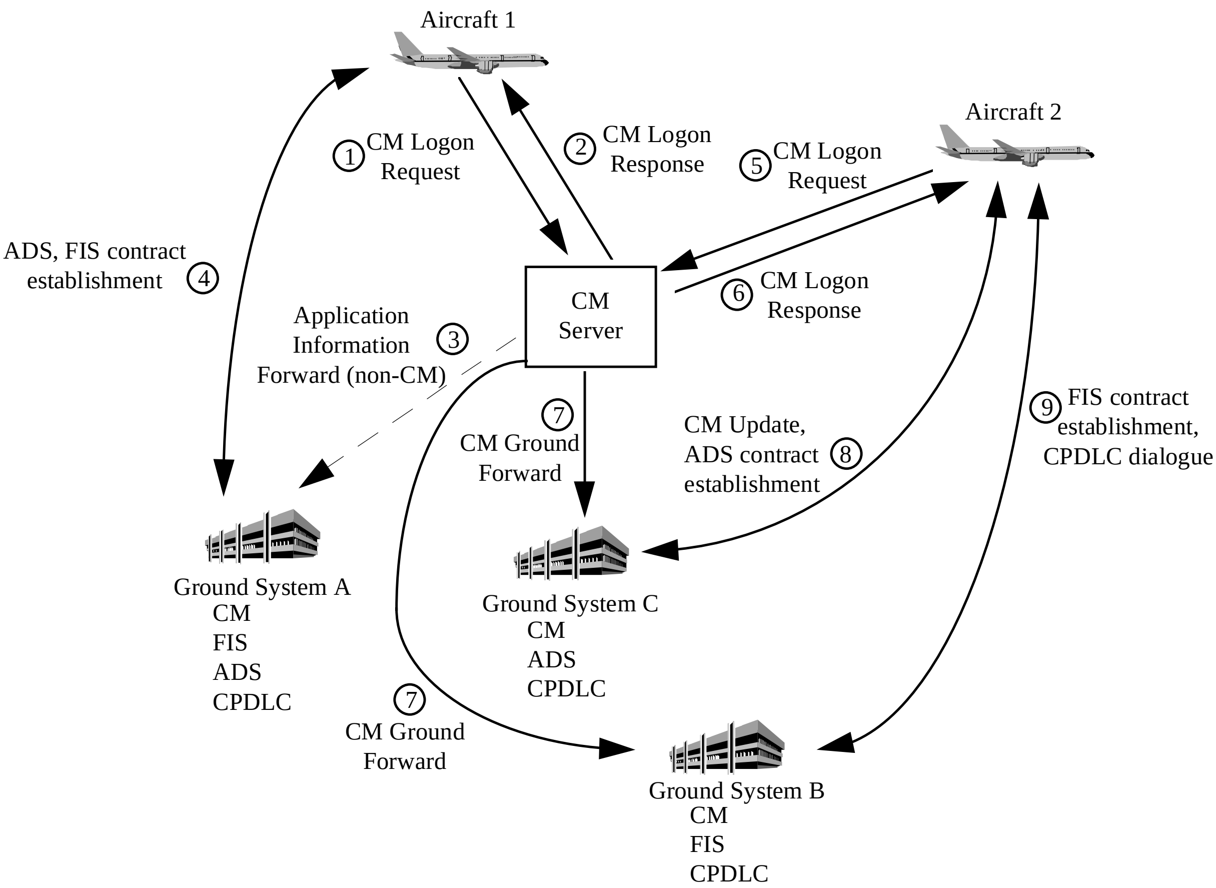

Figure 2.5. "Smart" CM Server Scenario

2.2.15 For the configuration depicted in Figure 2.5, the CM server has enough information to know what ground system an aircraft will need to perform data link services when it receives a CM logon from that aircraft. The ground systems can depend on the server giving the proper application addressing information to the aircraft and alerting the appropriate ground system that an aircraft has performed a logon. The ground systems support the applications as indicated. The numbers in Figure 2.5 are explained below:

1) Aircraft 1 sends a CM logon to the CM server.
2) The CM server examines Aircraft 1's logon information, and discerns that it will need to communicate initially with Ground System A. The CM server retrieves the application information for Ground System A from its local database, including the FIS and CPDLC addresses and version numbers and ADS version number, and replies to Aircraft 1 with this information contained in the logon response.
3) The CM server then needs to notify Ground System A that Aircraft 1 has its application information, and also needs to make sure that Ground System A has Aircraft 1's application information. Since the CM server knows that ground-ground CM connectivity does not exist between itself and Ground System A, the CM server uses a different means to pass Aircraft 1's application information to Ground System A.
4) Upon receipt of the Aircraft 1's application information, Ground System A establishes an ADS contract. Aircraft 1 then establishes a FIS contract.
5) Aircraft 2 sends a CM logon to the CM server.
6) The server discerns that Ground System B will need to communicate with Aircraft 2. The CM server replies with Ground System B's application information, including FIS and CPDLC addresses and version numbers.
7) The CM server then needs to notify Ground System B that Aircraft 2 has its application information, and also needs to make sure that Ground System B has Aircraft 2's application information. Accordingly, the CM server uses the CM ground forward function to pass Aircraft 2's application information to Ground System B. In addition, the CM server also realizes that Ground System C will need to monitor Aircraft 2 using ADS, so the CM server uses the CM ground forward function to pass Aircraft 2's application information to Ground System C as well.
8) Ground System C decides that it wants to give Aircraft 2 the option of starting a CPDLC dialogue. Using the information forwarded to it by the CM server, Ground System C sends a CM Update to Aircraft 2 with the appropriate application information and then establishes an ADS contract.
9) Ground System B uses the information forwarded to it by the CM server to establish a CPDLC dialogue. Aircraft 2 uses the FIS information given to it in the logon response to establish a FIS contract with Ground System B.

2.3 Abstract Internal Architecture

2.3.1 General

2.3.1.1 The architectural model for the CM Application conforms to the ULA SARPs (Ref [2]). One Application Entity (AE) is defined per CM Application. Each AE contains an ATN Application Service Element (ASE), which is the communication element responsible for the ATN Application. The CM SARPs specify the CM AE and CM ASE.

2.3.2 Types of ASE in CM

2.3.2.1 The CM application defines only one type of ASE - the CM-ASE. The CM-ASE has two variations, the CM-air-ASE and CM-ground-ASE. Additionally, the CM-ground-ASE has three functional modes, the CM-ground-ASE used for air-ground communications, the CM-ground-ASE used for initiating ground-ground communications, and the CM-ground-ASE used for responding ground-ground communications. The CM-air-ASE and CM-ground-ASE (air-ground) are used for air to ground communication, while the CM-ground-ASE (ground-ground initiator) and CM-ground-ASE (ground-ground responder) are used for the ground forward function. A CM-ground-ASE cannot act in two different modes at once; that is, a CM-ground-ASE cannot be an air-ground ASE and a ground-ground initiator ASE at the same time. Note, however, that a CM-ground-ASE that is a ground-ground responder is always independent of any other existing local CM-ground-ASE. The different modes of operation are depicted in Figures 2.6 and 2.7.

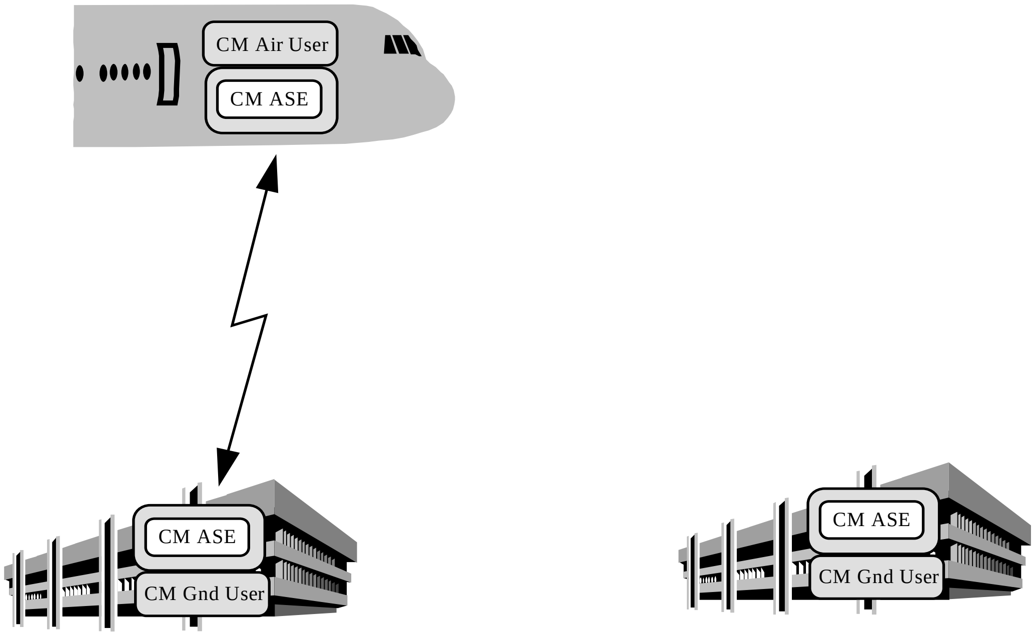

Figure 2.6. CM-ground-ASE in air-ground Mode

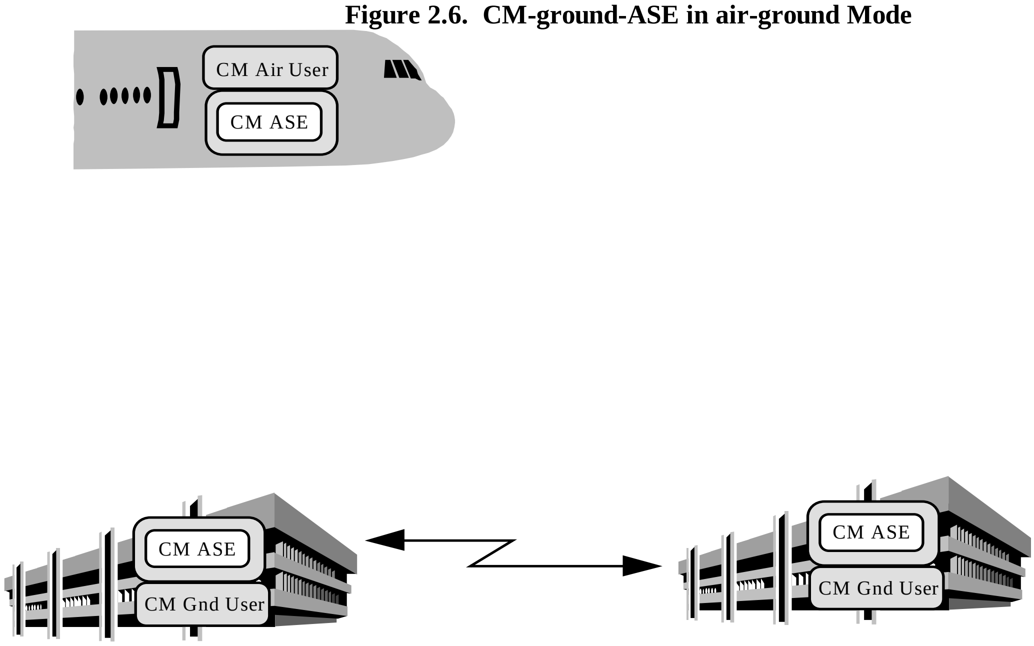

Figure 2.7. CM-ground-ASE in ground-ground Initiator and Responder Mode

2.3.3 CM-ASE Functionality

2.3.3.1 The CM-ASEs are responsible for all aspects of air-ground and ground-ground communication related to CM. They encode and decode the PDUs. They maintain a state machine that only allows PDUs to be sent at an appropriate time, and detect if the peer application sends PDUs at an inappropriate time.

2.3.3.2 There is no architectural capability for multiple instances of the CM-ASE within the same CM AE. This implies that the CM-ASE will generate and manage only one dialogue over the lifetime of the ASE

2.3.4 CM-User Functionality

2.3.4.1 In the model of CM in Figure 2.6, there is a module called the CM-User. The functionality of the CM-User is described in 2.1.7 of the CM SARPs. The CM-ground-user is responsible for initiating Update and Contact functions, responding to Logon requests, initiating and responding to Forward functions, providing ground system application information, making peer application and flight plan information available to the ground system as defined by the Registration function, dialogue maintenance, and air- and ground-ground dissemination of CM information. There are very few requirements placed on the CM-ground-user. This allows the implementers a great deal of freedom in the method of implementation of this component.

2.3.4.2 The CM-air-user is responsible for initiating the Logon function, both initially and when directed by a Contact function, replying to other requests from a ground system, providing aircraft application information and making peer application and flight plan information available to the end user as defined by the Registration function. Again, there is a great deal of freedom allowed in the method of implementation of this component.

2.3.5 Product Architecture

2.3.5.1 The SARPs have defined an abstract model for the purposes of definition. That is, it has split the functionality between the ASE and the user and has defined an abstract interface between the two. It is strongly emphasized that there is no requirement on an implementer to build such an interface. If it is convenient, from an engineering perspective, to build an interface between two modules that embody the functionality of the ASE and the user, then the implementer is free to do so. However, if it is more convenient to build the system with interfaces in other places, then that is also acceptable. In testing a product to see if it conforms to the SARPs, no test can be made to test internal interfaces within the system.

2.3.5.2 Also note that the internal structure of the ATN ASE is not standardised across applications; for example, CM is defined as a single module while FIS is defined as seven. The CM module handles the protocol for the CM application. If individual functions are not supported as allowed by the subsetting rules, the module will ensure that the protocol will handle the remaining subset of the full CM functionality.

2.3.5.3 The dialogue service interface is the ATN ASE's view of the ATN Upper Layer Architecture. The CM module uses the Dialogue Service for communication with the peer ASE through the ATN. The control function is used to for communication between different elements, such as the CM-ASE and CM-user.

2.4 Implementation Dependent Functionality

2.4.1 The CM SARPs specify some of the requirements for the user, but leave a lot up to the implementer. There are no requirements that state how the user interface appears, how CM interacts with the systems generating initial

CM address information, how CM interacts with higher level functionality and with other applications such as ADS, CPDLC or FIS. All of this is implementation dependent.

2.4.2 Another key aspect of implementation dependence is how the initial ground CM address is supplied. The logon function is initiated by the aircraft. It can be initiated manually by the pilot or automatically by the avionics. Regardless of how it is obtained, the initial address of the CM application in the ground facility needs to be made available to the airborne CM application. This is necessary in order for the airborne CM application to know the ground address it must connect with. There are a number of ways the initial address may be supplied. Among them are:

a) The avionics determines, based on the aircraft's position, what ground system would need to be logged on to and will perform a database or table lookup function in order to obtain the address,
b) The pilot entering the facility designation or short form address of the ground system, and an on-board mechanism (such as a database or table lookup function) completes the address,
c) A pre-set address in the aircraft of a ground-based CM address regional or global file server,
d) A ground system (e.g. airline or service provider) supplies a locally stored address to the aircraft, and the aircraft makes it available to the dialogue service provider, and
e) The pilot entering the full address as listed in an AIP or like document.

2.4.3 Note that the above methods are not meant to be an exhaustive list, but gives some ways that the address may be supplied.

2.5 Rationale for ASE/Users Split

2.5.1 The rationale for the split in functionality between the CM ASEs and the CM-users is as follows:

2.5.1.1 The ASE contains all functionality that is necessary to ensure the interoperability at the syntactic level. That is, two valid implementations of ASEs will be able to interact, passing data to each other in the correct order. They will be able to check the format of the data, ensure that it has been sent at an appropriate point in the dialogue and also ensure that the peer ASE is behaving according to the requirements in the SARPs. The ASE thus ensures interoperability.

2.5.1.2 The SARPs define some requirements for the users. These are the minimum user requirements necessary to ensure the semantic interoperability of the two peers. Thus the user requirements explain how each CM service is interpreted and how it should operate.

2.5.1.3 Some care has been taken to ensure that the requirements are not over-specified. That is, they do not specify rules which are not absolutely essential to the syntactic and semantic interoperability of the CM function. This implementation dependent part can be built by different manufacturers in different ways, without affecting the interoperability between different implementations. This implementation dependent part has not been specified in the SARPs, but should be specified by individual product manufacturers or regional standards.

2.6 Inter-relationship with other ATN Applications

2.6.1 The pre-requisite for the use of any data link application is the knowledge by the ground system or aircraft that is initiating the service of the ATN name and address of the contacted system application. The CM application is the most natural way to retrieve this addressing information. A CM-logon exchange is therefore required between the aircraft and the ground CM system administrating the ADS system. By passing the application information between the aircraft and the ground system and performing necessary address creation through the use of CM, the end systems are able to perform version negotiation within the end system itself. This makes efficient use of communication assets, as they are not required to perform version negotiation for each individual application. All of the application addresses obtained by CM will be made available to the local applications.

2.6.2 Support of CM is required if any data link applications are to be used (as per Sub-volume I). However, CM does not necessarily have to be involved in obtaining addresses or version negotiation of data link applications. If a ground system or aircraft has a priori knowledge of the application addresses and version numbers of data link applications it wishes to use in the peer system (e.g. from hard-coding the addressing data bases in the aircraft or the loading of this information at the gate before the take-off), then a CM logon does not have to occur in order to use the applications. In this case the ground system or aircraft commences use of the desired data link application using the appropriate addresses. A drawback of this method of data link use is that if there is an equipment change or software update, the data link addresses may no longer be valid and data link services will not be possible. This can have many safety issues (e.g. connections to the wrong aircraft, aircraft route changes that change the address profile, etc). Also note that since the dialogue service provider handles addressing, the data link applications' address must be made available to the dialogue service provider.

2.6.3 CPDLC DSC addresses

2.7 Distribution of CM Information

2.7.1 In order to support the CM operational concept, the CM application must be able to transfer data link application information to appropriate ATS facilities when needed. This function is achieved through the use of the CM-contact service and CM-forward service.

2.7.1.1 Ground-ground Distribution

2.7.1.1.1 Ground-ground distribution is facilitated by the CM-forward service. The CM-forward service is used to send the information received in a CM-logon message to an appropriate ATS facility. The receiving ATS facility can then use the received aircraft application information to perform a CM-update service in order to notify the aircraft of its data link application information.

2.7.1.1.2 The CM application does not provide a mechanism for an ATS facility to request an aircraft's application information from another ATS facility. If a ground facility wants to get an aircraft's application information forwarded to itself, it must use some other means to request the information.

2.7.1.2 Air-ground Distribution

2.7.1.2.1 Air-ground distribution can be facilitated by the use of the CM-contact service. Although this service is not an exact air-ground equivalent of the CM-forward service, it does provide a functional means to have an aircraft exchange application information to another ATS facility via a CM-logon service. This may be used in areas where ground-ground connectivity is not available, or if an ATS facility does not support the CM-forward service. Note that the aircraft may not actually perform the logon. In this case, the ATS facility that invoked the CM-contact service would get an indication from the aircraft that the contact was not successful. Another means must then be used to forward the application information.

2.7.2 The CM application does not provide a mechanism for an ATS facility to request an aircraft's application information from the aircraft itself. The ATS facility may use other means to instruct the aircraft to perform a CM-logon service in order to obtain the aircraft's application information (e.g. R/T).

2.8 Dialogue Management

2.8.1 A dialogue is a connection between the aircraft and the ground system for the purposes of CM interaction. Maintaining a dialogue refers to keeping a continual connection between the aircraft and ground system. Once a dialogue is maintained, it remains open until an end service or abort is initiated. Not maintaining a dialogue refers to opening a connection only for the duration of a single service and then closing the connection as soon as the service is complete. Not maintaining a dialogue is also referred to as a "one-shot" service.

2.8.2 Advantages of maintaining a dialogue are apparent where there may be many different sources of application information that need to be exchanged in a short period of time due to small FIRs. Keeping a connection open eliminates the constant setting up and tearing down of connections, which will speed transaction times.

2.8.3 In the case of a centralized CM server, maintaining a dialogue may also prove beneficial. If a dialogue is maintained between the CM server and a particular aircraft, the CM server can continually update the aircraft on new application information that it receives from ground systems using it. In this way, the aircraft can efficiently receive the new application information with a minimum of connection establishment/termination time delays. However, this will also result in additional complexity since there will need to be a ground forwarding (or some other means) of application information between the CM server and ground systems.

2.8.4 There are also advantages of not maintaining a dialogue. If there is only a need for a single exchange of application information as, for example, a logon, then there would be no need to leave a connection open and have to close it down later. In this case, not maintaining a dialogue is ideal--both aircraft and ground system receive the information they need and neither has to worry about closing the connection.

2.8.5 A CM ground implementer may choose to either support or not support the capability for the ground CM application implementation to maintain a dialogue. The CM air implementation must always allow for support of maintaining a dialogue.

2.8.6 If the maintain dialogue option is implemented, the CM-ground user has the option to either maintain or not maintain a dialogue when responding to a CM-logon service from the aircraft. Maintaining a dialogue may be required depending on the operating environment. If not required for a particular operational reason, the dialogue may not be maintained since the CM-logon service may be the only CM function performed between a particular aircraft and ground system.

2.8.7 If the dialogue is maintained, then the dialogue will remain open until the CM-ground-user invokes a CM-end service or an abort is sent or received.

2.8.8 The maintain dialogue function is only a capability of the CM-logon service. The CM-update and CM-contact services have no effect on whether or not a dialogue is maintained.

2.9 Protocol Monitoring

2.9.1 General

2.9.1.1 The generation and transmission of expected responses are monitored by the peer CM-ASE. If there is an unexpected event, the ASE will abort and give a reason (if possible). Likewise, if an application timer expires, the ASE will also abort. These cases are described in the following sections.

2.9.2 Service Timers

2.9.2.1 If the CM-logon response is not invoked by the CM-ground-user within a period of time locally specified in the aircraft (recommended as 4 minutes in 2.1.5.2.2 of the CM SARPs), the dialogue in place with the ground CM system is aborted. CM-users are informed of this situation by a CM-provider-abort indication received at both sides with a reason of “timer-expiry”.

2.9.2.2 If the CM-contact or CM-forward response is not invoked by the CM-air-user or CM-ground-user (respectively) within a period of time locally specified in the ground system (recommended as 8 minutes and 4 minutes, respectively, in 2.1.5.2.2 of the CM SARPs), the dialogue in place with the aircraft or ground CM system is aborted. CM-users are informed of this situation by a CM-provider-abort indication received at both sides with a reason of “timer-expiry”.

2.9.2.3 When a dialogue is to be closed, a D-END is initiated via a CM-end service. If a D-END confirmation is not received by the CM-ground-ASE within a period of time locally specified in the ground system (recommended as 4 minutes in 2.1.5.2.2 of the CM SARPs), the dialogue in place with the aircraft is aborted. CM-users are informed of this situation by a CM-provider-abort indication received at both sides with a reason of “timer-expiry”.

2.9.3 Errors

2.9.3.1 Unrecoverable System Error

2.9.3.1.1 The unrecoverable system error is intended to cover cases where a fault causes a system lockup or the system becomes unstable. Upon determining that it has become unstable, the CM application will attempt to abort with the reason “undefined-error”.

2.9.3.1.2 The unrecoverable system error processing is written as a recommendation in the SARPs instead of a requirement, as it is recognized that depending on the nature of the error in the system, it may not be possible to regain control in order to perform an abort.

2.9.3.2 Invalid PDU

2.9.3.2.1 An invalid PDU is defined as a PDU that cannot be decoded or that is not received when it is expected. For example, a PDU that somehow becomes garbled would be an invalid PDU, or a logon request received from an aircraft that did not have a Logon Request parameter would also be considered an invalid PDU. There are separate cases for handling invalid PDUs, one for the receipt of an Indication primitive (for D-START indication and D-DATA indication) and one for receipt of a Confirmation primitive (for D-START confirmation).

2.9.3.2.2 For the case of the D-START indication and D-DATA indication (2.1.5.4.3.1 in the CM SARPs), there is inherently a connection in place and if there is an invalid PDU then a D-ABORT must be invoked. For this case, the reason “invalid-PDU” is given in the abort, as well as the abstract value of “provider” for the *Originator* parameter value. The abstract value “provider” is used since this is not a user-invoked abort. Finally, if the CM-air-user or CM-ground-user is an active user (as defined in 2.1.5.3.2 Note 2 and 2.1.5.3.3 Note 2), then an indication that an abort due to an invalid PDU occurred is given.

2.9.3.2.3 For the case of the D-START confirmation (2.1.5.4.3.2 in the CM SARPs), there may or may not be a connection in place. If there is a connection in place (i.e. a dialogue is attempting to be started), then the D-START *Result* parameter will be set to accepted. In this case, a D-ABORT will be invoked with the abort reason "invalid-PDU" as well as the abstract value of "provider" for the *Originator* parameter value. The abstract value "provider" is used since this is not a user-invoked abort. If a connection is not in place, only the local active user (as defined in 2.1.5.3.2 Note 2 and 2.1.5.3.3 Note 2) need be informed that an abort due to an invalid PDU occurred. Note that the active user is notified regardless of whether or not a connection is in place.

2.9.3.3 Not Permitted PDU

2.9.3.3.1 A not permitted PDU is defined as a PDU that arrives when the CM-ASE is in a state that does not allow that PDU. For example, receiving a CMContactResponse from an aircraft when the ground system is in the *LOGON* state. This does not mean that the PDU cannot be decoded; that is an invalid PDU as described above. As for the invalid PDU, there are separate cases for handling not permitted PDUs, one for the receipt of an Indication primitive (for D-START indication and D-DATA indication) and one for receipt of a Confirmation primitive (for D-START confirmation).

2.9.3.3.2 For the case of the D-START indication and D-DATA indication (2.1.5.4.4.1 in the CM SARPs), there is inherently a connection in place and if there is an invalid PDU then a D-ABORT must be invoked. For this case, the reason "not-permitted-PDU" is given in the abort, as well as the abstract value of "provider" for the *Originator* parameter value. The abstract value "provider" is used since this is not a user-invoked abort. Finally, if the CM-air-user or CM-ground-user is an active user (as defined in 2.1.5.3.2 Note 2 and 2.1.5.3.3 Note 2), then an indication that an abort due to a not permitted PDU occurred is given.

2.9.3.3.3 For the case of the D-START confirmation (2.1.5.4.4.2 in the CM SARPs), there may or may not be a connection in place. If there is a connection in place (i.e. a dialogue is attempting to be started), then the D-START *Result* parameter will be set to accepted. In this case, a D-ABORT will be invoked with the abort reason "not-permitted-PDU" as well as the abstract value of "provider" for the *Originator* parameter value. The abstract value "provider" is used since this is not a user-invoked abort. If a connection is not in place, only the local active user (as defined in 2.1.5.3.2 Note 2 and 2.1.5.3.3 Note 2) need be informed that an abort due to an invalid PDU occurred. Note that the active user is notified regardless of whether or not a connection is in place.

2.9.3.4 D-START Confirmation Result or Reject Source Parameter Values Not as Expected

2.9.3.4.1 Only the CM-logon service provides the ability for the CM-ground-user to maintain a dialogue. The results in a D-START Confirmation received by the CM-air-ASE with the D-START *Result* parameter being set to the abstract value of "accepted". The CM-air-user never has the option to maintain a dialogue when responding to a CM-contact or CM-update service, nor does a CM-ground-user when using the CM-forward service. This is ensured by 2.1.5.4.5.1 in the CM SARPs, which checks D-START confirmations that are received by the CM-ground-ASE for the *Result* parameter being set to the abstract value of "accepted". If this is the case, then a D-ABORT is invoked with the reason "dialogue-acceptance-not-permitted" and the abstract value of "provider" for the *Originator* parameter value. The abstract value "provider" is used since this is not a user-invoked abort. If the CM-ground-user is an active user (as defined in 2.1.5.3.2 Note 2 and 2.1.5.3.3 Note 2) then an indication that an abort due to an illegal value of the D-START *Result* parameter occurred is given.

2.9.3.4.2 In the normal course of operation, CM-ASEs should not receive a D-START confirmation with the D-START *Result* parameter having the abstract value "rejected (transient)" nor D-START *Reject Source* parameter "DS provider". The D-START *Result* parameter is set by the CM-ASEs, and is normally either "rejected (permanent)" as in the case of a dialogue not being maintained or "accepted" for the case of a dialogue being maintained. The D-START *Reject Source* parameter is set by the dialogue service provider, and is normally "DS User". The D-START *Result* abstract value "rejected (transient)" or the D-START *Reject Source* abstract value "DS provider" is indicative of a communication failure, since they both specify that the dialogue was rejected by the provider. Therefore, an indication that an abort due to a communication service error is given to the CM-air-user or CM-ground-user, if active. A D-ABORT is not invoked since there is inherently no connection due to the communication error condition.

2.9.3.5 D-END Confirmation Not as Expected

2.9.3.5.1 When a CM-end service is requested by a CM-ground-user, a D-END request is invoked. Upon receipt of the corresponding D-END indication, the CM-air-ASE's only choice is to accept the D-END by setting the D-END response Result parameter to the abstract value "accepted". If for some reason this value is not present (as checked in 2.1.5.4.6.1 of the CM SARPs), the CM-ground-ASE will invoke a D-ABORT with the reason of "dialogue-end-not-accepted" and the abstract value of "provider" for the *Originator* parameter value. The abstract value "provider" is used since this is not a user-invoked abort. An indication is not given to the CM-ground-user, since the dialogue was already expected to end, regardless of whether it was graceful or not.

2.9.3.6 D-START Indication Quality of Service Parameter Not as Expected

2.9.3.6.1 The ATN Sub-volume I (Ref [5]) dictates the values used for application service priority and RER quality of service parameters for all ATN applications. These values must be adhered to so proper levels of flight safety and performance are maintained. For CM, the values used are "flight regularity communications" for application service priority and "low" for RER quality of service. If these values are not present (as checked in 2.1.5.4.7.1 of the CM SARPs), then the CM-ASE will abort with reason "invalid-QOS-parameter" and the abstract value of "provider" for the *Originator* parameter value. The abstract value "provider" is used since this is not a user-invoked abort.

2.10 Version Number Negotiation

2.10.1 The 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.

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

2.10.3 There are three possible cases for CM version number comparisons:

a) the aircraft's CM version number is greater than the ground's,
b) the aircraft's CM version number is less than the ground's, and
c) the aircraft's CM version number is equal to the ground's.

2.10.4 For the case of 2.10.3 a, 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 logon may be initiated.

2.10.5 For the case of 2.10.3 b, the CM applications on the ground is assumed to be backwards compatible with the CM application on the aircraft. The application information is accepted by the aircraft, and the lower-numbered 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).

2.10.6 For the case of 2.10.3 c, 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.

2.10.7 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.

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

2.11 CM Addressing Concerns

2.11.1 General

2.11.2 This section details some of the issues involving the use of application addresses. It gives insight into how addresses are constructed, and what address components are necessary depending on aircraft and ground system equipage, topology and implementation.

2.11.2.1 The CM service also provides a certain level of application address control and elementary security by limiting the knowledge of application addresses of the aircraft. Since only ATS facilities that are contacted by an aircraft, or those contacted by another ATS facility will have access to the aircraft's application addresses, only those ATS facilities will be capable of establishing application connections.

2.11.2.2 If new or replacement equipment or software is installed on an aircraft or within an ATS facility that affects the addressing of ATN applications, any resulting new addresses need to be made available to the CM user in order to perform data link operations. In some cases new equipment or software will have to acquire the previous addresses in order to perform data link operations.

2.11.2.3 The ATN is based on the 7-layer OSI reference model. The lower four layers provide the Internet Communications Service, as defined in Sub-volume V (Ref [4]). The upper three layers comprise the Upper Layer Communications Service, as defined in Sub-volume IV (Ref [2]).

2.11.2.4 The ATN uses the ISO Efficiency Enhancement option for the Presentation and Session layers. These protocols in essence make the Presentation and Session layers pass through, with the overhead of only two octets (one octet for each the Session and Presentation layer). Consequently, message overhead is greatly reduced. Another result of the Efficiency Enhancement option is that the Session selector and Presentation selector are not used. This means that the normal addressing functions of these layers are not used, so there is no upper layer addressing for the ATN. Consequently, all of the addressing is accomplished with the Transport Service Access Point (TSAP) address that is defined in Ref [4].

2.11.2.5 Application addresses on aircraft and within ATS facilities may not be long term. That is, the address of a particular application may be assigned as the aircraft powers up and remain in effect until the system restarts. After restart, new addresses may be assigned. This results from the assignment of the TSAP selector, which is administered on a local basis, and makes up part of an application's address. This reduces the possibility of an ATS facility storing the addresses for a particular aircraft and trying to establish an application connection at any time.

2.11.3 Address Creation

2.11.3.1 In order to provide the proper addresses for applications, the TSAP address must be constructed by the CM-user. The CM SARPs mention three different types of TSAP addresses, which are conventionally termed short TSAP, long TSAP, and actual TSAP. These are defined in Ref [4].

2.11.3.2 The type of TSAP used is dependent on the situation. The only types of TSAPs that are actually exchanged by the CM applications are the short and long TSAP. The short TSAP is used if the routing domain of each individual application is identical. By not sending redundant information (i.e. the routing domain information), bandwidth is saved. If the routing domain of each individual application is different, then the long TSAP (which specifies the routing domain parts) must be sent. Finally, regardless of the type of TSAP exchanged by CM, the actual TSAP must be constructed by either the CM-air-user or CM-ground-user as appropriate. Since there are parts of the actual TSAP that are identical for all ATN applications, there is no need to exchange them in CM services, and again, bandwidth is saved. However, in order to make use of the short or long TSAPs that are passed in CM services, these ATN-common parts along with the appropriate routing domain information (as appropriate) must be added by the users.

2.11.3.3 For the initial CM-logon service from the aircraft, the ground system's CM address needs to be known. This consists of the actual TSAP for the ground system's CM application. Possible ways of obtaining this information is detailed in 2.4.2.

2.11.3.4 Each type of TSAP is further detailed in the following sections.

2.11.3.4.1 Short TSAP

2.11.3.4.1.1 The short TSAP consists of the following fields:

a) the Administrative Region Selector (ARS) field, which is used to distinguish routing domains operated by the same State or Organisation,
b) the Location (LOC) field, which is used to distinguish routing areas within the same routing domain,
c) the System Identifier (SYS), which uniquely identifies the end system within the routing area
d) the Network Service Access Point (NSAP) Selector (NSEL) which identifies the End or Intermediate System network entity or network service user process responsible for originating or receiving Network Service Data Units (NSDUs), and
e) the TSAP Selector (TSEL), which identifies the address of the access point to the transport service.

2.11.3.4.1.2 The ARS field is used depending on the type of network addressing domain (i.e. fixed or mobile) and whether or not there are multiple routing domains in use. In the case of an aircraft (mobile network addressing domain), the ARS is mandatory as it identifies the aircraft on which the addressed system is located. For aircraft, the ARS takes the value of the aircraft 24 bit address. If more than one routing domain exists on board an aircraft then the LOC field is used to distinguish between them. In the case of the ground (fixed network addressing domain), the ARS field is optional. It need only be used if more than one routing domain exists on the ground. For ground systems, the ARS is set locally.

2.11.3.4.1.3 If the routing domain part (RDP) of the CM Long TSAP (described in 2.13.2.1.2) that is provided as part of the Logon Request parameter value is the same as an individual application's RDP, then the short TSAP may be used for that application. If the RDP of the CM Long TSAP is different than an individual application's RDP, than the long TSAP must be used for that application. This means that the short TSAP may be used for aircraft applications if there is only one routing domain on the aircraft.

2.11.3.4.2 Long TSAP

2.11.3.4.2.1 The long TSAP consists of:

a) the RDP, which specifies the routing domain of the application address, and
b) the short TSAP as defined above.

2.11.3.4.2.2 The RDP is further broken into the Version (VER), Administration (ADM), and Routing Domain Format (RDF) fields. The value of the VER field will represent either the Aeronautical Industry Service Communication (AINSC) or Air Traffic Service Communication (ATSC) organization which is responsible for the identified ATN network addressing sub-domain. The ADM value will contain the corresponding three letter representation of the responsible AINSC or ATSC organization. The RDF field is always set to a known value and is present for historical purposes.

2.11.3.4.2.3 If the RDP information is not the same for given applications (i.e. the same organization is not responsible for the ATN network addressing sub-domains), then the RDP needs to be passed for each application. This means that the long TSAP must be used and the short TSAP cannot be used since there will be no way to differentiate the routing domains.

2.11.3.4.2.4 The CM long TSAP address itself must be made available to users through published documentation. This address is needed in order to start the CM process, and cannot be discerned consistently from flight plan information and previous experience.

2.11.3.4.3 Actual TSAP

2.11.3.4.3.1 The actual TSAP consists of:

a) the Initial Domain Part (IDP), which is a value set by ISO to denote ATN applications, and
b) the long TSAP.

2.11.3.4.3.2 Since the IDP is a set value determined by ISO, it will always be the same for CNS/ATM applications and hence does not need to be sent between the aircraft and ground. It is used locally by the aircraft and ground in order to determine the full ATN address of available applications. Note that in the case of a short TSAP being passed, the long TSAP must first be created by combining the RDP that is common to the applications along with the short TSAP.

2.11.4 Flight Plan Correlation

2.11.4.1 There may be a requirement by certain organizations to perform flight plan correlation for every aircraft that is requesting data link services. Certainly, proper correlation of an aircraft to a flight plan ensures that the intended aircraft is being communicated with. However, the capability does exist for a ground system to accept any logon request. If the ground system does not have a flight plan for the aircraft requesting a logon, or the flight plan information from the aircraft does not correlate with the ground system's flight plan information appropriate to that aircraft, the ground system may not be capable of supplying information for a requested application.

2.11.4.2 CM provides the means to accomplish flight plan correlation by allowing for optional supplementary flight plan information to be exchanged by the CM-logon service. If a particular region requires supplemental flight plan information in a logon request from an aircraft in order to perform proper correlation, then local procedures must be implemented that make that optional information mandatory.

2.11.4.3 The filed flight plan of an aircraft will conform to the ICAO standard flight plan. The 24 bit aircraft address will be derived from the appropriate flight plan field.

2.11.4.4 When the CM-logon service is invoked, the aircraft will pass the 24 bit aircraft address. If the receiving ATS facility has the filed flight plan and the also has the flight plan data, then the aircraft sending the logon can be correlated with the proper correct flight plan.

2.11.4.5 Additional information of Flight Identifier, Departure and Destination Airports, Time and Date of Departure may be used to resolve potential multiple filed flight plans so that correlation may take place.

2.11.4.6 The above paragraphs point to a logical progression of flight plan correlation:

1) The aircraft is first attempted to be correlated by the 24 bit aircraft address.
2) If correlation with the 24 bit aircraft address is not possible, then the aircraft flight IDs can be examined.
3) If there are multiple flights with the same flight ID, then the departure and destination airports and/or the time and date of departure can be examined.
4) If these steps do not allow proper correlation, then other means must be employed.

3. FUNCTIONALITY OF SERVICES

3.1 Concepts

3.1.1 Users of the CM service are termed *CM-ground-user* and *CM-air-user* or just *CM-user* when it applies to both air and ground. The CM-user represents the operational part of the CM system. It is either the final end-user (e.g. a crew member or controller) or an automated system. The CM-user that initiates a CM air-ground or ground-ground service is termed the *calling* CM-user or *initiator*. The CM-user that the initiator is trying to contact is termed the *called* CM-user or *responder*.

3.1.2 This section describes first the information required by the ASEs from the CM-users. Then, it considers CM services in turn and provides an overview of the data flow within the ASE which handles the service primitives. The primitives are grouped according to the services they provide: performing a logon, update, contact, forward, ending a dialogue, or aborting a dialogue.

3.2 Service Parameters

3.2.1 Facility Designation

3.2.1.1 The four to eight character facility designation must be provided by the CM-user for dialogue service addressing purposes as well as for indication to the peer CM-user, and is used to identify either the called or calling ground component for a particular service. The called facility designation is used for dialogue service addressing purposes and the calling facility designation is used for indication to the peer CM-user.

3.2.1.2 Four characters are used to identify a particular facility, such as “ZYVR” for Vancouver. Up to eight characters may be used to identify a particular system within a facility; i.e. “ZYVRA123” may be the address for Vancouver en-route.

3.2.2 Aircraft Address

3.2.2.1 The 24 bit aircraft address must be provided by the CM-user for dialogue service addressing purposes as well as for indication to the peer CM-user, and is used to identify either the called or calling air component for a particular service. The called 24 bit aircraft address is used for dialogue service addressing purposes and the calling 24 bit aircraft address is used for indication to the peer CM-user.

3.2.3 Class Of Communication Service

3.2.3.1 The services CM-logon, CM-update, CM-contact and CM-forward each has an optional parameter called "Class of Communications Service". The Class of Communications Service is a means for the user to indicate the required performance in terms of end-to-end transit delay. Also note that the Class of Communication provided for CM has no relationship to the Class of Communication for other ATN applications.

3.2.3.2 The Class of Communications Service, if provided by the CM-air-user or CM-ground-user, is supplied to the Transport Service Provider (TSP) when the connection is established to help set up the connection. Values for these transit delays are given in 1.3.7, Note 2 of Sub-Volume I of the ATN SARPs. The Class of Communication is not guaranteed nor is a degradation of the provided class indicated to the users. It is the responsibility of the application to ensure that the transit delay is acceptable (i.e. by proper use of application timers).

3.2.3.3 If a connection is in place already when the CM-update or CM-contact service are used, then this parameter is not needed. If it is provided by the CM-ground-user when a connection is place, it is ignored.

3.2.3.4 The value of the Class of Communication requested by a CM-air-user or CM-ground-user for a dialogue is not transmitted nor indicated to the peer user.

3.2.3.5 There is no negotiation of the Class of Communication between air and ground CM-users. If the Class of Communication is not provided a default value will be used.

3.2.3.6 If the CM-air-user or CM-ground-user does not require a particular Class of Communication, the Class of Communication parameter does not need to be provided. In practice this means that the implementers are free to choose what to do; for example, a default or random value could be chosen, or some other means employed. If this is the case, it means that the Class of Communication is chosen by the dialogue service provider.

3.2.4 Version Number

3.2.4.1 The Version Number is used to indicate version number differences between peer CM ASEs. The version number is not supplied by the CM-user. This eliminates version number ambiguity and input errors.

3.2.4.2 If the CM Version Number of the sending CM-air-ASE or CM-ground-ASE is less than the receiving CM-ground-ASE Version Number, then the Version Number is indicated to the receiving CM-ground-user. This is to alert the receiving CM-ground-user that there may be features in later CM application versions that are not supported. Therefore, the receiving CM-ground-user must allow for these differences in order to permit proper operation of the CM application.

3.2.4.3 If the CM Version Number of the sending CM-air-ASE or CM-ground-ASE is greater than the receiving CM-ground-ASE Version Number, then the Version Number is provided to the sending CM-air-user or CM-ground-user in response. Since the CM versions are incompatible and further CM interaction cannot take place, there is no need for the receiving CM-ground-user to know what the sending CM ASE Version Number is. The Version Number of the receiving CM-ASE is provided to the sending CM-air-user or CM-ground-user so that a compatible CM ASE version may be tried, if available.

3.2.4.4 If the CM version numbers are equal, then the Version Number is not provided to either peer user. Since both CM ASE versions are fully compatible, there is no need to provide the CM version numbers to either user.

3.2.5 Logon Request

3.2.5.1 The Logon Request parameter is supplied by the CM-air-user and contains the application and addressing information of the aircraft along with optional flight plan information. It is used in the CM-logon service, and is indicated to the CM-ground-user when the air CM ASE version number is less than or equal to the ground CM ASE Version Number. It is used by the CM-ground-user to make the aircraft application and addressing information available to other applications within the ground system and to correlate the aircraft with the correct flight plan.

3.2.6 Logon Response

3.2.6.1 The Logon Response parameter is supplied by the CM-ground-user and contains the application and addressing information of the ground system. It is used in the CM-logon service, and is indicated to the CM-air-user when the air CM ASE version number is less than or equal to the ground CM ASE version number. It is used by the CM-air-user to make the ground system application and addressing information available to other applications within the aircraft.

3.2.7 Update Information

3.2.7.1 The Update Information parameter is supplied by the CM-ground-user and contains the application and addressing information that will be used to update the application and addressing information held by the aircraft. It is used in the CM-update service, and is always indicated to the CM-air-user. The CM-air-user replaces the ground system application and addressing information that it currently holds with the information contained in the Update Information parameter.

3.2.8 Contact Request

3.2.8.1 The Contact Request parameter is supplied by the CM-ground-user and contains the facility designation and address of the ground system that the CM-ground-user requests the aircraft to contact. It is used in the CM-contact service, and is always indicated to the CM-air-user. The CM-air-user uses the information contained in the Contact Request parameter to determine the facility that it needs to perform a logon to.

3.2.9 Contact Response

3.2.9.1 The Contact Response parameter is supplied by the CM-air-user and contains the success, or lack thereof, of the contact request. It is used in the CM-contact service, and is always indicated to the CM-ground-user. The CM-ground-user will determine if the directed contact was successful by the information in the Contact Response parameter.

3.2.10 Forward Request

3.2.10.1 The Forward Request parameter is supplied by the sending CM-ground-user and contains an aircraft's application and addressing information. It is used in a CM-forward service and is indicated to the receiving ground user when the sending ground CM ASE version number is less than or equal to the receiving ground CM ASE version number. It is used by the receiving CM-ground-user to make the aircraft application and addressing information available to other applications within the ground system and to correlate the aircraft with the correct flight plan.

3.2.11 Maintain Dialogue

3.2.11.1 The Maintain Dialogue parameter is used to indicate whether or not a connection is to remain in place after a CM-logon service is completed. It may only be provided by the CM-ground-user, and only during a CM-logon service.

3.2.11.2 The Maintain Dialogue parameter may only be used if the ground CM implementation supports the maintain dialogue feature.

3.2.11.3 The Maintain Dialogue parameter is only used by the CM-ground-user when the CM-ground-user requires a CM dialogue to remain open.

3.2.11.4 If the CM-ground-user wants to maintain a dialogue when responding to a logon request from an aircraft, the CM-ground-user will set the maintain dialogue parameter to indicate that the dialogue is to be maintained.

3.2.11.5 If the CM-ground-user does not want to maintain a dialogue, the maintain dialogue parameter will not be provided.

3.2.12 Result

3.2.12.1 The Result parameter is supplied by the receiving CM-ground-ASE and is used to indicate success, or lack thereof, of the forward request. It is used in the CM-forward service, and is always indicated to the sending CM-ground-user of the forward request.

3.2.13 Reason

3.2.13.1 The Reason parameter identifies the reason for the CM-provider-abort. The Reason parameter is not used in the CM-user-abort. It is always indicated to the CM users.

3.3 CM Logon Service Specifics

3.3.1 The CM-logon service provides a means for an aircraft to exchange application information with a ground system. It is a confirmed service, and is initiated only by the CM-air-user.

- The CM-logon service request is passed to the CM-air-ASE by the CM-air-user
- The CM-air-ASE:
    - Creates a Logon Request APDU based on CM-air-user provided data
    - Invokes a D-START request to pass:
        - The Logon Request APDU as User Data,
        - The Called (Facility Designation) and Calling (Aircraft Address) Peer IDs as supplied by the CM-air-user,
        - The CM-air-ASE version number as supplied by the CM-air-ASE as DS User Version, and
        - The QOS parameters, including the Class of Communication if provided, as Quality of Service
    - Starts the logon timer
- Upon receipt of the D-START indication, the CM-ground-ASE:
    - Confirms that the D-START contains a valid Logon Request APDU,
    - Confirms the value of the Calling Peer ID and QOS parameters,
    - Checks the version number of the CM-air-ASE:
        - If the CM-air-ASE version number is greater than the CM-ground-ASE's, a D-START response is invoked with the CM-ground-ASE version number as the DS User Version and "rejected (permanent)" as the Result
        - If the CM-air-ASE version number is less than the CM-ground-ASE's, a CM-logon service indication is invoked with the following information: Calling Peer ID (Aircraft Address), CM-air-ASE version number, and User Data (Logon Request)

        - If the version numbers are equal, a CM-logon service indication is invoked with the following information: Calling Peer ID (Aircraft Address) and User Data (Logon Request)
- The CM-logon service response is passed to the CM-ground-ASE by the CM-ground-user
- The CM-ground-ASE:
    - Creates a Logon Response APDU based on the CM-ground-user provided data
    - Invokes a D-START response to pass:
        - The Logon Response APDU as User Data,
        - The Maintain Dialogue parameter:
            - if provided by the CM-ground-user, the abstract value “accepted” as the Result,
            - if not provided by the CM-ground-user, the abstract value “rejected (permanent)” as the Result
- Upon receipt of the D-START confirmation, the CM-air-ASE:
- Confirms that the D-START contains a valid Logon Response APDU,
- Stops the logon timer,
- Checks the Result to see if a dialogue is to be maintained:
    - If a dialogue is not maintained (i.e. the Result has the abstract value “rejected (permanent)”), the CM-air-ASE checks the DS User Version (CM-ground-ASE version number),
    - The CM-air-ASE invokes a CM-logon service confirmation with the following data:
        - if the CM-air-ASE version number is greater than the CM-ground-ASE’s, the DS User Version (CM-ground-ASE version number),
        - if the CM-air-ASE version number is equal to the CM-ground-ASE’s, the User Data (Logon Response)
    - If a dialogue is maintained (i.e. the Result is “accepted”), a CM-logon service confirmation is invoked with the following data:
        - User Data (Logon Response), and
        - Result (Maintain Dialogue)

3.4 CM Update

3.4.1 The CM Update function provides a means for a CM ground user to send updated ground application and addressing information to an aircraft. It is an unconfirmed service, and is only initiated by the CM-ground-user.

- The CM-update service request is passed to the CM-ground-ASE by the CM-ground-user
- The CM-ground-ASE checks if a dialogue is in place or not:
    - If a dialogue is not in place, the CM-ground-ASE:
        - Creates an Update Information APDU based on the CM-ground-user provided data,
        - Invokes a D-START request with the following:
            - The Update Information APDU as User Data,
            - The Called (Aircraft Address) and Calling (Facility Designation) Peer IDs as supplied by the CM-ground-user,
            - The QOS parameters, including the Class of Communication if provided, as Quality of Service, and
        - Starts the update timer
    - If a dialogue is in place, the CM-ground-ASE:
        - Creates an Update Information APDU based on the CM-ground-user provided data,
        - Invokes a D-DATA request with the following:
            - The Update Information APDU as User Data.
- The Update Information will arrive at the CM-air-ASE in either a D-START indication (if a dialogue is not in place) or a D-DATA indication (if a dialogue is in place):

  - If the CM-air-ASE receives a D-START indication with the User Data containing an Update Information APDU, it will:
    - Check to make sure that the QoS parameters are correct,
    - Invoke a CM-update service indication with the following information:
      - The Calling Peer ID as the Facility Designation parameter value,
      - The User Data as the Update Information parameter value, and
    - Invoke a D-START response with the abstract value "rejected (permanent)" as the Result
  - If the CM-air-ASE receives a D-DATA indication with the User Data containing an Update Information APDU, it will:
    - Invoke a CM-update service indication with the following information:
      - The User Data as the Update Information parameter value.
- Upon receipt of a D-START confirmation, the CM-ground-ASE will:
  - Stop the update timer,
  - Check to see that the Result has the abstract value "rejected (permanent)" and that the Reject Source has the abstract value "DS User".

3.5 CM Contact

3.5.1 The CM-contact service allows a ground system to request that an aircraft logon to another ground system. It is a confirmed service, and is only initiated by the CM-ground-user.

- The CM-contact service request is passed to the CM-ground-ASE by the CM-ground-user
- The CM-ground-ASE checks if a dialogue is in place or not:
  - If a dialogue is not in place, the CM-ground-ASE:
    - Creates a Contact Request APDU based on the CM-ground-user provided data,
    - Invokes a D-START request with the following:
      - The Contact Request APDU as User Data,
      - The Called (Aircraft Address) and Calling (Facility Designation) Peer IDs as supplied by the CM-ground-user,
      - The QOS parameters, including the Class of Communication if provided, as Quality of Service, and
    - Starts the contact timer
  - If a dialogue is in place, the CM-ground-ASE:
    - Creates a Contact Request APDU based on the CM-ground-user provided data,
    - Invokes a D-DATA request with the following:
      - The Contact Request APDU as User Data
- The Contact Request will arrive at the CM-air-ASE in either a D-START indication (if a dialogue is not in place) or a D-DATA indication (if a dialogue is in place):
  - If the CM-air-ASE receives a D-START indication with the User Data containing a Contact Request APDU, it will:
    - Check to make sure that the QoS parameters are correct,
    - Invoke a CM-contact service indication with the following information:
      - The Calling Peer ID as the Facility Designation parameter value, and
      - The User Data as the Contact Request parameter value, and
  - If the CM-air-ASE receives a D-DATA indication with the User Data containing a Contact Request APDU, it will:
    - Invoke a CM-contact service indication with the following information:
      - The User Data as the Contact Request parameter value.

- The CM-air-user will then perform a CM-logon service with the facility designation as identified in the Contact Request parameter. Upon completion of the CM-logon service, the CM-air user will invoke a CM-contact service response
- Upon receipt of the CM-contact service response from the CM-air-user, the CM-air-ASE will check to see if a dialogue exists or not:
    - If a dialogue is not in place, the CM-air-ASE:
        - Creates a Contact Response APDU based on CM-air-user provided data,
        - Invokes a D-START response with the following information:
            - The abstract value "rejected (permanent)" as the Result parameter,
            - The Contact Response APDU as the User Data parameter
    - If a dialogue is in place, the CM-air-ASE:
        - Creates a Contact Response APDU based on CM-air-user provided data,
        - Invokes a D-DATA request with the following information:
            - The Contact Response APDU as the User Data parameter
- The CM-ground-ASE will receive either a D-START confirmation (if a dialogue is not in place) or a D-DATA indication (if a dialogue is in place) containing the results of the contact.
    - Upon receipt of a D-START confirmation, the CM-ground-ASE will:
        - Check the User Data to ensure it contains a Contact Response APDU,
        - Stop the contact timer,
        - Check to see that the Result has the abstract value "rejected (permanent)" and that the Reject Source has the abstract value "DS User", and
        - Invoke a CM-contact service confirmation with the User Data as the Contact Response parameter value.
    - Upon receipt of a D-DATA indication, the CM-ground-ASE will:
        - Check the User Data to ensure it contains a Contact Response APDU, and
        - Invoke a CM-contact service confirmation with the User Data as the Contact Response parameter value.

3.6 CM Forward

3.6.1 The CM-forward service allows a ground system to forward aircraft application data to another ground system. It is a confirmed service and is always initiated by a CM-ground-user that supports the CM-forward service.

- The CM-forward service request is passed to the sending CM-ground-ASE by the sending CM-ground-user
- The CM-ground-ASE:
    - Creates a Forward Request APDU based on sending CM-ground-user provided data
    - Invokes a D-START request to pass:
        - The Forward Request APDU as User Data,
        - The Called (Receiving Facility Designation) and Calling (Sending Facility Designation) Peer IDs as supplied by the CM-ground-user,
        - The sending CM-ground-ASE version number as supplied by the sending CM-ground-ASE as DS User Version, and
        - The QOS parameters, including the Class of Communication if provided, as Quality of Service
    - Starts the forward timer
- Upon receipt of the D-START indication, the receiving CM-ground-ASE:
    - Confirms that the D-START contains a valid Forward Request,
    - Confirms the value of the Calling Peer ID and QOS parameters,
    - Checks to see if it supports the CM-forward service:

        - If the receiving CM-ground-ASE does not support the CM-forward service, the receiving CM-ground-ASE:
            - Creates a service-not-supported APDU,
            - Invokes a D-START response with the following information:
                - The service-not-supported APDU as the User Data,
                - The abstract value “rejected (permanent)” as the Result parameter,
        - If the receiving CM-ground-ASE does support the CM-forward service, the receiving CM-ground-ASE:
            - Checks the version number of the sending CM-ground-ASE:
                - If the sending CM-ground-ASE version number is greater than the receiving CM-ground-ASE’s, the receiving CM-ground-ASE will create a incompatible-version APDU and invoke a D-START response with the following:
                    - The incompatible-version APDU as User Data,
                    - The receiving CM-ground-ASE version number as the DS User Version, and
                    - The abstract value “rejected (permanent)” as the Result
                - If the sending CM-ground-ASE version number is less than the receiving CM-ground-ASE’s, a CM-forward service indication as well as a D-START response is invoked.
                    - The CM-forward service indication contains the following:
                        - Calling Peer ID (Sending Facility Designation),
                        - DS User Version (sending CM-ground-ASE version number), and
                        - User Data (Forward Request)
                    - The receiving CM-ground-ASE creates a success APDU and then invokes a D-START response containing the following:
                        - The success APDU as User Data, and
                        - The abstract value “rejected (permanent)” as the Result
                - If the version numbers are equal, a CM-forward service indication as well as a D-START response is invoked.
                    - The CM-forward service indication contains the following:
                        - Calling Peer ID (Sending Facility Designation), and
                        - User Data (Forward Request)
                    - The receiving CM-ground-ASE creates a success APDU and then invokes a D-START response containing the following:
                        - The success APDU as User Data, and
                        - The abstract value “rejected (permanent)” as the Result
- Upon receipt of the D-START confirmation, the sending CM-ground-ASE:
    - Confirms that the D-START contains a valid Forward Response APDU and that the Result has the abstract value “rejected (permanent)” and the Reject Source has the abstract value “DS user” ,
    - Stops the forward timer,
    - Checks the User Data to see the results of the forwarding:
        - If the User Data has the abstract value “service-not-supported”, a CM-forward service confirmation is invoked with the Result parameter set to the abstract value of “service-not-supported”,
        - If the User Data has the abstract value “incompatible-version”, a CM-forward service confirmation is invoked with:
            - The Result parameter set to the abstract value of “incompatible-version”,

        - The DS User Version Number parameter set to the CM-ASE Version Number parameter, and
    - If the User Data has the abstract value “success”, a CM-forward service confirmation is invoked with the Result parameter set to the abstract value of “success”.

3.7 CM End

3.7.1 The CM-end service provides the capability for a ground system to terminate a CM dialogue. This service is only used if a dialogue is in place. It is an unconfirmed service, and is only initiated by a CM-ground-user.

- The CM-end service request is passed to the CM-ground-ASE by the CM-ground-user
- The CM-ground-ASE:
    - Invokes a D-END request, and
    - starts the end timer.
- Upon receipt of a D-END indication, the CM-air-ASE:
    - Invokes a CM-end service indication, and
    - Invokes a D-END response with the D-END Result parameter set to the abstract value “accepted”
- Upon receipt of a D-END confirmation, the CM-ground-ASE:
    - Checks to see if the Result parameter has the abstract value “accepted”, and
    - Stops the end timer.

3.8 CM User Abort

3.8.1 The CM-user-abort service provides the capability for an aircraft or ground system to abort communication with its peer. It is an unconfirmed service, and may be invoked at any time by either a CM-air-user or a CM-ground-user.

- The CM-user-abort request is passed to either the CM-air-ASE or CM-ground-ASE by the CM-air-user or CM-ground-user, respectively.
- Upon receipt of a CM-user-abort request, the CM-air-ASE or CM-ground-ASE (as appropriate):
    - Checks to make sure the ASE is active,
    - Stops any timers that are set, and
    - Invokes a D-ABORT request with the D-ABORT Originator parameter set to the abstract value “user”
- Upon receipt of a D-ABORT indication, the CM-air-ASE or CM-ground-ASE (as appropriate):
    - Checks to make sure the ASE is active,
    - Stops any timers that are set, and
    - If the CM-air-user or CM-ground user (as appropriate) is an active user, the CM-air-ASE or CM-ground-ASE:
        - Invokes a CM-user-abort service indication, if the D-ABORT Originator parameter contains the abstract value “user”, or
        - Invokes a CM-provider-abort service indication with the APDU contained in the D-ABORT User Data parameter as the CM-provider-abort service Reason parameter value. Note that neither a CM-air-user nor CM-ground-user may put an abort reason in the D-ABORT.

3.9 CM Provider Abort

3.9.1 The CM-provider-abort service provides the capability for the CM service provider to inform its users that it can no longer provide the CM service. This service is only invoked by the CM service provider.

- Upon receipt of a D-P-ABORT indication, the CM-air-ASE or CM-ground-ASE (as appropriate):

- Checks to make sure the ASE is active, and
- Invokes a CM-provider-abort service indication with the CM-provider-abort Reason parameter set to the abstract value “communication-service-failure”.

3.10 CM Registration

3.10.1 CM registration is used to make the addresses that are exchanged through other CM services available to other applications in the aircraft or ground system and the dialogue service provider. There are no messages exchanged for the CM registration service. This is a local implementation issue.

4. CM SECTION DESCRIPTION

4.1 Section 2.1.2: GENERAL REQUIREMENTS

4.1.1 Version Number

4.1.1.1 This section is included to allow the exchange of version numbers of the CM application so that version negotiation may take place and future versions of the protocol negotiated.

4.1.2 Error Processing Requirements

4.1.2.1 In the abstract service definition, each service has a set of parameters and the abstract syntax of those parameters specified. Thus information which is not a valid syntax is not allowed to be input.

4.1.2.2 In the protocol description, it is not permitted to call a service when in an inappropriate state. Thus making use of the abstract services is not permitted at these times, i.e. a contact service cannot be invoked during the LOGON state.

4.1.2.3 An implementation should not allow the user to take such invalid actions; however, there is no requirement to prevent an implementation from allowing this. The error processing requirements section thus says that if the implementation allows the user to enter invalid information, the system must inform the user that an entry error has occurred. In that case, the error is locally detected the dialogue does not need to be aborted.

4.2 Chapter 2.1.3: ABSTRACT SERVICE DEFINITION

4.2.1 The Concept of an Abstract Service

4.2.1.1 Section 2.1.3 concerns the CM abstract service. The following paragraphs provide an explanation of “abstract service”.

4.2.1.2 In order to define the CM ASE (i.e. the part of the abstract service provider that contains the protocol machine – see section 2.1.5 of the CM SARPs), it is necessary to describe its reactions to both PDUs arriving from the peer application, and the inputs from the user. The PDUs are well defined in the protocol. The actions of the user, however, are not. The SARPs do not attempt to dictate the actions of the user except where absolutely necessary. Despite this, in order to define the ASE it is necessary to have a clear definition of user actions.

4.2.1.3 In order to get around this conundrum, an “abstract service” is defined. An abstract service is a textual description of the interactions between the user and the ASE. These interactions are precisely defined in section 2.1.3 of the CM SARPs. Having this definition allows the ASE to be specified precisely in terms of its reactions to the arrival of PDUs and the invocation of the services by the user. This, therefore, is the reason for defining an abstract service interface.

4.2.1.4 The abstract service interface is defined as being an interface between the ASE and the "user-part" of the software. These are known as the CM ASE and the ADS user. The ADS user is *not* generally the human user; it is that part of the system that uses the CM ASE.

4.2.1.5 If one was to buy an CM application, one would be buying a suite of executable code. From the code itself it is impossible to know whether or not the abstract service interface has actually been implemented. Therefore the CM SARPs do not require that the abstract service interface has to be built. It only requires that, when one examines it from an external point of view, it behaves in the same way as if it had been built. This is the explanation of statement 2.1.3.1.1 in the SARPs.

4.2.1.6 Thus the implementers may choose to build an CM application with a real internal interface that corresponds to the abstract service interface, or they may choose not to – it is entirely up to them. However, it should be realized that there are a number of good reasons why one might not want to build a system with an interface exactly like the abstract service interface. Examples include:

- There may be a more efficient way of building the software.
- The abstract service interface does not include parameters that are needed locally, but do not affect the state machine; for example, a real interface might include an indication of which aircraft an ADS report has come from.
- It may not be easy to build the abstract service from the development tools that are being used.
- The abstract service interface does not have any programming language bindings. A real interface would require an interface defined in a particular programming language.

4.2.1.7 Implementation of the abstract service interface is not mandated by the CM SARPs. The requirements for CM set out by ICAO are limited in scope – they are designed only to ensure interoperability between air and ground systems, and to ensure that they meet the stated functionality requirements. The CM SARPs do not specify the nature of any internal interface within the software, nor do they specify the human interface. Individual implementations need to define their own internal interfaces to suit their own requirements.

4.2.1.8 In summary, an abstract service interface is defined in the SARPs in order to be able to define the ASE protocol machine. It does not have to be built in any implementation, and there are several good reasons why it should not be implemented exactly as defined. A real implementation of the CM SARPs would normally be expected to define its own internal interfaces.

4.2.1.9 The CM application abstract service consists of eleven functions listed in 2.1.3.2.1.

4.2.2 Functional Model of the CM Application

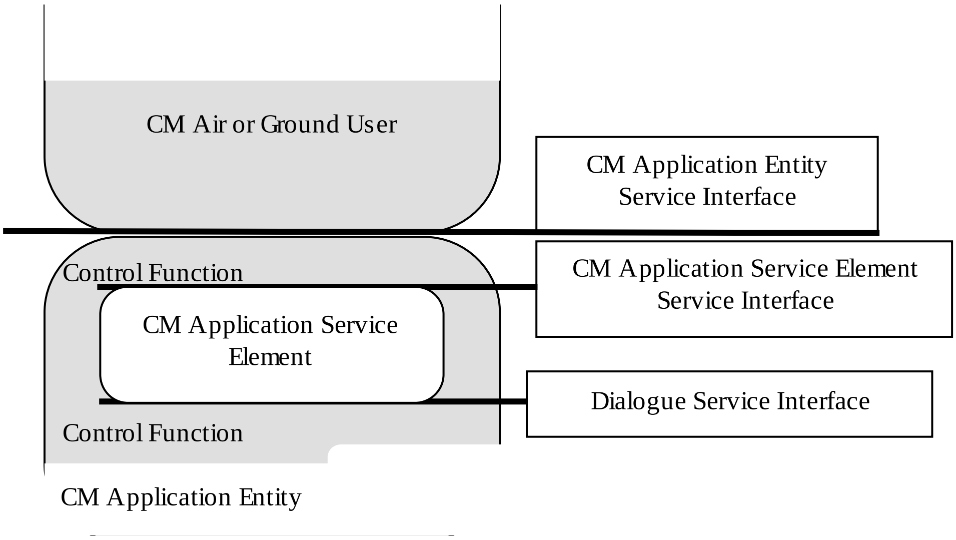

Figure 2.8. Functional Model of the CM Application

4.2.2.1 Figure 2.8 shows an abstract model for the CM application. This is the same as Figure 2.1.3-1 from the CM SARPs. Just as with the abstract service, this model shows a design of the CM application, breaking it down into modules. However, there is no requirement that an implementation actually builds it this way. The figure is presented here in order to explain the terms that are used throughout the document. It is not required that the design of an implementation follows this structure.

4.2.2.2 The figure shows three modules:

- the CM user (which could be a CM-air-user or CM-ground-user),
- the control function, and
- the CM ASE (application service element - which could be a CM-air-ASE or CM-ground-ASE).

4.2.2.3 In addition, it defines the CM application entity as the control function together with the CM ASE.

4.2.2.4 Abstract interfaces are shown between the different modules:

- the CM application entity service interface – which is the same as the abstract service interface defined in 2.1.3,
- the CM application service element service interface – which is also the same as the abstract service interface defined in 2.1.3, and
- the dialogue service interface – which is defined in the upper layer architecture SARPs (Ref [2]), and is identical for all air/ground applications.

4.2.2.5 Since the CM application entity service interface is identical with the CM application service element service interface, the control function module passes calls directly from one to the other without interference.

4.2.3 Conventions

4.2.3.1 Service Primitives

4.2.3.1.1 The CM SARPs defines seven services (CM-logon, CM-update, CM-contact, CM-end, CM-forward, CM-user-abort, and CM-provider-abort). Each primitive consists of the name of the CM service and a suffix that indicates at what point in the service the primitive occurs (request, indication, response, confirmation), e.g. CM-logon request. The primitives are further explained below:

- the CM user that initiates the service calls on the CM ASE to perform an action – this is called the "request"
- after the request is passed to the CM ASE on the other side of the communications link, it uses the service to pass the information on to its CM user – this is called the "indication"
- the CM user that has received the indication may choose to respond to it, in which case it calls upon its CM ASE to send a reply – this is called the "response"
- finally, the CM ASE receiving the response provides its CM user (which started the sequence of events) with the information – this is called the "confirmation".

4.2.3.1.2 The terms "request", "indication", "response" and "confirmation" are well understood in the field of communications protocols. A given CM service need not use all four primitives. Some CM services make use of one (indication), some two (request and indication or request and confirmation), some three (request, indication and confirmation) and some all four (request, indication, response and confirmation). These primitives are illustrated in Figure 2.9.

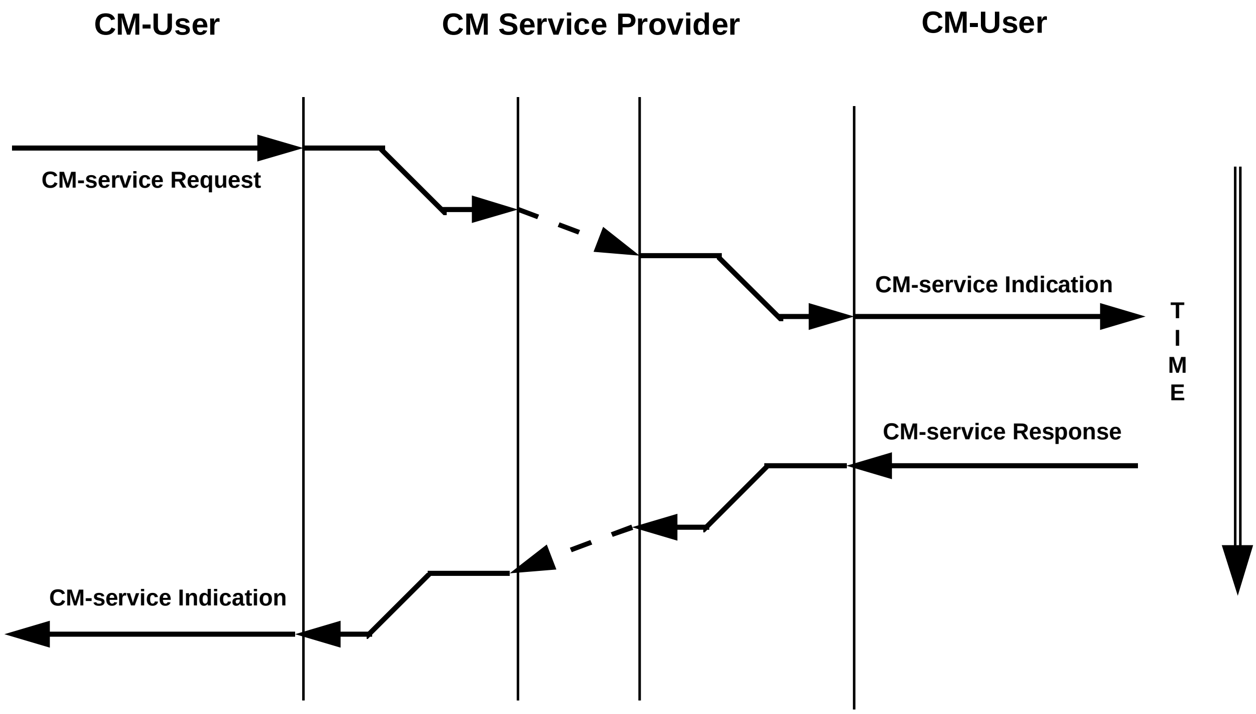

Figure 2.9. Generic CM-User Primitives

4.2.3.1.3 Each user-initiated service is also classified as being confirmed or unconfirmed. A confirmed service involves the return of an indication to the initiating user. Figure 2.9 is an example of a confirmed service. An unconfirmed service does not have an indication returned to the initiating user. Figure 2.10 illustrates an unconfirmed service.

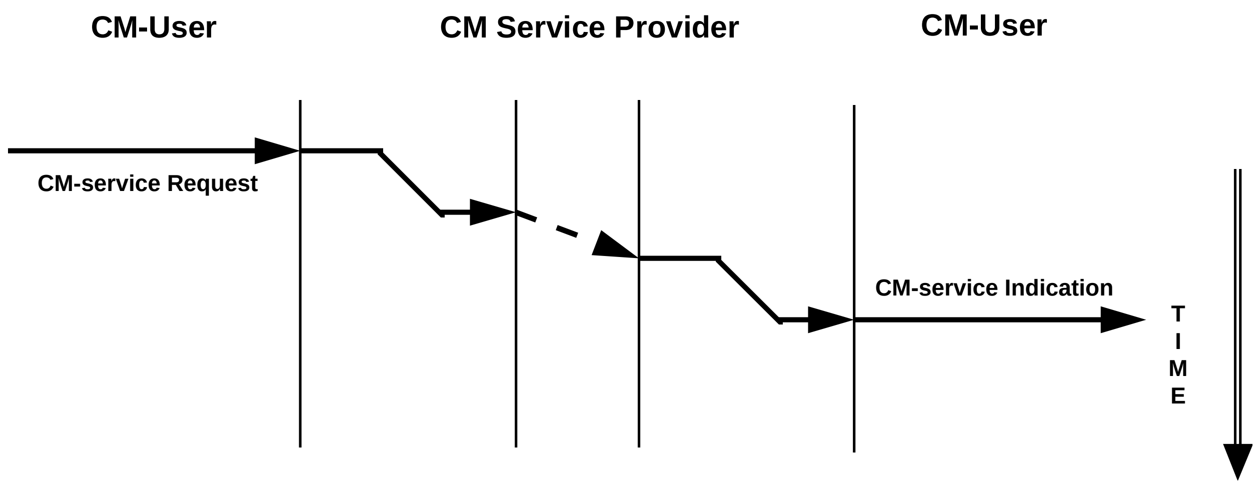

Figure 2.10. Generic Unconfirmed CM Service

4.2.3.1.4 For a provider-initiated service, an indication primitive is given to both CM-users. The provider-initiated service is generated by the service provider in response to an internal condition. Figure 2.11 illustrates a provider-initiated service.

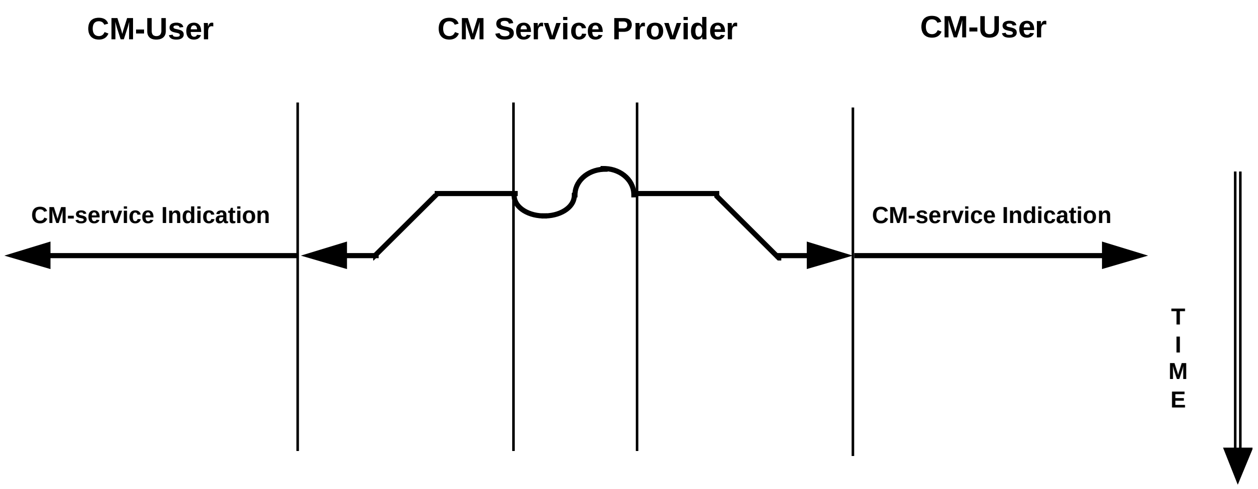

Figure 2.11. CM Service Provider-initiated Service

4.2.3.2 Detailed Service Descriptions

4.2.3.2.1 Each of the detailed service descriptions is defined in the same way. Firstly there is either one or two tables indicating the parameters of the service and their status in each of the primitives. Secondly, each of the parameters has a short description, and a statement of what the abstract syntax of the parameter is. This is further described in the following sections.

4.2.3.3 Service Primitive and Parameters

4.2.3.3.1 Each service has a set of parameters. (You may choose to think of the service as a procedure or function calls in a programming language.) The parameters used in the request, indication, response and confirmation are different. (Thus you may choose to think of them as four different, but related, procedure calls.)

4.2.3.3.2 The services are depicted in the SARPs by primitive and parameter tables. Not all services require all primitives to be used. That is, if a particular primitive is not needed due to reasons like redundant information being relayed, then the parameter column for that primitive is omitted. If a parameter column for a primitive is present, but all of the parameters are left blank, that means that the primitive is used by the service but does not carry any user-provided data.

4.2.3.3.3 It should also be noted that not all parameters need be present for all indications of a particular primitive.

4.2.3.3.4 For a specific primitive, each parameter is described by a value that dictates the terms under which that parameter is used. If the use of any parameter does not follow the rules as set forth by the primitive and parameter tables, there is an error in the implementation. The abbreviations used in the primitive and parameter sections are described below:

- blank – this means that the specific parameter will not be used in this service primitive
- C – this means that the parameter is conditional upon some state. A "C" differs from a "U" (User Option) due to the fact that if the stated condition exists, the parameter must be supplied, while a "U" means that the parameter's use is wholly up to the user. The exact conditions under which the parameter is used is explained in the text.
- C(=) – typically this is used when a request has an optional parameter. The C(=) will be in the "indication" position of the table. It means that if the user provides the parameter in the request, then it must also be present in the indication – what is more, it must have the same value as the parameter in the "request" column. The same applies to the "response" and "confirmation" columns.
- M – this means that the parameter must always be present, and no option not to use it exists.
- M(=) – this means that the parameter must always be present. If the parameter is in the "indication" column, then it must have a value equal to that of the "request" column. The same applies to the "response" and "confirmation" columns.
- U – this means that the use of the parameter is a user option. Therefore the presence of the parameter is completely optional, and will be used based upon a user's particular need.

4.2.3.3.5 The CM-logon and CM-forward services are presented with two parameter tables. The reason for this is that they may be operated in one of two ways, with each way using different primitives. The CM-logon service may be operated with a request, indication, response and confirmation, or may be operated with a request and confirmation only. The CM-forward service may be operated with a request, indication and confirmation, or may be operated with a request and confirmation only. Both cases are shown for these services.

4.2.3.4 Service Parameters

4.2.3.4.1 Throughout these service descriptions every parameter is described. In particular, there is a note that explains the purpose of the parameter, and a mandatory statement that states what values it may contain. In many cases, the mandatory statement states that it must conform to a named ASN.1 abstract syntax. These can be found in section 2.1.4.

4.2.3.4.2 Some primitive parameters have the same contents as APDU fields in the ASN.1 description. In most cases, the CM-ASE is tasked to copy the parameter value within the APDU field. In order to avoid confusion by defining identical data structures twice, the type of those primitive parameters is specified by simply referring to the corresponding ASN.1 type in the APDU. The ASN.1 is used in the service definition as a syntax notation only and does not implicitly imply any local encoding of these parameters. The implementation of these parameters remains a local implementation issue.

4.2.3.4.3 For the other parameters, the syntax is described by enumerating the authorised abstract values.

4.3 Chapter 2.1.4: FORMAL DEFINITION OF MESSAGES

4.3.1 Encoding/Decoding Rules

4.3.1.1 This section defines which PDUs the system must be able to encode and decode.

4.3.1.2 A system is not required to be able to encode or decode all messages specified in the ASN.1 description. An air system is not required to be able to encode a CMGroundMessage APDU nor to decode a CMAircraftMessage APDU. A ground system is not required to be able to encode a CMAircraftMessage APDU, but must be able to both encode and decode a CMGroundMessage APDU (due to the ground forwarding requirement).

4.3.1.3 This section defines the abstract syntax of the protocol. That is, it defines the structure of the PDUs that are to be sent between aircraft and the ground systems. It is written in a notation that is called ASN.1. It is strongly recommended that the reader is familiar with ASN.1 before attempting to understand the detail of section 2.1.4.2.

4.3.2 CM ASN.1 Abstract Syntax

4.3.2.1 Data types exchanged by CM ASEs are described in the CM SARPs by using a machine-independent and language-independent syntax. There is no constraint put on the implementers concerning the machine, nor the development language to be selected, for implementing the protocol.

4.3.2.2 The ASN.1 module *CMMessageSetVersion1* contains the data types of the protocol data units handled by the CM ASEs. Unlike common OSI ASEs (e.g. ACSE), no object identifier has been attached to the CM ASN.1 specification. Indeed, the ULA architecture releases the applications from negotiating during the dialogue establishment as the applicable abstract syntax. Object identifiers related to CM applications (application context name and version number) are defined in the ULA SARPs.

4.3.2.3 The CM ASN.1 is organized alphabetically.

4.3.2.4 ASN.1 Tags

4.3.2.4.1 Tags are used in ASN.1 to allow to distinguish data types when confusion is possible. For instance, when a data type contains two optional elements of the same type, if only one is encoded then there is no means for the decoder to know which element the decoded value is attached to.

4.3.2.4.2 Even if tag values are not used by the Packed Encoding Rules, the ASN.1 grammar mandates the use of tags in some cases. When specifying the CM data types, the following rules have been used:

- tags are always used within CHOICE data type, starting at 0 and then incremented by 1 for each entry.
- tags are not used at all in SEQUENCE data type when no confusion is possible. When an optional element is defined, all elements in the sequence are tagged.

4.3.2.5 Extensibility Markers

4.3.2.5.1 In order to allow the upgrade of the ASN.1 specification when new requirements are determined, the extensibility ASN.1 feature (ellipse, depicted by "...") has been used in applicable data types. This allows future modifications to data types as the applications evolve while retaining backwards compatibility.

4.3.2.5.2 The CMAircraftMessage, CMGroundMessage, CMAbortReason for the CM ASN.1 all have extensibility markers; the CMAircraftMessage and CMGroundMessage so that future messages can be added and the CMAbortReason so that future abort reasons may be added.

4.3.2.6 Entry Points

4.3.2.6.1 The top level (which will typically be used as an entry point in any ASN.1 compiler), is titled "Aircraft generated and Ground generated message choice". It consists of two structures: one is a choice of PDUs generated by the aircraft, and the other is a choice of PDUs generated by the ground system.

4.3.2.6.2 An index of the types defined in the ASN.1 is given in section 6.

4.3.2.7 Time Representation

4.3.2.7.1 Data types have been specified for containing time indication (Date, DateTime, Year, Month, Hours, Minutes, Seconds). This way of representing time is preferred over the pre-defined ASN.1 representations (GeneralizedTime and UTCTime) for optimization of the PER encoding.

4.4 Chapter 2.1.5: PROTOCOL DEFINITION

4.4.1 Message Sequence Diagrams

4.4.1.1 Time sequence diagrams or message sequence diagrams are used to denote the relationship between the primitives that form a CM service and the order in which they occur.

4.4.1.2 Implicitly, the concept of asynchrony is given through these message sequence diagrams, e.g. the indication/confirmation primitives occur some time after the request/response primitives.

4.4.1.3 Inherent to the service model is the notion of queuing. The CM-service indications and confirmations are delivered to the CM-users in the order that the corresponding CM-service requests and responses were issued.

4.4.1.4 Each message sequence diagram has the same structure. A generic message sequence diagram is illustrated in Figure 2.12.

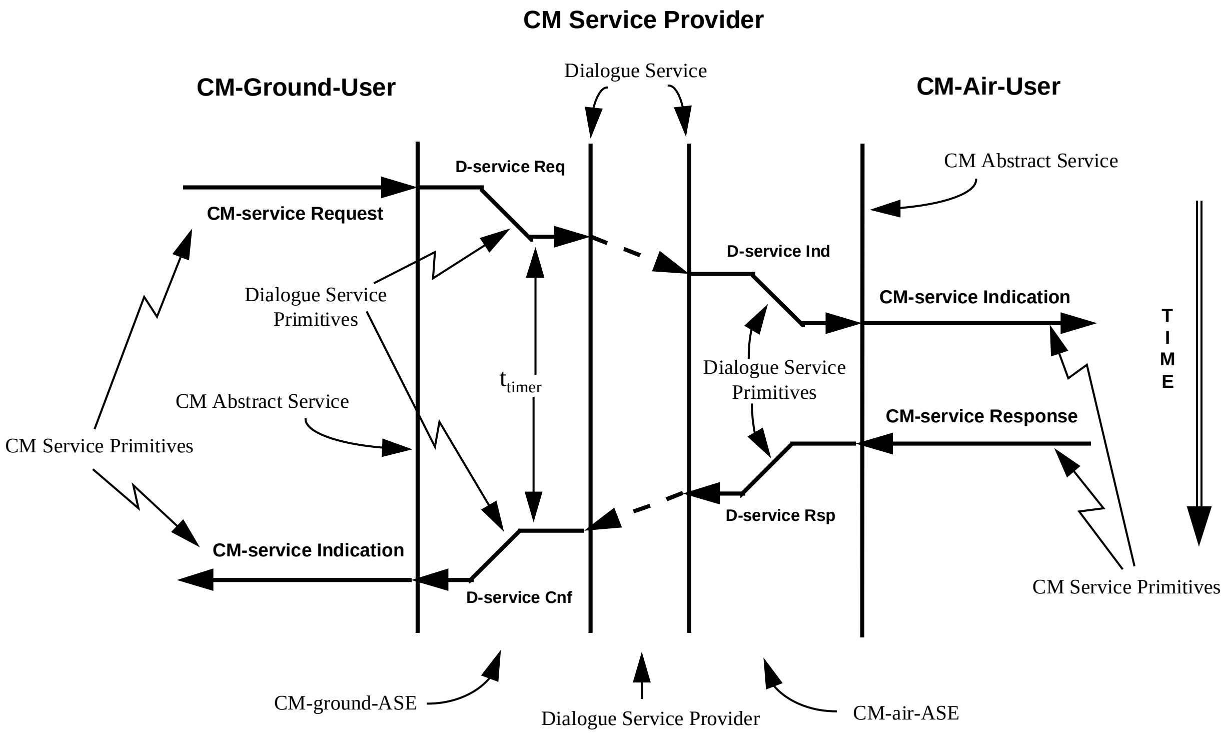

Figure 2.12. Generic Message Sequence Diagram

4.4.1.5 There are four vertical lines that separate the five major components in the CM system. From left to right they are:

- the CM ground user – that part of the ground system that uses the CM service to provide information to human or higher-level system users
- the CM ground ASE – that part of the ground system that implements the CM protocol
- the dialogue service provider – that part of the ground system, the air system and the networks that, together, provide the dialogue service, as defined in the upper layer architecture – on the figures this is the thin strip down the middle
- the CM air ASE – that part of the air system that implements the CM protocol
- the CM air user – that part of the avionics that uses the CM service to provide information to human or higher-level system users.

4.4.1.6 The middle three sections of the diagrams (CM-ground-ASE, dialogue service provider and CM-air-ASE) together form the CM service provider, and are labeled as such on the diagrams.

4.4.1.7 The outer two vertical lines represent the CM abstract service. Any lines crossing them represent the invocation of one of the CM service primitives. The CM service primitives are labeled in the CM-user part of the figure.

4.4.1.8 The inner two vertical lines represent the dialogue service. Any lines crossing them represent the invocation of one of the dialogue service primitives. The dialogue service primitives are labeled in the CM-ASE part of the figure.

4.4.1.9 The diagrams represent a sequence of events. Time is always considered to run down the figure from the top (representing the earliest time) to the bottom (representing the latest time).

4.4.1.10 If the ASEs set timers, these are marked on the figures by vertical lines with arrows at both ends. This is depicted in Figure 2.12 by the “t_{timer}” label.

4.4.1.11 It should be noted that the message sequence diagrams in the SARPs representing abort situations can be overlaid on top of any of the other figures to represent an abort in action.

4.4.2 CM Service Provider Timers

4.4.2.1 This section lists the technical timers that are defined in the protocol, and suggests values for them.

4.4.2.2 The purpose of the technical timers in the CM service provider is not operational. For operational reasons, there may be a requirement to have other timers that are shorter than the ones described here. The purpose of these technical timers is only to ensure that the ASE protects itself when communicating with a system that has failed, for some reason, to respond.

4.4.2.3 For example, suppose a ground system sends a contact request to an aircraft. Suppose further that the aircraft system has not implemented the CM protocol correctly, and it locks up without sending the contact response back to the ground system. The ground system will be in a state that is waiting for a result. The specification prevents the ground system from sending up another message until the first has been dealt with. This combination of events would make the ground system lock up unless a technical timer is used. When in this state, the $t_{contact}$ technical timer will eventually reach its maximum value. The ground system can then abort the connection. Thus the ground system protects itself from getting indefinitely locked up because of another system's failure.

4.4.2.4 It should be noted that the values set in these technical timers have been calculated thus: It has been assumed that the slowest possible network is in operation and that all systems in the path have their maximum delay. Should a reply to a request take longer than this, then it is assumed that something must be failing somewhere.

4.4.2.5 The assignment of values for timers must be optimized based on operational testing of the application. In such testing, incompatible timer values and optimum combinations can be identified. Implementations of CM protocol are required to support configurable values for all timers and protocol parameters, rather than having fixed values. This allows modification as operational experience is gained.

4.4.3 CM-ASE Protocol Description

4.4.3.1 The protocol description explains the rules by which the ASEs work. There is a detailed specification of actions taken by the ASEs when triggered by certain events:

- the arrival of a PDU through the dialogue service,
- the invocation of one of the service primitives by the user,
- the expiration of one of the internal timers, and
- an unrecoverable system error.

4.4.3.2 Both the air and ground ASEs have modules that mirror each other. The exception to this case is the when the ground is in a ground-ground initiator or responder mode. In this case, the two ground ASEs mirror each other.

4.4.3.3 There is a requirement that the ASE does not accept the invocation of primitives when no actions are described for that primitive in that state (2.1.5.3.1.1). Some explanation of this statement is needed:

4.4.3.4 The air and ground CM ASEs have several different states. When an ASE is in a particular state, only some primitives are sensible to invoke. For example, if a CM-logon service request is invoked, it is not sensible to invoke a second CM-logon service request until a reply has been received for the first one. There is no statement in the description of the protocol that explains what the air ASE should do if it receives a second CM-logon service request before the reply to the first one has been received. The SARPs therefore require (by statement 2.1.5.3.1.1), that the air user must not invoke a CM-logon service request during this period.

4.4.3.5 Thus only actions which are permitted are described. If an action is not described, then it is not permitted.

4.4.4 CM-ASE State Tables

4.4.4.1 State tables are provided. These should be an exact reflection of the CM protocol description, in a condensed form. The state tables are only presented for guidance, since the textual protocol description always takes precedence.

4.5 Chapter 2.1.6: COMMUNICATIONS REQUIREMENTS

4.5.1 Encoding Rules

4.5.1.1 This section states that PER (Packed encoding rules) must be used to encode the PDUs. PER is an ISO standard and is particularly efficient at encoding data. Implementers may use ASN.1 compilers to generate code that creates PER automatically.

4.5.2 Dialogue Service Requirements

4.5.2.1 The dialogue service requires a number of parameters to operate. This section defines those parameters that are not defined elsewhere, and states that the dialogue service must exhibit consistent behaviour with the Upper Layer Communications (Ref [2]).

4.6 Chapter 2.1.7: CM USER REQUIREMENTS

4.6.1 The requirements set out in this section are in line with those specified by the ADSP in the ICAO Manual of Data Link Applications (Ref [3]). These requirements are necessary for proper interoperability of ATN applications, and go into more technical detail than those specified by the ADSP Manual (Ref [3]).

4.7 Chapter 2.1.8: SUBSETTING RULES

4.7.1 There is some functionality within the CM SARPs that ground implementers may choose not to incorporate. For example, if a particular implementation of a ground system is designed never to use the contact function because it will always have ground-ground communications with other CM ground systems, then there is no requirement for it to have the code to support that function. An aircraft, on the other hand, must have the code for accepting and acting on a contact message, since it may fly into an area that does use the contact function.

4.7.2 There are some combinations of functionality that allow interoperability and some that do not. This section defines the combinations of functionality that are interoperable. It should be noted that aircraft must retain full technical functionality to be SARPs compliant.

4.7.3 The combinations or functionality are defined in a set of tables. The purposes of these tables are summarized below:

- Version number – only one version is defined. This is a placeholder for when future versions are defined.
- Protocol Options – this defines a number of options for parts of the protocol that may be implemented. The options may be implemented together. Each has a name associated with it – the predicate.
- CM-ground-ASE configurations – this defines seven combinations of protocol options, each of which yields a coherent protocol.
- CM-air-ASE configurations – this defines a single combination of protocol options, which is the only combination that yields a coherent protocol.
- Supported CM service primitives – this defines the conditions under which the service primitives are applicable.
- Supported CM APDUs – this defines the conditions under which the PDUs are applicable.

4.7.4 Only version 1 of the CM protocol is defined. Any CNS/ATM-1 compliant CM system must support this version.

4.7.5 Service Options

4.7.5.1 Some of the CM-services have options within itself. That is, there are more levels of functionality within a service besides full support or no support of that service. These services include the CM-logon, CM-update, CM-contact and CM-forward service.

4.7.5.2 The CM-logon, CM-update and CM-forward service can each have two possible implementation options for the CM-ground-ASE: one supports the maintaining of dialogue, the other does not. Since maintaining a dialogue is a CM-ground-ASE option, an implementation may not have a need to incorporate that function. Therefore, this allows an implementer to build a system that does not support maintaining a dialogue an is SARPs compliant.

4.7.5.3 The CM-forward service has three different options: forward initiator, forward responder and forward user. Forward initiator is used to denote a ground system that has built in the optional functionality to invoke the CM-forward service. Forward responder is used to denote the mandatory functionality that every CM ground system must employ, which is the ability to recognize a CM ground forwarded message and respond with (at minimum) "service not supported". The forward user is used to denote a ground system that has built in the optional functionality to make use of (i.e. distribute internally within the local receiving ground system) the information that is received from a CM ground forwarded message. A ground system may choose to implement a forward initiator but not a forward user, or vice versa.

4.7.6 The CM-ground-ASE must support the CM-logon, CM-forward (responder), CM-abort, and CM-provider-abort services. These represent the minimal functionality that a ground system can implement, and is called the core functionality. In total, there are 28 valid subsets, or implementations, that a CM-ground-ASE may have. A CM-air-ASE must be able to support all of the CM functionality. Therefore, there is only one valid configuration for an aircraft.

5. DIMENSIONS

5.1 PDU Size

5.1.1 Theoretical Limits

5.1.2 Error Handling

5.1.3

5.2 Number of Concurrent CM Connections

5.2.1 Based on peak traffic loads for data link equipped aircraft operating in the airspace in question along with whether or not there is a requirement for dialogue maintenance. The numbers specified by the ADSP can be found in Part II, Appendix A of Chapter 3, Table 3A-1 in Ref 1.8x, and are reproduced in Table 5-1.

Table 5-1. Exchange Rates Expected for DLIC Messages

Parameter	Oceanic-Continental enroute low density	Oceanic high-density	Continental high-density	Terminal area high-density	Aerodrome (Including: Approach/ Departure/ Taxi)
DLIC message exchange per aircraft	2 per ATSU	2 per ATSU	2 per ATSU	2 per ATSU	2 per flight
Instantaneous number of aircraft to be supported per ATSU	300	750	1250	450	250

5.2.2 There is nothing in the CM SARPs that restricts or dictates the number of connections that should be supported, either in the air or on the ground. This decision will be made based on many factors, including the airspace type, aircraft equipage expectations and operational procedures put into practice. Some concerns are given below.

5.2.3 Aircraft CM Connections

5.2.3.1 An aircraft will need to support at least one active CM connection. In reality, this implies needing to support at least two connections; one for active use, and one to recognize another CM request that cannot be complied with since the only connection is busy. If an aircraft is operating in an environment where a CM server exists or where a dialogue is maintained to a single centre, then one connection may be sufficient operationally. However, if the aircraft is operating in an environment where there are many different centres that need to perform CM services with the aircraft concurrently, then the aircraft may need to support multiple CM connections. The ground system topology will in part drive the aircraft equipage.

5.2.3.2 An aircraft CM implementation should take into account the possibility of receiving multiple connection requests at the same time.

5.2.4 Ground CM Connections

5.2.4.1 A ground system will probably need to support more than one CM connection. The numbers in Table 5-1 illustrate this probability. However, there may be implementation issues that will reduce the number of connections needed. For instance, having ground-ground forwarding capability would reduce the number of CM connections needed by ground systems. Since other interested ground systems would receive the forwarded aircraft's application information, there would be no need to make CM connections to the aircraft.

5.2.4.2 There are also considerations that can drive the number of CM connections higher. If there is an operational requirement in a specific area to maintain a CM dialogue with each aircraft, then that ground system will obviously need to allow for one CM connection per expected aircraft. If the expected number of aircraft per ATSU are as in Table 5-1, this can lead to significant cost and performance concerns.

5.2.4.3 An ground CM implementation should take into account the possibility of receiving multiple connection requests at the same time.

6. INDEXES / TABLES

6.1 Message Content Glossary Table

6.1.1 Table 6-1 contains the glossary for various message elements as used in the CM SARPs

Table 6-1. Message Content Glossary Table.

AEQualifier	An integer from 0-255 that identifies a particular application.
Air Applications	An indication of 1-256 airborne data link applications. Consists of *Application Name*, *Version Number*, and, when required for ground initiated applications, *Application Address.*
Aircraft Flight Identification	Field 7 of the ICAO flight plan.
Airframe Address	24 bit ICAO identification.
Airport	An IA5 string of 4 characters indicating an airport's identification.
APAddress	An application's unique technical communications address.
Application Name	An IA5 string of 3 characters indicating an application name (e.g., ADS, CPC, FIS, or CMA.)
Departure Airport	Name of departure airport.
Destination Airport	Name of destination airport.
CM Address	*Application Address* for CM.
ETD	Estimated time and date of departure.
Facility Designation	An IA5 string of 4 to 8 characters indicating the ICAO facility designation for a ground system.
Ground Applications	An indication of 1-256 ground data link applications. Consists of *Application Name*, *Version Number*, and *Application Address* data.
HHMMSS	Hour, minutes, seconds.
Response	An indication of whether the requested action was successful.
Result	An indication of whether or not the ground forwarding was successful.
Time	Time in hours and minutes.
Time Hours	Time as hours of day.
Time Minutes:	Time as minutes of an hour.
Version Number	Version number of the specified application.
YYMMDD	Date in year, month, day

6.2 ASN.1 Type Index

6.2.1 Table 6-2 lists each ASN.1 type defined in the CM SARPs in alphabetical order.

6.2.2 The second column lists those ASN.1 types that are used in the definition of the ASN.1 type in the first column, and the third column lists those ASN.1 types that use it. The second and third columns are therefore inverse references.

6.2.3 The range and resolution at the primitive level are given where applicable in the fourth column.

Table 6-2. ASN.1 Type Index

Type	Types Used by this Type	Types Using this Type	Range and Resolution
AircraftFlightIdentification		CMLogonRequest	

Airport		CMLogonRequest	
AEQualifier		AEQualifierVersion AEQualifierVersionAddres s	
AEQualifierVersion	AEQualifier VersionNumber	CMLogonRequest CMLogonResponse	
AEQualifierVersionAddress	AEQualifier VersionNumber APAddress	CMLogonRequest CMLogonResponse	
APAddress	LongTsap ShortTsap	AEQualifierVersionAddres s	
CMAbortReason		CMAircraftMessage CMGroundMessage	
CMAircraftMessage	CMLogonRequest CMContactResponse CMAbortReason		
CMContactRequest	FacilityDesignation LongTsap	CMGroundMessage	
CMContactResponse	Response	CMAircraftMessage	
CMForwardRequest	CMLogonRequest	CMGroundMessage	
CMForwardResponse		CMGroundMessage	
CMGroundMessage	CMLogonResponse CMContactRequest CMForwardRequest CMForwardResponse CMUpdate CMAbortReason		
CMLogonRequest	AircraftFlightIdentification LongTsap AEQualifierVersionAddress AEQualifierVersion FacilityDesignation Airport DateTime	CMAircraftMessage CMForwardRequest	
CMLogonResponse	AEQualifierVersionAddress AEQualifierVersion	CMGroundMessage CMForwardResponse CMUpdate	
CMUpdate	CMLogonResponse	CMGroundMessage	
Date	Year Month Day	DateTime	
DateTime	Date Time	CMLogonRequest	
Day		Date	Range: 1 - 31 Resolution: 1
FacilityDesignation		CMContactRequest CMLogonRequest	
LongTsap	ShortTsap	APAddress CMContactRequest	

		CMLogonRequest	
Month		Date	Range: 1 - 12 Resolution: 1
Response		CMContactResponse	
ShortTsap		APAddress LongTsap	
Time	Timehours Timeminutes	DateTime	
Timehours		Time	Range: 0 - 23 Resolution: 1
Timeminutes		Time	Range: 0 - 59 Resolution: 1
VersionNumber		AEQualifierVersion AEQualifierVersionAddres s	
Year		Date	Range: 1996 - 2095 Resolution: 1

7. EXAMPLE SCENARIOS

8. EXAMPLE ENCODING

TBD