EU/G7 Healthcards - WG7

Interoperability of Healthcard Systems Part 2

Achieving Interoperability

Prepared by

Dr David Markwell, MB BS.

(as coordinator of EU/G7 Healthcards WG7 Interoperability)

ADDRESS FOR COMMENTS
The Clinical Information Consultancy,
93 Wantage Road, Reading, Berkshire, RG30 2SN, UK
Tel +44-1734 584954 Fax +44-1734 504464
Email dmarkwell@cix.compulink.co.uk
Web site: http://www.compulink.co.uk/~cic/euhci.htm

Status	: Final version
Version	: 1.0d
Date	: 26 August 1996
Reference	: EUHCID-2.DOC

DEVELOPED by the Core Technical Group of the EU Healthcards Interoperability Feasibility Study

AGREED by the EU Healthcards Interoperability Feasibility Study Consensus Meeting on 3 July 1996

REVISED to take account of continuing discussions of Card Terminal API proposals from France and Germany.

STATUS

Final Version of this part of the report

Table of contents

Revisions since previous draft (1.0b)

1. Alternation of wording to align with Part 3.
2. Revision of interim notes on the MCT/GIP-CPS discussions on the CT-API.

Interoperability of Healthcard Systems

Final version - Part 2 - Achieving Interoperability

2 Preface

This document is the second of three parts of the Final Report of the Healthcard Interoperability Feasibility Study. It considers the application of the general concepts in Part 1 to the projects covered by the terms of reference for this work and to the further development of interoperable healthcard implementations. It provides a foundation for the specific interoperability proposals in Part 3.

This document is based on an earlier Interim Report revised, revised to take account of discussions in the EU healthcard Feasibility Study - Core Technical Group (CTG) during the first half of 1996. It formed the basis for agreement between EU funded healthcard projects at a consensus meeting on 3 July 1996.

This is the final version of this part of the report.

The parts of the document are available in Microsoft Word™ 6.0 files. These files are named in accordance with the following convention – – EUHCI*[a]-[n]*.DOC

[a] is a letter signifying the version of the document.

[n] is a digit signifying the part of the report.

Advice to readers

This document is a record of the issues and an outline of the rationale behind the specification.

If you wish to study the proposed interoperability specification you may wish to skip this part and turn directly to Part 3 of this report.

3 Introduction

3.1 Objectives

The objective of the Healthcard Interoperability Feasibility Study is to facilitate and encourage interoperability between healthcard systems by:

- Preparing a high level draft specification, or framework, for interoperable healthcard interfaces;
- Working with healthcard projects under the fourth framework to demonstrate the practical possibilities for interoperability.

This report contains the second draft outline of the proposed high level specification for interoperability. It also suggests the extent to which current projects may be able to implement this.

3.2 Associated projects

The EU Fourth Framework programme has funded three projects that involve the development and implementation of healthcards. These associated projects are DiabCard, CardLink and TrustHealth. The brief for the feasibility study is to work closely with the consortia involved in these projects to facilitate agreement and practical progress towards interoperability. In addition, we have established links with two other projects that are developing and piloting the use of cards in a manner that is directly relevant to the development of interoperability. The projects are Panacea (a healthcard project funded under the EU Eureka programme) and Quasi Niere (a large scale implementation of healthcard in renal replacement therapy recently launched in Germany).

3.3 Demonstrating interoperability

As a part of their negotiations for funding, the consortia involved in the CardLink and DiabCard projects have agreed to demonstrate interoperability. This demonstration of interoperability will initially apply only to the reading of administrative and emergency health data from cards issued by the other project. However, this limited demonstration should be a step towards wider interoperability rather than an end in itself. It should therefore be achieved in a manner that can be extended to other card systems rather as a one-off development.

3.4 G7 Healthcards

Interoperability is considered to be one of the most important prerequisites for widespread use of healthcards. Many of the potential clinical and administrative benefits depend on widespread use. Therefore, interoperability is a key to unlocking these benefits to the realisation of a global market for healthcards.

Harmonisation of Healthcards has also been identified as a sub-project in the G7 Global Healthcare Applications Project. Japan has indicated strong interest and has contributed a working document on the Content Access Manager, which is a key element of their approach to interoperability.

The results of the Healthcards Interoperability Feasibility Study are expected to form the central element in the EU input into this wider international initiative.

3.5 Strategy

The main theme of the strategy is to re-use rather than re-invent. The published Standards from ISO and CEN, existing work of EU funded projects, nationally endorsed initiatives and relevant industrial contributions will be adopted wherever possible. The task is to assemble this existing work, if necessary identifying pragmatic profiles and subsets, to deliver results that meet the perceived needs for functional interoperability.

3.6 Achieving interoperability

Part 1 of this interim report has considered the general requirements for interoperability. It classifies interoperability into three categories:

- Functional interoperability of Healthcard Systems:
  - Some or all of the functions of one Healthcard System make use of and/or to update computer readable data held on the Healthcards issued by another Healthcard System.
- Technical interoperability of Healthcard Systems:
  - Some or all of the computer readable data on cards issued by one Healthcard System can be read by one or more Card Access System (CAS) in another Healthcard System.
- Technical interoperability of Healthcard System components:
  - One or more of the components of one Healthcard System can be interchanged with the equivalent component of another Healthcard System without impeding or changing the overall operation of the Healthcard System.

The objective of this report is to facilitate functional interoperability between Healthcard Systems. This requires technical interoperability of Healthcard Systems and agreement on the extent of interoperability that is desirable and acceptable. The remainder of this report focuses on the technical aspects of interoperability.

Technical interoperability can be subdivided according to:

- Healthcard System components and interfaces between them;
- Healthcard System functions, data content and data structure.

4 User requirements for technical interoperability

4.1 Introduction

The best approach to any specification exercise is to start from a consideration of the requirements. The requirements for Healthcard System interoperability that are addressed by this report fall into two distinct categories:

- Immediate requirements for initial demonstration of interoperability within the timescales of the current project;
- Requirements for future implementation of interoperable Healthcard Systems on a wider European and/or Global scale.

The immediate requirements are clearly defined but are only a foundation for more extensive and useful future development. It is therefore important that the approach taken to meeting these requirements is a sound basis for future development rather than an end in itself. In particular, the techniques used to support limited interoperability between a small number of Healthcard Systems should be capable of stepwise evolution to support anticipated future requirements.

4.2 Immediate requirements

The immediate requirements are for users of one Healthcard System to read the administrative and emergency information from healthcards issued by another Healthcard System. The ability to read this information must also be reciprocated by the other Healthcard System.

The user should not need to consciously distinguish between the cards presented to them. They should be able to use the same reader, the same user interface and the same commands to view accessible card data irrespective of which Healthcard System issued the card. Furthermore, they should be prevented from viewing or updating information that they are not authorised to access.

4.3 Future requirements

Future requirements include:

- Extension of the ability to read administrative and emergency information from cards issued by a wider range of Healthcard Systems;
- Integration of access to interoperable healthcards into a wider range of applications;
- Extension of the information that is readable by other systems beyond the initially specified administrative and emergency data sets;
- Addition of security mechanisms that allow a healthcard held by a patient or healthcare professional to be unlocked or validated by the use of PIN or other agreed method;
- Extension of security mechanisms to restrict reading information from healthcards issued by a particular Healthcard System. A method of restriction that is acceptable to the issuer must be enforced by all interoperable systems. This mechanism is likely to involve the use of a healthcare professional card to enable an authorised person to read additional information;

- Addition of facilities to allow cards issued by one Healthcard System to be written to or updated by users of another Healthcard System. This will need to be controlled by an extension of the security mechanism to restrict the ability to write to or update cards. A security mechanism that is acceptable to the issuer (and to author(s) of any information that can be updated) must be enforced by all interoperable systems. This mechanism is likely to involve the use of a healthcare professional card to enable an authorised person to write or update information;
- Extension of security mechanisms and healthcard updating facilities to ensure that information written to or read from a healthcard is attributed to its author. A mechanism that is reliable, resistant to fraud and acceptable to the issuer of the card must be enforced by all interoperable systems. This mechanism is likely to involve the use of a healthcare professional card to apply a digital signature to information written to a card;
- Extension of all aspects of interoperability over time to include future generations of healthcare applications and cards. This should ensure that healthcards do not become unusable and that Healthcard Systems are able to implement new generations of healthcards while retaining the ability to access previously issued healthcards;
- Cost containment to ensure that, within the constraints of their capacities and functionality, low-cost Healthcard Systems are able to interoperate with more sophisticated Healthcard Systems.

4.4 Meeting requirements

The user requirements must be met by Healthcard Systems that comprise several components. Application software carries out the direct task of interacting with and providing services to the user. It must meet the user requirements but to do so it needs to be supported by other components such as a card reader and the healthcard itself.

5 A modular approach to Healthcard System interoperability

5.1 Introduction

In theory an application may interact directly with a reader or card but in practice a modular approach using intermediate interface software is more cost-effective. A modular approach reduces application development effort and reduces the risk of errors. It also facilitates interoperability by allowing changes in individual components to be accommodated by the interface software and thus transparent to the application. Furthermore, it allows specifications and components used in other card applications to be reused in Healthcard Systems.

There are several ways in which an Healthcard System can be described and subdivided. For the purposes of this study we consider four components and three distinct interfaces at which interoperability is required. These are illustrated in Figure 1 and described below.

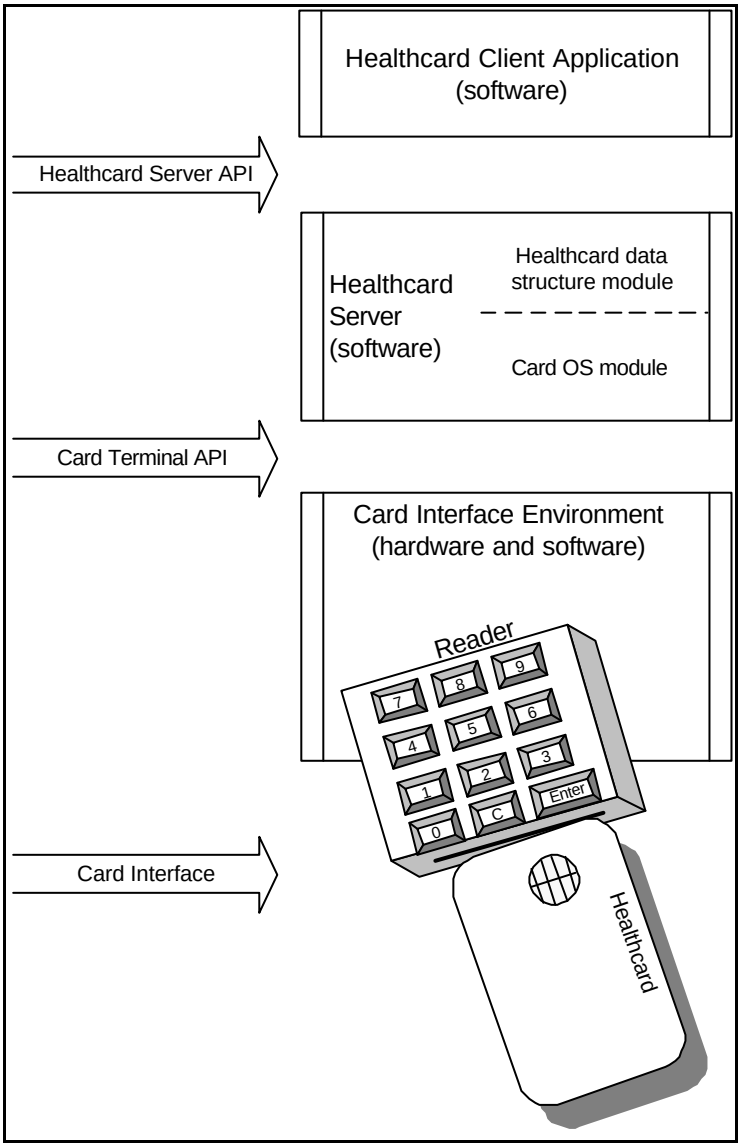

Figure 1. A modular approach to Healthcard System interoperability

5.2 Healthcard Client Application

5.2.1 Description

Definition

A Healthcard Client Application consists of one or more software modules or programs which, in order to fulfil user requirements, request services from a Healthcard Server.

Role

A Healthcard Client Application provides a user-interface that must:

- Allow users to access card related functions such as reading and/or updating a card;
- Display relevant information read from a healthcard;
- Enable entry of relevant information to be written to a healthcard.

The information to be accessed, displayed and entered through the Healthcard Client Application may include:

- Information required for the purposes of interoperability;
- Information particular to a specific Healthcard System.

The card related functions available to the user through the Healthcard Client Application may include:

- Functions required for the purpose of interoperability:
  - E.g. reading the interoperable information.
- Functions particular to a specific Healthcard System:
  - E.g. Issuing a new card.
- Functions required for the purpose of interoperability used in connection with information that is not required for the purposes of interoperability:
  - E.g. reading information that is specific to the Healthcard System.

The Healthcard Client Application may also interact with databases, networks and may meet other user requirements that do not use healthcards.

5.2.2 Developing interoperability

Points to be agreed

Agreement is required on the following points before an interoperable Healthcard Client Application can be developed:

- The information and/or functions that are subject to interoperability;
  - To facilitate support for “old” healthcards and Healthcard Client Applications a distinction must be made between data that is mandatory on the card, at the Healthcard Server API and in the Healthcard Client Application display.
- The minimum requirement for a user interface to enable the request and display of interoperable information;
- The specification of the Healthcard Server API with which Healthcard Client Applications must interact.

Development required

Once agreement has been reached on the above points Healthcard Client Application developers need to develop software that:

- Provides a user-interface that meets the agreed minimum requirements;
- Interacts with the agreed Healthcard Server API in the following ways:
  - Issuing Healthcard Server API commands and dealing appropriately with any status or error messages returned by the API;
  - Receiving and appropriately processing information received from the Healthcard Server API so that it can be displayed or otherwise used within the Healthcard Client Application;
  - Presenting relevant information from the Healthcard Client Application to the Healthcard Server API in the structures required to support API commands.

5.2.3 Specification of the interoperability demonstration

Information and functions subject to interoperability

The interoperability demonstration between CardLink and DiabCard will support read-only access to administrative information and emergency clinical information. Part 3 of this Report specifies the data content and structure. The data content to be conveyed is based closely on the CardLink emergency data set. A distinction is made between information that can cross the Healthcard Server API and information that it is mandatory for the Healthcard Client Application to display or process.

Minimum requirements for a user interface

The agreed minimum specification for an interoperable Healthcard Client Application describes what the interface must do not how it must be done.

The Healthcard Client Application must:

- Allow the user to read an interoperable healthcard without manually specifying the Healthcard System by which that card was issued;
- Display the interoperable healthcard information read from the card:
  - The display should allow the user to distinguish between information that is not present on an individual healthcard and information that is not available from healthcards of a particular type. Information may be unavailable as a result of:
    - The capabilities or storage capacity of the healthcard;
    - Limited interoperability between the user's system and the healthcard;
    - Security restrictions and interoperability agreements which prevent access to some information.

    For example, the interface may indicate that card data:
    - States that the patient is allergic to drug X;
    - States that the patient is not allergic to drug X;
    - States that the patient is not known to be allergic to drug X;
    - Contains no statement about allergy to drug X because the clinician has not entered any information on this point;
    - Contains no statement about allergy to drug X because the healthcards of this type has no facility to store information on this point.

Specification of the Healthcard Server API

See section 5.3.

5.2.4 Existing work and required development

The associated healthcard projects have developed user interfaces for their own card systems. These will form the basis for development of the interoperability demonstration between CardLink and DiabCard.

The CardLink and DiabCard Healthcard Client Applications will need extensions or modifications to:

- Allow the display of the interoperable information;
- Provide a user option that initiates reading of interoperable data from any interoperable healthcard;
- Issue commands to and receive information from the agreed interoperable Healthcard System API;
- Disable Healthcard System specific options (such as updating the card) when processing a healthcard from another Healthcard System.

5.3 Healthcard Server API

5.3.1 Description

Definition

The Healthcard Server API consists of the specification of an Application Programming Interface that is to be provided by Healthcard Servers and used by Healthcard Client Applications to enable communication with healthcards.

Role

A common Healthcard Server API must be supported by Healthcard Servers that provide access to one or more types of healthcard. This allows Healthcard Client Applications to use the same interface commands to access to different Healthcard Servers. Therefore, if a Healthcard Server exists to support a particular healthcard, that card can be accessed by any Healthcard Client Application that supports the common Healthcard Server API.

The Healthcard Server API must specify:

- The commands supported by the Healthcard Server;
- The data structures used in association with the commands to pass information to and from the Healthcard Server.

The full Healthcard Server API provided by an Healthcard Server may include:

- Commands required for the purpose of interoperability:
  - E.g. A command that the Healthcard Client Application can issue to read the interoperable information from a healthcard.
- Data structures associated with the interoperable commands:
  - E.g. The structure in which the Healthcard Server presents the interoperable information to the Healthcard Client Application.
- Commands specific to the particular Healthcard System:
  - E.g. Commands to read specialised information and commands to add information to the card.
- Data structures specific to the particular Healthcard System:
  - E.g. The structure in which Healthcard System specific information is presented to the Healthcard Client Application.

5.3.2 Developing interoperability

Points to be agreed

The following points must be agreed to specify an interoperable Healthcard Server API:

- The information and/or functions that are subject to interoperability;
- Whether the Healthcard Server API should support asynchronous communication:
  - If asynchronous communication is required, an additional command is needed to retrieve information obtained by the previous command:
    - E.g. An asynchronous command to read information through the Healthcard Server returns immediately without any data. This allows the Healthcard Client Application to continue other processes while reading continues. The Healthcard Client Application can intermittently check the progress of reading sending another asynchronous command to the Healthcard Server. When reading is complete, the Healthcard Server responds to this command by returning the data to the Healthcard Client Application;
    - Asynchronous commands are essential for processes that take a perceptible time to complete, unless the operating system allows asynchronous processing by multi-threading or pre-emptive multi-tasking.
- Healthcard Server API commands:
  - Required to:
    - Initialise the interface;
    - Undertake general functions such as checks for the presence of a card or checks of PIN validation;
    - Read specified sets of information from the healthcard;
    - Write or update specified sets of data on the healthcard.
  - Responses must be defined for each of these commands to report success of the command or the cause of failure.
- Structures in which interoperable information is to be presented to, and/or returned from, the Healthcard Server API:
  - The data structure should allow the Healthcard Client Application to distinguish between information that is not present on a healthcard and information that is not available from healthcards of a particular type. Information may be unavailable as a result of:
    - The capabilities of the healthcard;
    - Security restrictions and interoperability agreements;
    - Limited interoperability between the user's system and the healthcard.
- An approach for Healthcard Server API commands that are specific to particular Healthcard Systems:
  - The non-availability of these commands in respect of other healthcards or other Healthcard Systems should be handled in a safe and user-friendly manner;
  - The approach should take account of possible future extension of the interoperable Healthcard Server API to provide similar interoperable functions.

Development required

The development required to support an agreed Healthcard Server API is described in sections 5.2.2 and 5.4.2.

5.3.3 Specification of the interoperability demonstration

The information and/or functions that are subject to interoperability

The interoperability demonstration between CardLink and DiabCard will support read-only access to administrative information and emergency clinical information. Part 3 of this report specifies the data content and structure. The data content to be conveyed is based closely on the CardLink emergency data set. A distinctions is made between the information that can cross the Healthcard Server API and information that it is mandatory for the Healthcard Client Application to display or process.

Support for asynchronous commands

Support is required for asynchronous Healthcard Server API commands because it takes several seconds to read the information required from a healthcard. Furthermore, many interoperable systems will inevitably run in environments that do not support pre-emptive multitasking or multi-threading.

Healthcard Server API commands

The interoperability demonstration needs to support the following functions:

- Healthcard communication:
  - A command to initiate reading of interoperable administrative and emergency data from the card;
  - A command to check the current status of a previous read command and if reading is complete to return the data read.
- Card control:
  - Commands to enable a card to be inserted, detected, initialised, powered off and removed. Alternatively the Healthcard Client Application may have direct access to the Card Terminal API. When a card is initialised the Healthcard Server must be automatically configured to process subsequent commands in a manner that is appropriate to that type of card.
- Interface control:
  - Commands to open and close the interface may be required. These commands may be passed through the Healthcard Server to the Card Terminal API. Alternatively the Healthcard Client Application may have direct access to the Card Terminal API.

Part 3 of this report specifies the Healthcard Server API commands to be supported in the interoperability demonstration.

Structures in which interoperable information is to be presented

Structures need to be agreed for the return of interoperable administrative and emergency clinical data.

There is a European Prestandard on “Identification, administrative, and common clinical data structure for Intermittently connected Devices used in healthcare (including computer readable

cards)" ENV 12018:1995 (HCI-CEN-01). This proposes a structure for healthcard information beyond the scope of the current study. Nevertheless, it provides a good basis for further development from the foundation laid by CardLink. It adopts a similar approach using ASN.1 but applies it more formally and completely. Furthermore, the normative status of this document requires us to consider its provisions carefully.

Part 3 of this report specifies the structure of the administrative and emergency medical information to be passed across the Healthcard Server API in interoperability demonstrations.

The proposed structure is more flexible than the one adopted by CardLink 1 and follows the spirit of the ENV12018. However, care has been taken to ensure that:

- The data content required by CardLink users is fully supported;
- Cards already issued by CardLink can be mapped through a suitable Healthcard Server to the meet the mandatory requirements of the interoperable Healthcard Server API.
- Existing CardLink applications can display the mandatory minimum interoperable data set without significant changes to their user interfaces.

The proposed structure does not fully comply with ENV12018. However, the provisions of ENV12018 have been adopted wherever they are fully consistent with the objectives of the interoperability feasibility study. Reasons for deviation from this European Prestandard include the following:

- Data items required by ENV12018 that are:
  - Not supported by existing CardLink or DiabCard healthcards;
  - Excluded from CardLink or DiabCard data sets on the explicit advice of users;
  - Believed to conflict with data protection legislation in some Member States.
- Data items that are optional in ENV12018 but are:
  - Unlikely to be relevant for the intended administrative and emergency purposes of the proposed interoperable cards;
  - Add to the potential for variety, complexity and inconsistent interpretation and may therefore limit interoperability.
- Lack of clarity in ENV12018 about the implication of the mandatory or optional status of different data items. It may be mandatory to:
  - Record a data item;
  - To record and data items if it is applicable to the card holder;
  - To be able to store the information in a card if the issuer, user and/or patient wishes to do so;
  - To store the information in a card if it has been recorded;
  - To display (or process) the information if it is stored in the card;
  - To display (or process) the information if it is stored in the card and is relevant to the Healthcard Client Application being used;
  - To display (or process) the information if it is stored in the card and is supported by the Healthcard Client Application.

Healthcard Server API commands that are specific to particular Healthcard Systems

The Healthcard Server API may be extended to support specific commands and data structures that are only supported by cards issued by the same Healthcard System.

Healthcard System specific commands should be:

- Prefixed with a widely-recognised Healthcard System identifier;
- Disabled on Healthcard Server modules supplied to users of other Healthcard Systems.

When a Healthcard Servers receives an unrecognised commands it must:

- Return an unrecognised command error code;
- Ignore any values or buffer pointers passed with the command;
- Return to the same state that it was in prior to the unrecognised command so that it can process the next valid command in the usual way.

5.3.4 Existing work and required development

The CardLink project already has an API that meets many of the requirements stated here. This has been used as the basis for development of the proposed Healthcard Server API for the interoperability demonstration.

The DiabCard interface is more specialised and requires more changes to meet the requirements of interoperability.

The Panacea project plans to fully implement the European Prestandard on "Identification, administrative, and common clinical data structure for Intermittently connected Devices used in healthcare (including computer readable cards)" ENV 12018:1995 (HCI-CEN-01).

See also 5.2.4 and 5.4.4.

5.4 Healthcard Server

5.4.1 Description

Definition

A Healthcard Server consists of one or more software modules that provide a common Healthcard Server API to allow access to different healthcards issued by different Healthcard Systems. The Healthcard Server uses lower level Card Terminal API command to interact with the Card Terminal and to read, write or update healthcard data.

Role

The main roles of a Healthcard Server are:

- To map data between the form of presentation used at the Healthcard Server API and the files and formats appropriate to a healthcard issued by a particular Healthcard System;
- To take account of the card operating system and issue the correct commands to locate, read and write data.

Subdivision of the Healthcard Server

The Healthcard Server can be subdivided into two modules:

- The *healthcard data structure module (HS-HDSM):*
  - This is concerned with mapping between the Healthcard Server API and the logical layout of the healthcard data.
- The *card operating system module (HS-COSM):*
  - This issues commands appropriate to the card operating system in use.

Initialisation

When a card is inserted, the Card Terminal API returns to the Healthcard Server information that allows it to identify the card operating system. The HS-COSM is then configured for that card operating system. This may be accomplished by reading a configuration file or by loading a Healthcard Server module that contains a HS-COSM appropriate for that card operating system.

The HS-COSM can then be instructed to read configuration information from the card to identify the card data structure. At this point the HS-HDSM is configured to support the appropriate card data structure. This may be accomplished by reading a configuration file or by loading a Healthcard Server module that contains a HS-HDSM appropriate for that healthcard data structure.

Access to card data

When a request to read data from a card is received from the Healthcard Server API, this is interpreted by the Healthcard Server, which issues the necessary commands to the Card Terminal API. These commands are passed through the Card Terminal to the card. They select appropriate files and read the required data through the Card Terminal. The data returned by the Card Terminal API is then processed by the Healthcard Server and presented to the Healthcard Server API in the specified structure.

When a request to write to or update the card is received from the Healthcard Server API, it must be accompanied by the relevant information in the specified structure. The Healthcard Server processes this into the form required to allow it to be written to the inserted healthcard. This includes reformatting the data to meet the requirements for healthcards issued by a particular system. It may

also need to take account of the information currently on the card (e.g. extending or rewriting a file, flagging previous information as obsolete, updating reference tables, adding check sums and digital signatures). The appropriate commands are then sent to the card through the Card Terminal.

Options for Healthcard Server design

There are two approaches that could be taken to the overall design of the Healthcard Server:

- Separate modules could be provided and could communicate through an common interface (see Figure 2);
- The two components of the Healthcard Server (HS-COSM and HS-HDSM) could be combined in a single software module (see Figure 3).

Although there will inevitably be a degree of separation between the components, a common interface at this level is likely to be difficult to define and implement. Furthermore, while such an interface would add flexibility, it is not essential to the objective of providing healthcard interoperability. The interface needed depends on the requirements for a particular healthcard and how those requirements can best be met by the card operating system in use. Defining a common interface between the HS-COSM and HS-HDSM component would be equivalent to defining a common card operating system. It would then require the HS-COSM to provide a map from that common operating system to the operating system of the actual card. A simple common interface would inhibit the use of more powerful operating systems while a more comprehensive interface would place an unreasonable burden on the HS-COSM layer for cards with more primitive operating systems.

Initially, the integration between different HS-COSM and HS-HDSM modules will be an internal matter for those supporting a particular healthcard design with a particular operating system. This may lead to some duplication of software libraries where the same card operating system is used for several differently structured healthcards.

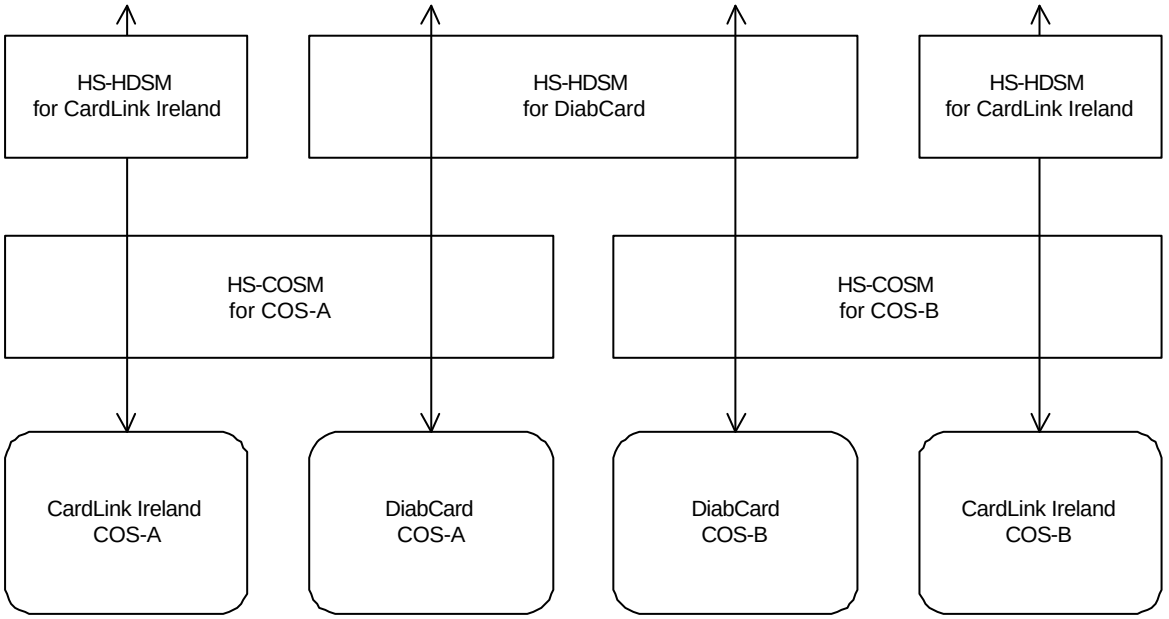

Figure 2. Healthcard Servers with separate HS-HDSM and HS-COSM modules

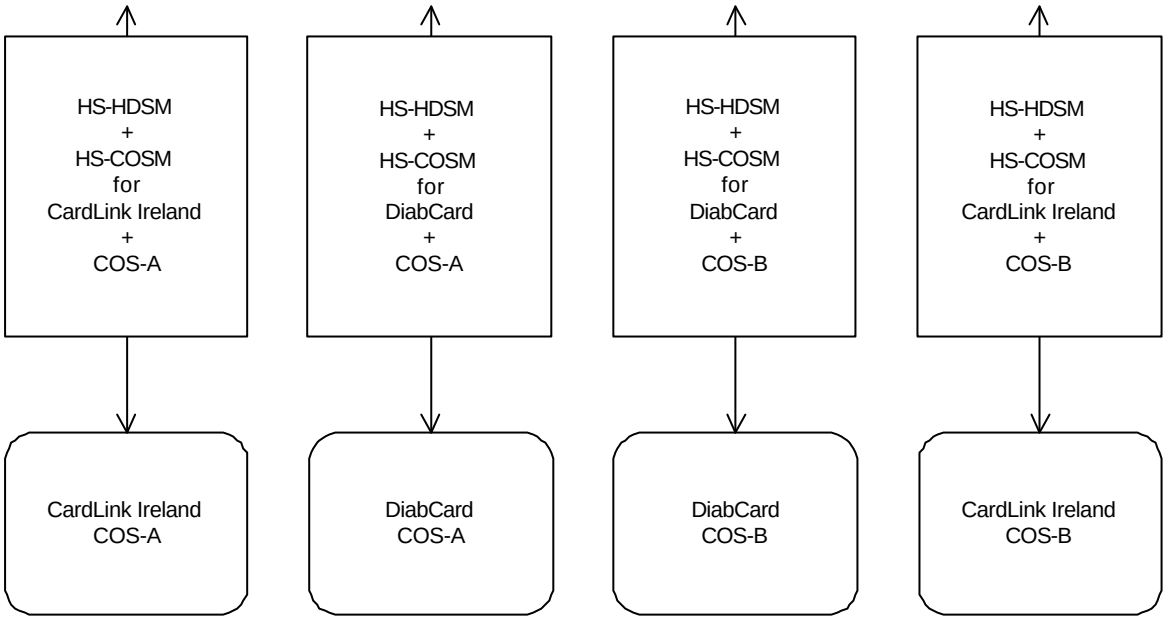

Figure 3. Healthcard Servers specific to combinations of card OS and data structure

Options for Healthcard Server module design

There are two distinct approaches that can be taken to the development of the Healthcard Server modules (HS-COSM and HS-HDSM)

Library option

- A separate library of routines could be provided by those responsible for the design of each healthcard data structure. A core routine of the Healthcard Server software would identify the data structure of the healthcard being accessed. It would then load and call the relevant library routines:
  - The main advantage of this approach is that it provides unlimited flexibility for the programmer to parse and restructure data in a manner that is optimised for a particular healthcard;
  - The disadvantage of this approach is that each of these libraries needs to be compiled and tested in any system operating environment used by a Healthcard Client Application.

Configuration file option

- A Healthcard Server Configuration Table (HSCT) could be defined to specify the structure of the interoperable data on a healthcard. These tables could be distributed to interoperable systems. Larger capacity healthcards could carry their own HSCT in a standardised location on card. A general purpose Healthcard Server would identify the structure of the card being read, read the relevant HSCT from the card or from the local or network system and then use this to control the relevant mapping:
  - The main disadvantages of this approach are that:
    - The use of an HSCT may limit the flexibility of healthcard data structure;
    - General purpose Healthcard Servers are unlikely to be optimised for mapping all supported data structures.
  - The advantage of this approach is that only one general purpose Healthcard Server needs to be supported for each operating environment used by a Healthcard Client Application. HSCT maintenance and distribution should be simpler than maintenance, distribution and installation of program libraries.

5.4.2 Developing interoperability

Points to be agreed

To enable the development of interoperable Healthcard Servers agreement is required on the following points:

- The specification of the Healthcard Server API (see 5.3.2);
- The specification of the Card Terminal API (see 5.5.2);
- The approach to be taken to the recognition of different card operating systems;
- The approach to be taken to the recognition of different healthcard data structures:
  - This includes consideration of the need to translate coded or abbreviated data.
- The approach to be taken to the support and integration of HS-COSM and HS-HDSM functionality.

Development required

Healthcard Server software must be developed to allow interoperable data on cards issued by each Healthcard System to be mapped to the Healthcard Server API.

5.4.3 Specification of the interoperability demonstration

Specification of the Healthcard Server API

See 5.4.3.

Specification of the Card Terminal API

See 5.5.3.

Approach to be taken to the recognition of different card operating systems

The card operating system should be recognised by reading the Answer to Reset (ATR) when the card is initialised. A Card Terminal API function is required that enables this information to be obtained from any type of card supported by the Card Terminal.

If the card operating system detected cannot be supported by the currently loaded Healthcard Server then it must load an alternative Healthcard Server that does support it. At this stage the healthcard data structure may be unknown (i.e. if two Healthcard Systems issue cards with the same operating system). The Healthcard Server loaded may still be inappropriate for the card but it will at least allow the Healthcard Server to read the Application Identity File from the card (see below).

The approach to be taken to the recognition of different healthcard data structures

Once a Healthcard Server that supports the card operating system is in place the Healthcard Server should immediately read the Application Identity File from the card. This allows the Healthcard Server to check the Card Issuer Identifier.

If the Card Issuer Identifier indicates that the card is not one of those supported by the currently loaded Healthcard Server then it must load an alternative Healthcard Server that supports both the card operating system and the data structure used by that Card Issuer.

Support for and integration of HS-COSM and HS-HDSM functionality.

The options discussed in CardLink Proposition for Interoperable solutions (HCI-CL-01) include "dynamic selection of card module DLLs" (solution 1) and "parametrable and dynamic card module" (solution 2). These are broadly equivalent to the two options discussed above.

A similar proposal is made in the DiabCard paper on DiabCard System Architecture (HCI-DC-01). In this case the Healthcard Server is represented by the "Card Access Service" and the configuration for different healthcard data formats is undertaken by a "Card OS Specific Agent". This is broadly equivalent to the configuration file option described above.

The use of dynamically loaded Healthcard Server library modules appears to be the most practical option for the interoperability demonstration. These modules need to support any combinations of card operating system and healthcard data structure that are in use. The approaches taken at this may differ from the eventual approach. Use of separate function libraries may be practical for demonstration purposes even if the longer term approach agreed uses a configuration table.

5.4.4 Existing work and required development

The *Card Modules* of CardLink provide a good starting point for development of Healthcard Servers.

The existing CardLink Card modules and the equivalent elements of the DiabCard software will need to be updated to support:

- The Healthcard Server API;
- The card operating systems used by participating Healthcard Systems;
- Any revisions to the healthcard data structure required to support the interoperable data.

Modification or development of Healthcard Server software will also need to take account of and make appropriate use of:

- Healthcard Server software provided by the other associated project to enable access to interoperable data on other Healthcard Systems;
- The Card Terminal API.

5.5 Card Terminal API

5.5.1 Description

Definition

The Card Terminal API consists of the specification of an Application Programming Interface that is to be provided by Card Interface Environments and used by a Healthcard Servers to control Card Terminals and to communicate with cards. Its commands are independent of the type of card, card reader and of the application(s) for which the card is used.

Role

A common Card Terminal API must be provided by the Card Interface Environment to support the use of one or more types of card reader. This allows Healthcard Servers to use the same interface commands to access different card readers and cards. with a common interface to the card and card reader irrespective of the type of card or reader used. Therefore, if Card Terminal software exists to support a particular card reader, that card can be used with any Healthcard Server that supports the common Card Terminal API.

The Card Terminal API must specify:

- The commands to be supported by the Card Interface Environment;
- The data structures used in association with the commands to pass information to and from the Card Interface Environment.

The full Card Terminal API provided by a Card Interface Environment may include:

- Commands required for the purpose of interoperability:
  - E.g. A command to check and initialise the system for an inserted healthcard.
- Data structures associated with the interoperable commands:
  - E.g. The structure in which the Card Interface Environment presents information about an inserted card to the Healthcard Server.
- Commands specific to a particular Card Terminal:
  - E.g. Commands that utilise a built in screen or keypad.
- Data structures specific to a particular Card Terminal:
  - E.g. The form in which information typed on a built-in keyboard is presented.

5.5.2 Developing interoperability

Points to be agreed

The following points must be agreed to specify an interoperable Card Terminal API:

- The extent of command specialisation and decomposition:
  - Several related functions of a Card Terminal can be carried out either with a single command with different parameters or with separate commands. Different approaches result in variations between card interface commands that have the same effect.
- Card communication transparency:
  - The card operating system mapping is undertaken by the HS-COSM component of the Healthcard Server (see 5.4.1). Therefore, the assumption is that commands can be

passed direct to the card with the Card Interface Environment simply providing a channel through which the command is sent.

- Whether the Card Terminal API should support asynchronous communication:
  - If asynchronous communication is required a command to retrieve information obtained by the previous command is required:
    - An asynchronous command to read information from the card returns immediately without any data. This allows the Healthcard Server and the Healthcard Client Application to continue other processes while reading continues. The progress of reading is checked using another asynchronous command. When reading is complete, the Card Terminal responds to this command by returning the data that has been read to the Healthcard Server;
    - Asynchronous Card Terminal API commands are essential for processes that take a perceptible time to complete, unless the operating system allows asynchronous processing by multi-threading or pre-emptive multi-tasking. However, asynchronous operation is not needed if each individual command can be completed in a fraction of a second.
- Card Terminal API commands to be supported:
  - Commands are required to:
    - Initialise the interface;
    - Undertake general functions such as checks for the presence of a card;
    - Communicate with the healthcard.
  - Responses must be defined for each of these commands to report success of the command or the cause of failure.
- Structures in which data is passed to and from the Card Terminal API:
- An approach for Card Terminal API commands that are specific to particular card readers or card reader features.

Development required

The development required to support an agreed Healthcard Server API is described in sections 5.4.2 and 5.6.2.

5.5.3 Specification of the interoperability demonstration

General approach

The interoperability demonstration will use a subset of the Card Terminal API defined in the Multifunction Card Terminal (MCT) produced by the Joint Panel in Healthcare of the German National Research Centre for Information Technology (HCI-MCT-01). However, the specification highlights the services required rather than a specific implementation. The objective is to enable future migration to support different CT-API specifications with a minimum of disruption to other interoperable components.

The need for a flexible approach arises from an awareness of other work in this field. This includes:

- The French GIP CPS "Gestionnaire" and " Commandes Lecteur" specifications[1];
- Current discussions within the TrustHealth project between French and German experts responsible for the MCT and GIP-CPS proposals;
- The Japanese CAM Card Reader Device Driver specification.

[1] “Gestionnaire” and “Commandes Lecteur Mode Transparent” (GIP-GEST and GIP-LECT)

A limited implementation that supports only the core services defined in the Part 3 of this report will be sufficient for the initial demonstration of interoperability.

The extent of command specialisation and decomposition

- The specialisation and decomposition of commands will be as specified for the MCT.

Card communication transparency

The Card Terminal API will provide a transparent channel through which commands can be passed direct the card. The MCT command for this process is:

- CT_data(CtNumber, **destination**, source, lenc, **command**, lenr, **response**):

  destination = 0 (first card in the specified Card Terminal);
  destination = 2 (second card in the same Card Terminal);
  command = The command (including data) submitted to the card;
  response = The response returned by the card.

The MCT specification makes one exception to this rule. It allows a limited range of industry standard card access commands to be mapped to a memory card without a processor. This exception is not essential for the interoperability demonstration but will be useful if the same readers are used for other cards (e.g. the German Health Insurance card).

Whether the Card Terminal API should support asynchronous communication

The current MCT specification does not support asynchronous communication at the Card Terminal API.

It is probably unnecessary to support asynchronous communication at this interface in the interoperability demonstration. Card read commands issued by Healthcard Server can if necessary be adjusted to avoid prolonged periods of waiting at this interface. However, in future, further consideration should be given to this point.

Card Terminal API commands to be supported

The three CT-API commands defined in the MCT specification should be supported in the interoperability demonstration. These are:

- ct_open;
- ct_data;
- ct_close.

The CT_data command is used to convey commands to the card or to the Card Terminal (depending on the value of destination parameter - dad).

Communication with the card is transparent so no definition of card commands is required at the Card Terminal API.

In the case of communication with the Card Terminal, the MCT specification identifies a set of “General CT commands” and a further set of commands for Card Terminals with built-in keypads and displays.

For the purposes of the interoperability demonstration only three General CT commands need to be supported. These are:

- REQUEST ICC;
- GET STATUS;

- EJECT ICC.

The current CardLink Read Functions appear to map to these MCT functions as follows:

Table 1. Comparison of CardLink and MCT functions at the Card Terminal API

CardLink	MCT
Open_Reader	ct_init ct_data(dad=01, command=RESET_CT, unit=CT … ct_data(dad=01, command=GET_STATUS, unit=CT,… The last of these commands returns most of the information about the card reader that the Open_Reader returns.
Close_Reader	ct_close
Reader_Status	ct_data(dad=01, command=GET_STATUS, unit=CT,tag=ICC Returns status bytes indicating insertion status of all ICC connections.
New_Card	ct_data(dad=01, command=REQUEST_ICC, unit=ICCnumber, qualifier=NoMessage+RespondATRHistorical, …
Input_Card	ct_data(dad=00, command=CommandToICC, …
Output_Card	ct_data(dad=00, command=CommandToICC, …,response= ResponseFromICC
Power_Off	ct_data(dad=01, command=EJECT_ICC, unit=ICCnumber, qualifier=NoMessage+NoAction, …
Remove_Card	ct_data(dad=01, command=EJECT_ICC, unit=ICCnumber, qualifier=StdMessage+Throwout*, … *If the reader can eject the card.
Get_Output	Not supported directly but the last data retrieved should be available from internal buffers.

Structures in which data is passed to and from the Card Terminal API

These are as defined for the relevant MCT commands.

Card Terminal API commands that are specific to particular card readers or card reader features

If a Card Terminal without a built in display or keypad receives a command that requires one of these devices it should report an error to the Healthcard Server. However, if a Card Terminal without a built-in display receives a General CT command that includes information to be displayed, it should carry out the command without reporting an error. For example, if the command specifies that the Card Terminal should display the message “Please insert card” while waiting for a card to be inserted, the Card Interface Environment should still await card insertion even though the message cannot be displayed.

5.5.4 Existing work and required development

The MCT appears to meet all the immediate needs for the healthcard interoperability demonstration. It has already been implemented for demonstration purposes in Germany. Further implementations are required to support interoperable sites

CardLink has already developed and used a Reader Module Driver with a Reader Interface that provides similar functionality (see HCI-CL-01). DiabCard has proposed a high level Reader Interface with similar functions (see HCI-DC-01). Software developed for these interfaces will need some revision and development to support and make use of the MCT. These development requirements are notes in 5.4.4 and 5.6.3.

5.6 Card Interface Environment

5.6.1 Description

Definition

Hardware and software that together provides the Card Terminal API and communicates with one or more cards.

Role

The Card Interface Environment is concerned with enabling cards with different communications protocols to be accessed through a common Card Terminal API. It must be able to:

- Process and respond to commands from the Card Terminal API that require the Card Terminal to act in a particular way (e.g. check for the presence of card, input and verify a PIN);
- Detect the protocol used by an inserted card;
- Undertake the necessary communication with the card to carry out the commands presented at the Card Terminal API in a manner that is appropriate to the type of card in use.

Components of the Card Terminal

- Hardware - A Card Terminal (or reader):
  - A Card Terminal includes:
    - Contacts for one ID-1 IC card;
    - Optionally contacts for one or more additional ID-1 IC-cards;
    - Optionally contacts for one or more ID-000 plug in IC cards.
  - This may be:
    - Integrated in a computer system;
    - Connected to a computer system as an external device.
  - An external Card Terminal may have:
    - A keypad and display.
- Software:
  - Card Terminal control program:
    - This may be located within the Card Terminal.
  - The Host Transport Service Interface:
    - Manages communication with the Card Terminal;
    - Provides the Card Terminal API;
    - Located in the host environment.

5.6.2 Developing interoperability

Points to be agreed

- Agreement on the Card Terminal API;
- Agreement on the card types and communication protocols to be supported at the Card Interface;
- Agreement on which card readers will be supported in which computer operating systems:
  - The computer operating system is involved because the Card Interface Environment includes:
    - Driver software that runs on the host system;
    - A communications layer linking the host system to the card reader.
  - Once the Card Terminal API is agreed, it only requires the relevant Card Interface Environment software to be written to enable other card readers or operating systems to be supported.

Development required

Widespread healthcard interoperability requires development of software required to provide the agreed Card Terminal API for a wide range of card readers.

5.6.3 Specification of the interoperability demonstration

The Card Terminal API

See section 5.5.3

Card types and communication protocols to be supported at the Card Interface

See section 5.7.3

Card readers and computer operating systems

The range of card readers used for interoperability demonstrations may be limited by the availability of driver software that provides the required CT-API. However, appropriate software will be developed for at least one type of card reader in use in each project.

5.6.4 Existing work and required development

The CardLink card reader interface is only currently operational with asynchronous cards with the T=0 transmission protocol. One the other hand, the DiabCard card reader interface only supports the use of T=1 cards.

Software will need to be developed to provide the agreed Card Terminal API for the card readers used at the sites where interoperability is to be demonstrated. The Card Interface Environment will need to support T=0 and T=1 cards in the same card readers.

5.7 Card Interface

5.7.1 Description

Definition

The Card Interface is the physical connection and communication protocol between the card and Card Terminal.

Role

Several protocols are used for communicating with cards. These depend on the physical characteristics of the card and the medium used for storage of data on a card. The current work focuses on IC cards rather than cards that use other media (e.g. magnetic and optical cards are not considered).

IC cards may either communicate synchronously or asynchronously. Cards that communicate asynchronously must incorporate a microprocessor that manages access to the memory with the card. Cards that communicate synchronously do not require a processor as their memory is directly addressed by the card access protocol.

5.7.2 Developing interoperability

Points to be agreed

Agreement is required on the card types and communication protocols to be supported by the Card Interface.

Two approaches could be employed to deliver Healthcard System interoperability:

- A single protocol could be agreed for use by all healthcards:
  - This is probably both impractical and undesirable. It would force a decision on a single current generation protocol. If the most basic protocol is specified, this will lock all users into an approach which limits functionality and performance. If a more advanced protocol is chosen, this will impose higher cost cards on implementers even where the objectives are limited and could be met with older, cheaper cards. In either case, migration to new protocols would be incompatible with this approach to interoperability.
- A Card Terminal capable of supporting multiple protocols could be specified for interoperable card systems:
  - A Card Terminal that supports several different protocols provides the opportunity for Healthcard Systems that use different types of card to interoperate. It also allows newer technologies to be accommodated by extension of the Card Terminal to cover a logical extension to the Card Interface without sacrificing existing functionality;
  - The Card Interface specification should include:
    - Support for different card communication protocols;
    - A clearly defined way of determining the protocol required by negotiation with a card inserted in the Card Terminal.

Development required

See section 5.6.2.

5.7.3 Specification of the interoperability demonstration

The interoperability pilot will support the following cards:

- Any cards that fully implement the T=0 protocol;
- IBM cards that support the T=1 protocol;
- Other cards that support the T=1 protocol in a manner that is 100% compatible with IBM T=1 card.

There is concern that some T=1 cards implement different options or different interpretations of the published T=1 protocol. Therefore, there is no guarantee that these cards will all be supported. Particular concern has been expressed about the compatibility between IBM T=1 cards and some Japanese T=1 cards.

5.7.4 Existing work and required development

The Card Interface for contact oriented Integrated Circuit Cards included in part 2 of the Multifunction Card Terminal (HCI-MCT-01) appears to meet the requirements for this interface. It incorporates relevant standards and support for a subset of inter-industry commands. It accommodates IC cards that support the following transmission protocols:

- Asynchronous transmission using character protocol (T=0);
- Asynchronous transmission using block protocol (T=1);
- Synchronous transmission using serial data access protocol (SDAP);
- Synchronous transmission using two wire bus protocol (2WBP);
- Synchronous transmission using three wire bus protocol (3WBP).

It is desirable to support the full functionality of the MCT with respect to different protocols. In the interoperability demonstration only T=0 and fully ISO 7816-3 conformant T=1 cards will be supported. Furthermore, where there are differences in interpretation of ISO7816 by manufacturers, it is possible that some T=1 cards may not be fully supported in this interoperability demonstration.

The required development has been outlined in section 5.6.4.

5.8 Healthcard

5.8.1 Description

Definition

A computer readable card used in connection with the administration or provision of healthcare.

Role

An interoperable healthcard must be accessible using at least one of the communication protocols defined in the Card Interface specification. The previous section discusses the relative merits of a single protocol specified for use by interoperable healthcards and a more flexible approach that supports several widely used protocols.

An interoperable healthcard must also contain information required to support one or more of the functions to which interoperability applies. This information must exist in a form that can be recognised and used by Healthcard Systems that support interoperability.

5.8.2 Developing interoperability

Discussion

Healthcard data structure requirements can be met in one of two ways:

- A single common arrangement of healthcare application files and single common structure of the data within these application files:
  - It is unlikely that this approach will be acceptable and applicable to healthcards issued for different purposes:
    - Even in respect of data sets to which interoperability applies it is inevitable that some Healthcard Systems will require and support levels of detail that are greater than the minimum required for interoperability. Low cost, low capacity cards will be unable to support these levels of detail but it may still be worthwhile to enable sharing of the information that is available;
    - Some Healthcard System will organise information in particular ways to deal with security or functional considerations or to retain compatibility with earlier versions. If interoperability is dependent on a particular file and data structure within the card, it may conflict with these other operational requirements.
- Arrangements of healthcare data within healthcards that are recognised and mapped by Healthcard Server software to common agreed data structures at the Healthcard Server API:
  - This requires that Healthcard Servers modules are available to support each of the healthcard data structures;
  - It also requires an unambiguous method of recognising the differences between the data structures used.

The Healthcard Server approach is the most practical way forward but unnecessary variations in data structure within the card should still be avoided.

Points to be agreed

The following points need to be agreed to enable interoperability between healthcards with different data structures:

- A common mechanism by which any Healthcard Server can identify the card operating system of an inserted card:
  - This can be done by reading the Answer to Reset (ATR) when a card reset is initiated.
- A common mechanism by which any Healthcard Server that supports access to cards with a particular operating system can identify the Card Issuer and hence the healthcard data structure:
  - This requires a common file containing the Card Issuer Identifier and other relevant application identification information to be present on every healthcard. This must be read before the data structure is known so it is essential that this is implemented in the same way on all cards that share the same card operating system.
- The information and functions that are subject to interoperability.

Development required

Healthcard data structures may need to be extended or modified to include the interoperable information if this is not already supported by the existing design. Although the data structure and representation need not be the same on the card as at the Healthcard Server API, it is important to ensure that mapping is possible between these structures.

Healthcard Server software must be developed to map each healthcard data structure to the common Healthcard Server API. Healthcard Server developments are also required to support cards with different operating systems (see 5.4.2)

5.8.3 Specification of the interoperability demonstration

Identifying the card operating system of an inserted card.

This will be done by reading the Answer to Reset (ATR) when a card reset is initiated.

Identify the Card Issuer and hence the healthcard data structure.

A common Card Application Data file will be present on all interoperable cards. This will contain the Card Issuer Identifier and other relevant application identification information. This method of accessing this file must be the same on all cards with a particular operating system. This approach is generally in line with the CardLink specification with some revisions to support a wider range of interoperable applications. However, for cards that cannot be recognised in this way a trial and error approach may be used.

The information and functions that are subject to interoperability;

The interoperability demonstration between CardLink and DiabCard will support read-only access to administrative information and emergency clinical information. Part 3 of this report specifies the data content and structure. The data content to be conveyed is based closely on the CardLink emergency data set. A distinction is made between the information that can cross the Healthcard Server API and information that must be stored on the healthcard.

5.8.4 Existing work and required development

The individual projects already have cards in circulation and other cards due for issue soon. Ideally the data formats used on these cards should be supported by an interoperable Healthcard System. However, some changes will be required to support the proposed set of administrative and emergency information.

Healthcard Server software will need to be developed to support the revised CardLink and DiabCard data structures and card operating system (see 5.4.4).