Technical Document - Confidential

GSM PROTOCOL STACK

G23

ACI – APPLICATION CONTROL INTERFACE

DRAFT DESIGN SPECIFICATION

Document Number:	8411.605.99.100
Version:	0.5
Status:	Draft
Approval Authority:	
Creation Date:	1998-Jul-06
Last changed:	2015-Mar-08 by XGUTTEFE
File Name:	g23acids.doc

Important Notice

Texas Instruments Incorporated and/or its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products, software and services at any time and to discontinue any product, software or service without notice. Customers should obtain the latest relevant information during product design and before placing orders and should verify that such information is current and complete.

All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment. TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI products, software and/or services. To minimize the risks associated with customer products and applications, customers should provide adequate design, testing and operating safeguards.

Any access to and/or use of TI software described in this document is subject to Customers entering into formal license agreements and payment of associated license fees. TI software may solely be used and/or copied subject to and strictly in accordance with all the terms of such license agreements.

Customer acknowledges and agrees that TI products and/or software may be based on or implement industry recognized standards and that certain third parties may claim intellectual property rights therein. The supply of products and/or the licensing of software does not convey a license from TI to any third party intellectual property rights and TI expressly disclaims liability for infringement of third party intellectual property rights.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products, software or services are used.

Information published by TI regarding third–party products, software or services does not constitute a license from TI to use such products, software or services or a warranty, endorsement thereof or statement regarding their availability. Use of such information, products, software or services may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Change History

Date	Changed by	Approved by	Version	Status	Notes
1998-Jul-06	AK/ES/PZ		0.1		1
1998-Sep-09	BO		0.2		2
1998-Oct-07	AK		0.3		3
1999-May-10	AK		0.4		4
2003-May-13	XGUTTEFE		0.5	Draft	

Notes:

1. Initial version

2. Revised
3. Chapters added 4.x.x
4. Document number changed

Table of Contents

List of Figures and Tables

List of References

[ISO 9000:2000] International Organization for Standardization. Quality management systems - Fundamentals and vocabulary. December 2000

1.1 References

[1] Digital cellular telecommunications system (Phase 2);
Use of Data Terminal Equipment - Data Circuit terminating Equipment (DTE - DCE) interface for Short Message Service (SMS) and Cell Broadcast Services (CBS);
GSM 7.05, November 1996, version 4.8.0

[2] Digital cellular telecommunications system (Phase 2); AT command set for GSM Mobile Equipment (ME)
GSM 7.07, May 1996, version 4.1.0

[3] ITU-T Recommendation T.32; Terminals for telematic services;
Asynchronous facsimile DCE control - service class 2
ITU-T.32, 1995

[4] ITU-T Recommendation V.25 ter; Series V: data communication over the telephone network;
Interfaces and voiceband modems; Serial asynchronous automatic dialing and control
ITU-T V.25 ter, 1997

[5] ACI - Application Control Interface, Design Specification,
8411.802.98.100, Condat GmbH, September 1998

[6] ACI - Application Control Interface, AT Command Interface Description,
8410.011.98.102, Condat GmbH, September 1998

[7] ACI - Application Control Interface, Functional Interface Description,
8411.802.98.100, Condat GmbH, September 1998

[8] GTI - Generic Target Interface, Interface Description,
8410.008.98.002, Condat GmbH, August 1998

[9] MNI - Mobile Network Interface, Interface Description,
8410.012.98.100, Condat GmbH, August 1998

1.2 Abbreviations

ACI	Application Control Interpreter (AT-Commands)
DTE	Data Terminal Equipment
MMI	Man Machine Interface

2 Overview

G23 is a software package implementing Layers 2 and 3 of the ETSI-defined GSM air interface signaling protocol, and as such represents that part of a GSM Phase 2 compliant mobile station's protocol software which is both, platform and manufacturer independent. Therefore, G23 can be viewed as a building block providing standardized functionality through generic interfaces for easy integration.

The G23 suite of products consists of the following items:

- Layers 2 and 3 for speech & short message services,
- Layers 2 and 3 for fax & data services,
- Application control interface,
- Basic public MMI [02.30] and
- Test and integration support tools.

This document describes the design specification of the Application Control Interface (ACI).

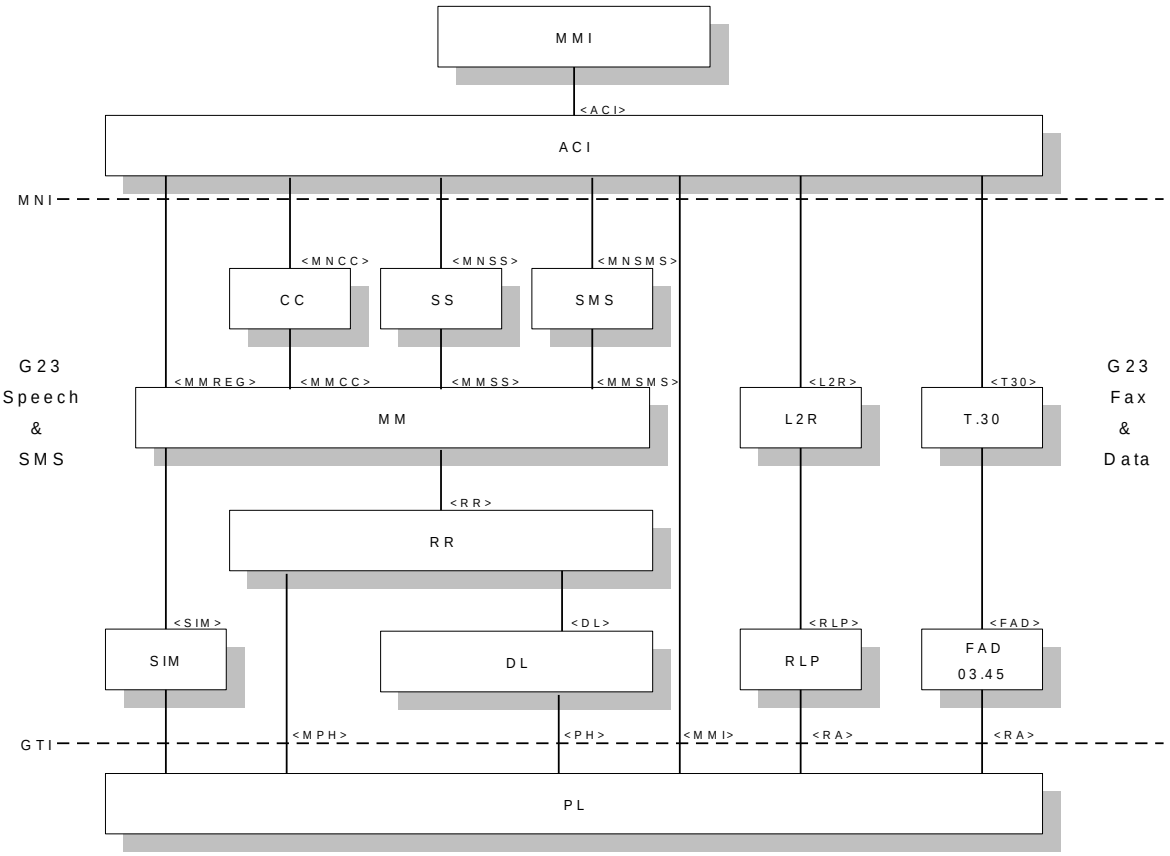

Fig. 1: G23 Software Architecture

3 ACI - Application Control Interface

The ACI, which represents in general Layer 4 functionality, provides the main access to G23. The ACI abstracts from the Layer 3 SAPs MNI and the driver interface of the G23 Protocol Stack.

ACI implements the following functionality:

- transformation of AT-Commands to ACI function calls
- ACI functional interface for the embedded MMI
- interfacing the G23 via MNI
- interfacing to GSM and Non-GSM driver
- phone-book
- SMS and SMS-CB book
- ..
- ..
- ..
- ..
- ..

ACI is embedded in the following interfaces:

- AT-Command interface [6], the use access for remote DTE applications
- ACI functional interface [7], the use access for MMI applications
- network layer interface (MNI) [9], the used relation to the GSM protocol stack
- Generic Target Interface (GTI) [8], the used relation to the GSM and Non-GSM driver

3.1 ACI Software Architecture

The following shows the architecture of the ACI and the functional blocks which are implemented by the entity.

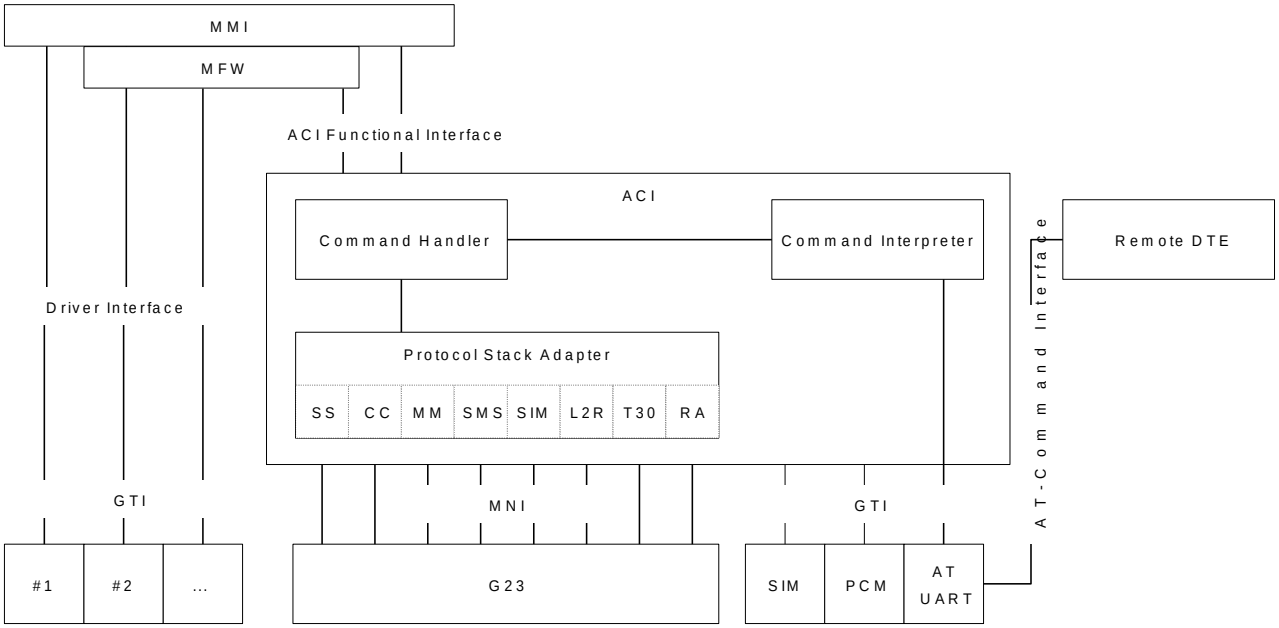

Fig. 2: ACI Software Architecture

The AT-Command Interface supports an access for a Remote DTE. The ACI Functional Interface supports a direct access to the functionality of the ACI entity.

In general driver (LCD, keyboard, battery ...) are calling MFW/MMI directly in order to save resources. If necessary ACI asks MFW/MMI to call ACI-Functions to forward needed driver events via AT-Commands to the Remote DTE. The exception are the driver for permanent memory, SIM and ACI serial interface which are fully controlled by ACI.

ACI is a implemented as a library which is composed of the following function blocks (modules):

- Command Interpreter:
  The Command Interpreter is responsible for handling the AT-Command protocol based on ASCII character strings. The handling includes parsing the ASCII string for AT-Commands and parameters, checking for syntactical correctness and generating respective result messages.
  The Remote DTE sends AT-Commands to the ACI. The ACI Command Interpreter checks the commands and performs the requested ACI-Operation. The result is given back to the Remote DTE as AT-Commands. Additionally, some unsolicited results from ACI are performed as AT-Commands (e.g. notification of an incoming call, indication of a network change ...).

- Command Handler:
  The Command Handler is responsible for the execution of the invoked AT commands. The handler provides a functional interface to the above layer. Therefore every AT command has it's corresponding functional counterpart, which is responsible for the execution of the command. A simple management concept is implemented by the Command Handler to avoid collision in case that the MMI/MFW are accessing services or transactions of the Protocol Stack simultaneously.

- Protocol Stack Adapter:
  The Protocol Stack Adapter is used for canalizing the command requests to their respective protocol stack entities. Therefore every entity of the GSM Protocol Stack has a corresponding adapter module which is responsible for the communication with the entity. The adapter also knows the current settings for the entity and provides extended functionality

  PSA CC:
  The adapter for the Call Control entity provides a call table which is capable of handling up to 7 mobile originated and up to 7 mobile terminated calls. Also supported are the functionality's of call hold and call retrieve and multiparty calls. Both type of calls, voice and data, are handled, as well as the toggling between them by using an in-call modification.

  PSA MM:
  The Mobility Management entity adapter supports the functionality's related to the network registration procedures. This includes the modes of automatic and manual registration, the capability to search for the available operators and the handling of the list of found PLMN's

  PSA SS:
  Like the PSA CC the adapter for the Supplementary Services entity provides a table for handling up to 7 mobile originated and up to 7 mobile terminated service transactions for call independent services. This includes facilities like call forwarding and call barring.

  PSA SIM:
  The adapter for the Subscriber Identity Module entity detects the status of the SIM card. This includes whether the card is inserted or removed and if the input of security codes like PIN and PUK are required. The adapter also supports the access to the different types of information's which are stored on the SIM card. Complex data fields like a phone-book are copied into a RAM area where they can be sorted for a efficient way of getting access to it.

  PSA SMS:
  The Short Message Service adapter supports the functionality's related to the exchange of short messages. This includes the sending of MO and command messages, the receiving of MT, CB and status messages, the storing and deletion of messages and the configuration of the SMS entity.

  PSA L2R:
  TBD

  PSA T30:
  TBD

  PSA RA:
  TBD

4 ACI Interfacing

As outlined above there are two principal ways to use ACI. An external Remote DTE uses the AT-Command Interface. An internal MMI uses the Functional Interface to save resources.

4.1 AT-Command Interface

The GSM standardized AT-Commands are specified in GSM 07.07 [2] and V25ter [4]. In addition, some AT-Commands are supported to use the Short Message Entity (SMS), described in GSM 07.05 [1]. To use the FAX functionality, the

AT-Command set for a Class 2.0 FAX Modem is implemented as described in CCITT-T.32 [3]. These documents set the standards for our implementation.

The AT-Commands supported by ACI is specified in the document "ACI- Application Control Interface, AT-Command Interface Description" [6].

4.2 ACI Functional Interface

The Functional Interface supported by ACI is specified in the document "ACI- Application Control Interface, Functional Interface Description" [7].

4.3 MNI Primitive Interface

ACI uses the Primitive Interface supported by the G23 Protocol Stack entities and specified in the document "MNI - Mobile Network Interface, Interface Specification [9].

4.4 Driver Interface

ACI uses the Driver Interface (Permanent Memory, SIM and ACI-UART) supported by G23 and specified in the document "GTI - Generic Target Interface, Interface Description [8].

4.5 Operating System Interface

ACI uses the functionality of the Virtual Operating System supported by G23 and specified in the document "GTI - Generic Target Interface, Interface Description [8].

5 Module Architecture

The following is a brief description of the architecture for the different modules of the ACI. The ACI consists of the three modules Command Interpreter (CI), Command Handler (CH) and Protocol Stack Adapter (PSA), and the two Interfaces between CI ⇔ CH and CH ⇔ PSA.

5.1 Command Interpreter

The Command Interpreter is responsible to implement the ASCII based AT-Interface described in [1],[2],[3] and [4]. Therefore the CI has to parse an incoming command line for syntactical correct AT-Commands. These commands are then sequentially transferred to the CH for further execution. The result of a command which is provided by the CH has to be converted by the CI to meet the above specifications and is send as a command reply to the user. Commands which do not agree with the AT-Command syntax are rejected by the CI immediately. Mode parameter of AT-Commands which are used to set the behavior for unsolicited result codes, and AT-Commands which are used to test the possibilities of a command, have to be fully handled by the CI, because this is not supported by the CH. Driver related AT-Commands (e.g. keyboard, display, indicator) are also not handled by the CH. Therefore the CI has to interact with the MMI/MFW to process these commands, because the MMI/MFW has the control of these drivers. Another fact which is not captured by the CH is the interpretation of keystroke sequences. If the dial-string of the dial command contains keystroke sequences, the CI has to convert the sequences by a common decoder, and call the respective function of the CH to perform the desired action.

In detail the CI scans a command line for a AT-Command body and verifies this substring with the table of known AT-Commands. If the command was found, a dedicated handler function for that command is called, passing the rest of the command string as a parameter. This function is responsible to check for valid parameters of that command in detail. The test option for a command has to be handled by that function, and results in a string giving the possible settings for the different parameters of that command. If no syntactical error was found the CI calls the respective function of the CH by passing the command parameters. Results send by the CH have to be checked whether they are requested or not and should then be converted and forwarded to the user.

5.2 Command Handler

The Command Handler is responsible for the processing of the AT-Commands. To invoke AT-Commands the CH has to be accessed using the functional ACI-Interface. For every AT-Command exists a functional counterpart which has to be called

to perform the action. The AT-Command options are passed using the parameters of respective function calls. The CH verifies these parameters for syntactical correctness and reject the command with a failure indication in case that a parameter is unknown or not supported by the CH. Parameters which are declared as not present are substituted by former entered values or by defaults if no value was given.

Due to the possibility that the CH can be accessed both by the MMI/MFW and the CI, an simple mechanism is implemented by the CH to avoid a collision. The following describes the principles used by that mechanism. For more clarification of that, the term local is used to identify the MMI/MFW if accessing the CH and the term remote identifies a DTE which accesses the CH via the CI.

Every service or transaction of the Protocol Stack entities is an indivisible Object and cannot be shared by the two users of the CH. Therefore these objects are dedicated to one user (local or remote) at a time. The objects which are related to a transaction are dynamic. That means, that they are created, modified and deleted by a user. (e.g. call and supplementary service transactions) During the life-cycle of an object, only the user who has created the object has the right to control the object. Protocol Stack services are therefor more static objects. They exist only once in the system and do not need to be created. These objects are not specially dedicated to one user. But if one user has started an operation on that object, the other user is blocked for that object until the operation is completed.

To manage this behavior of the CH, each caller of a function has to identify itself by a source parameter. During the creation of a dynamic object or the start of an operation for a static object, this source identifier is stored for that object, to define the owner. If the local or remote user wants to control an object that he don't owns, the command will be rejected. On the other hand, if the Protocol Stack generates messages which are related to an object, the source identifier determines the owner of that object and to whom the responses have to be send.

In addition to that the CH stores the actual configuration for each user individual. Therefore the settings for local user (e.g. bearer capabilities, type of address, etc.) do not interfere with the settings done by the remote user.

Due to these architectural principles the CH can be operated by a remote and a local user widely in parallel. Results and indications which are primary related to the remote user are also passed to the local user for information. Therefore the local user is able to indicate or to keep track of the actions done by the remote interface.

5.3 Protocol Stack Adapter

The PSAs adapt the Protocol Stack entities to the CH. Every single entity of the G23 MN-Interface has its own Protocol Stack Adapter to communicate with. The corresponding PSA and PS entity exchange their information's using a primitive interface. If an action has to be performed by a PSA, a dedicated function takes the necessary parameters for that action, fills in the primitive buffer and sends it to the desired entity. On the other hand, if a primitive is received by a PSA, an related function is called which has the knowledge of the primitive content. This function interprets the contents of the primitive and updates the current status for that entity. The primitive interface of the entities is abstracted and summarized if possible to simplify the access by the CH. In addition to that, some entities provide an extended functionality by the implementation of supporting and utilizing functions.

5.3.1 PSA Mobility Management

The PSA Mobility Management provides the following functionality to CH. Start a network registration, stop a network registration, request the list of available networks and select a special network. Besides that the CH will be informed if the mobile is registered to the network, de-registered to the network and if a network should be selected. The current status of the Mobility Management entity informs the CH about the list of available networks, the used network, the registration mode, the registration status, the cause for a de-registration and the network search result. In addition to that, a user-customizable list of known networks is supported, which declares different representations for each network. This includes string representations for a short and a long network name and a numeric string build out of the mobile country code and the mobile network code. Different accessing function are provided for the CH to search the list for the different representations.

5.3.2 PSA Call Control

The PSA Call Control provides the following functionality to CH. Originate a new call, indicate alerting, accept an incoming call, clear a call, release a call, hold a call, retrieve a call, modify a call, send user data, send DTMF-tones, use call-dependent supplementary service and notify the network. Besides that the CH will be informed if a call is connected or disconnected, about the call proceeding, about an incoming call or a missed call, if the called party is alerted, if a call is held or retrieved, if the call is modified, about the receiving of user data, the confirmation of DTMF-tones and about network notifications. In addition to that the PSA for Call Control manages a call table which is capable of handling up to 7 mobile originated and 7 mobile terminated calls. For each call stands an entry of the call table which gives full information about the current call status. This includes the transaction identifier, the state, type, priority and the bearer capabilities for the call, the progress

description, the description for the called and the calling party, user data, network notifications, facility information and failure causes. Utility functions are provided by the PSA to simplify the use of the call table for the CH. This includes searching for calls, dial-string to call address conversions, transaction identifier assignment, allocation and de-allocation of call table entries and the assembling of call related supplementary service messages. Independent of a special call informs the current status of the Call Control entity about the number of existing mobile originated and mobile terminated calls, about the TCH assignment and about the last missed call due to a full call table.

5.3.3 PSA Subscriber Identity Module

The PSA Subscriber Identity Module (SIM) provides the following functionality to CH. Verify, change, enable and disable a PIN, unblock the SIM card, synchronize the SIM, activate the SIM, read and write data of a special file in binary mode, read, write and search a record of a special file in absolute or relative access mode, increment the data of a special file. Besides that the CH will be informed if a PIN is verified, changed, enabled or disabled, if the SIM card is unblocked, the SIM is synchronized inserted or activated, if binary data was read or written, if record data was read, written or found or if data was incremented. The current status of the SIM entity informs the CH about the PIN status, the chipcard identification, the phase of the card, the SIM status and the last datafield and data which was read, written or searched. In addition to that the PSA for SIM manages a phone-book which is sorted by name and by number to allow quick access to the different search requests. The phone-book is a summary of the phone-books found in the SIM and in the Permanent Configuration Memory. Different accessing function are provided for the CH like read, write and search a phone-book entry.

5.3.4 PSA Supplementary Service

The PSA Supplementary Service provides the following functionality to CH. Originate a new service, clear a service and transfer information for an existing service. Besides that the CH will be informed about an incoming service, about a cleared service, about received information's for an exiting service and about service failures. In addition to that the PSA for Supplementary Service manages a service table which is capable of handling up to 7 mobile originated and 7 mobile terminated services. For each service stands an entry of the service table which gives full information about the current service status. This includes the transaction identifier, the state and type of the service, the service version, the facility information and failure causes. Utility functions are provided by the PSA to simplify the use of the service table for the CH. This includes searching for services, transaction identifier assignment, allocation and de-allocation of service table entries and the assembling of supplementary service messages. Independent of a special service informs the current status of the Supplementary Service entity about the number of existing mobile originated and mobile terminated services.

5.3.5 PSA Short Message Service (SMS)

The PSA Short Message Service provides the following functionality to CH. Send, read, store and delete a message, send a command message and configure the behavior on incoming SMS, which includes the selection of the preferred memory types and the management of mobile terminated, cell broadcast and status report messages. Besides that the CH will be informed about an incoming message, an incoming status report, about previously requested mobile originated, mobile terminated or cell broadcast messages and failure causes. The current status of the SMS entity informs the CH about the configuration of the preferred memory types, the mode for mobile terminated messages, the mode for cell broadcast messages, the mode for status report messages, the last send mobile originated message and the last send command message. In addition to that, the PSA for SMS manages a list of the available SMS messages. This is a copy of the stored SMS messages of the SIM card. The list is sorted to allow a quick access by the CH. Different accessing function are provided for the CH like read, write and search a message entry.

5.3.6 PSA Rate Adaptation (RA)

To be defined.

5.3.7 PSA Layer 2 Relay Functionality (L2R)

To be defined.

5.3.8 PSA Fax Protocol Entity (T.30)

To be defined.

5.4 CI ⇔CH Interface

The Command Interpreter (CI) uses the ACI functional interface defined in [7] to access the Command Handler (CH). This interface represents the different AT-Commands by providing separate functions to set, query or test for parameters. If the operation which was invoked by the function call was completed immediately, the function return code indicates the result of the operation. In case that the command could not be completed immediately, e.g. due to an asynchronous interaction with the Protocol Stack entities, the function return code indicates that the operation is still be executed. As soon as the operation is completed, the result will be passed via a call-back mechanism. For every function call of the CH, the CI has to identify itself by setting the source identifier to remote. This is important to avoid collision with the MMI/MFW which also uses this interface.

5.5 CH ⇔PSA Interface

The Command Handler (CH) interfaces with the Protocol Stack Adapters (PSA) via shared memory areas. Every PSA communicates with the CH using a private memory area. This area is used to store all relevant data which has to be exchanged between the specific PSA and the CH.

The PSA are includes two subareas which are provided to store the current setting for the local and the remote user separately. The CH updates these settings according to the source identifier, which is passed during every CH function call. Another subarea is included in the PSA area, which stores the current status of the PSA entity. If the PSA receives primitives from its associated entity, the PSA status will be updated.

If the CH wants to use a functionality of the PSA, it has to update the setting for the current user (local/remote). After that the CH calls a command function of the PSA and indicates the command type and the current setting that the PSA has to work with. On the other hand, if the PSA entity has changed its state, it updates the PSA status for that entity. The PSA then notifies the CH by calling a function of the CH and indicates the type of event which has occurred.

See the following figure for an overview of the memory areas and their usage.

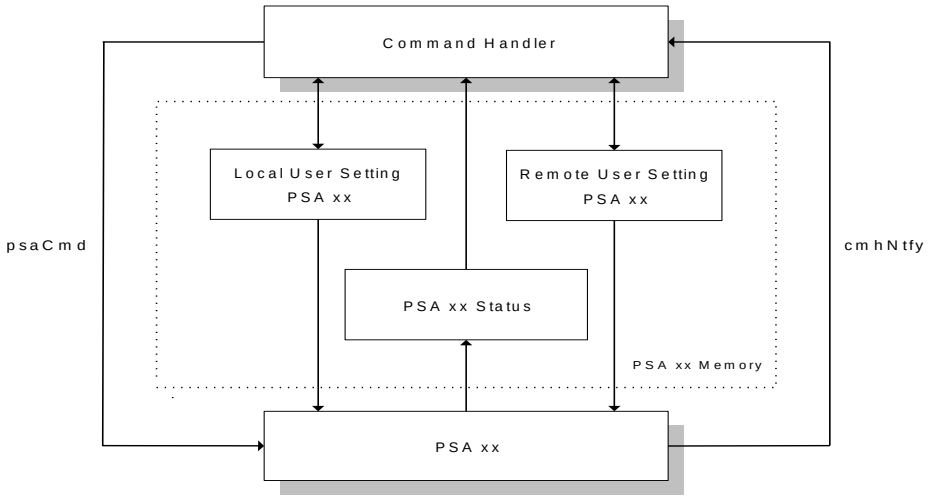

Fig. 3: PSA Memory Areas

A.

Appendices

A. Acronyms

DS-WCDMA	Direct Sequence/Spread Wideband Code Division Multiple Access

B. Glossary

International Mobile Telecommunication 2000 (IMT-2000/ITU-2000)	Formerly referred to as FPLMTS (Future Public Land-Mobile Telephone System), this is the ITU's specification/family of standards for 3G. This initiative provides a global infrastructure through both satellite and terrestrial systems, for fixed and mobile phone users. The family of standards is a framework comprising a mix/blend of systems providing global roaming. <URL: http://www.imt-2000.org/>