

Software System Development for Spacecraft Data Handling & Control

Data Handling System Software Requirements Document

<i>TERMA Doc. no.:</i>	<i>TERMA/OBOSS-II/TN/010</i>
<i>Date:</i>	<i>06.05.99</i>
<i>Issue:</i>	<i>1</i>
<i>Revision:</i>	<i>A</i>
<i>Distribution:</i>	<i>ESTEC, Dornier, TERMA Elektronik</i>
<i>Prepared by:</i>	<i>Jan Storbank Pedersen</i>
<i>Approved by:</i>	<i>Gert Caspersen</i>
<i>Authorised by:</i>	<i>Gert Caspersen</i>

The intellectual property right of this document is vested in TERMA Elektronik AS.

Document Change Record

Issue	Date	Change
1	12.03.99	Initial issue of document
1A	06.05.99	Updated to reflect ESTEC review comments

Document Status Sheet

Page	Issue
i-vi	1A
1-14	1A

Table of Contents

1	Introduction	1
1.1	Scope	1
1.2	Abbreviations and Acronyms	1
1.3	Outline of Document	1
2	Documents	3
2.1	Applicable Documents	3
2.2	Reference Documents	3
3	General Description	4
3.1	Function and Purpose	4
3.2	Environmental Considerations	4
4	Specific Requirements	5
4.1	Functional Requirements	5
4.1.1	Telecommands and Telemetry	5
4.1.2	Telecommanding	5
4.1.3	Telecommand Transmission and Distribution	6
4.1.4	Telecommand Verification	6
4.1.5	Satellite Status Reporting/Telemetry	7
4.1.6	Telemetry Timing	7
4.1.7	Memory Management	8
4.1.8	Function Management	8
4.1.9	On-board Scheduling	8
4.1.10	On-board Monitoring	9
4.1.11	On-board Storage and Retrieval	10
4.1.12	On-board Traffic Management	10
4.1.13	Packet Routing	11
4.2	Performance Requirements	11
4.3	External Interface Requirements	11
4.4	Operational Requirements	12
4.5	Resource Requirements	12
4.5.1	Hardware	12
4.5.2	Software	12
4.6	Verification Requirements	12
4.7	Acceptance Testing Requirements	12
4.8	Documentation Requirements	12
4.9	Security Requirements	13
4.10	Quality Requirements	13

Data Handling System Software Requirements Document

4.11 Reliability Requirements	13
4.12 Maintainability Requirements	13
4.13 Reusability Requirements	13
4.14 S/W Design and Programming Requirements	14
4.14.1 Design Standards	14
4.14.2 Programming Standards	14

1 Introduction

1.1 Scope

This document is the software requirements document for a service layer that constitutes part of an onboard data handling software system. An initial version of the software was developed as part of the On-Board Operations Support Software (OBOSS) ESA project on a 1750A platform. The aims of the current development are to port the initial version to an ERC32 platform, make the design HRT-HOOD compliant and to extend the functionality. The present document defines the requirements to the extended system.

1.2 Abbreviations and Acronyms

ESA	European Space Agency
ESF	Execution Skeleton File
HRT-HOOD	Hard Real-Time HOOD
HOOD	Hierarchical Object-Oriented Design
ICU	Intelligent Control Unit
ITT	Invitation To Tender
OBDH	On-Board Data Handling
OBOSS	On-Board Operations Support Software
OBOSS-II	Software System Development for Spacecraft Data Handling & Control
PUS	Packet Utilisation Standard
RTU	Remote Terminal Unit
SVF	Software Validation Facility
TC	Telecommand
TM	Telemetry
UCF	User Configuration File
UART	Universal Asynchronous Receiver and Transmitter

1.3 Outline of Document

The structure and contents of the document is as follows.

Data Handling System Software Requirements Document

Section 2 enumerates applicable and reference documents.

Section 3 defines the general requirements.

Section 4 defines the specific requirements.

2 Documents

2.1 Applicable Documents

[PSS-05] ESA Software Engineering Standards, ESA PSS-05-0, February, 1991

[Guidelines] Guidelines for the Development of Reusable Software, CRI/OOSS/WP-1.2.1/001/1.C, March, 1996

[ITT] Software System Development for Spacecraft Data Handling and Control, AO/1-3282/97/NL/PA, ESTEC

2.2 Reference Documents

[Ada-83] Reference Manual for the Ada Programming Language, United States Department of Defense, ANSI/MIL-STD-1815A-1983, 1983

[PUS] Packet Utilisation Standard, ESA, PSS-07-101, Issue 1, 1994

[Dom] SW System Development for SC Data Handling and Control, Domain Analysis, DOR-OBOII-98-0001, Issue C, February 1999

[Formats] Software System Development for Spacecraft Data Handling & Control, Representation of Telecommand and Telemetry Packets, TERMA/OBOSS-2/TN/007, Issue 1.2, February 1999

[Sched] 32-Bit Microprocessor and Computer System Development, Schedulability Analyser and Scheduler Simulator Software User Manual, 32B-SBI-SUM-0189-001/2, Spacebel Informatique S.A., 1996

3 General Description

3.1 Function and Purpose

The onboard operation support software (OBOSS) will constitute a largely mission-independent set of services that implement major parts of the Packet Utilisation Standard [PUS]. The section is based on the analysis presented in [Dom]. The on-board software will manage the treatment of telecommand packages and the production of telemetry packets. The overall functionality to be implemented can be divided into the following major areas, reflecting the structuring of the operational requirements defined in [PUS]:

- Telecommanding
- Telecommand transmission
- Telecommand verification
- Satellite status reporting/telemetry
- Telemetry timing
- Memory management
- Function management
- On-board scheduling
- On-board storage and retrieval
- On-board traffic management
- Packet routing

3.2 Environmental Considerations

In order to demonstrate the feasibility of the services provided by the OBOSS, a number of typical, mission-specific properties of medium-size satellites will be used. Some of the properties are explicitly stated as requirements in section 4.

4 Specific Requirements

This section contains the actual requirements on the software to be developed as part of the OBOSS-II project.

4.1 Functional Requirements

This section contains the functional requirements.

Where relevant, requirements contain an explicit reference to the (sub-)sections or requirements within [PUS] to which they are related or to properties within a table of [Dom].

4.1.1 Telecommands and Telemetry

In order to facilitate a uniform treatment of packets a set of overall requirements are established.

Req-4.1.1.1 All telecommands shall contain an unambiguous identification of their source.

Req-4.1.1.2 All telemetry packets shall include an unambiguous identification of their destination.

Req-4.1.1.3 Any telemetry or telecommand packet shall belong to a class identified by a type and a subtype.

Req-4.1.1.4 Extractors and constructors shall be provided for all packet classes and parameter structures based on parameter specifications provided in the data description language.

4.1.2 Telecommanding

Req-4.1.2.1 Each onboard device shall be individually commandable from ground (TC-3 of [PUS]).

Req-4.1.2.2 It shall be possible to access any on-board device as a virtual device using telecommands.

4.1.3 Telecommand Transmission and Distribution

Req-4.1.3.1 The on-board software shall not allow the transmission of a higher priority telecommand to interrupt the transmission of a lower priority one (Table 3-2 (2.1-2.2) of [PUS]).

Req-4.1.3.2 It shall be possible to distribute one or more on/off commands to specific addresses (6.3.1 of [PUS]).

Req-4.1.3.3 It shall be possible to distribute one or more register load commands to specific addresses (6.3.2 of [PUS]).

4.1.4 Telecommand Verification

A telecommand will explicitly specify the stages, if any, at which its execution is to be verified. The possibilities are any combination, including none, of:

- acceptance
- start of execution
- progress of execution
- completion

Req-4.1.4.1 Telecommands shall be verified according to the verification scheme defined by the telecommand (TCVERIF-1, TCVERIF-2.2 of [PUS]).

Req-4.1.4.2 In order to facilitate error handling on ground, a globally unique set of error codes used to indicate failure shall be applied to the entire on-board system.

Req-4.1.4.3 Telecommand verification packets shall only be sent to the source of the telecommand.

Req-4.1.4.4 Verification of start of execution shall include checking of any invariant applying to parameter values in the command.

Req-4.1.4.5 A failed telecommand verification shall be reported (TCVERIF-2.1 of [PUS]).

Req-4.1.4.6 Verification of telecommand acceptance shall include the checking of the service type/subtype and an error control as specified by the checksum type of the telecommand.

4.1.5 Satellite Status Reporting/Telemetry

Req-4.1.5.1 It shall be possible to define and re-define housekeeping report contents (7.3.1 and 7.3.2 of [PUS]).

Req-4.1.5.2 It shall be possible to enable and disable the sending of different housekeeping reports (7.3.3 of [PUS]).

Req-4.1.5.3 It shall be possible to get a report containing the housekeeping parameter definitions (7.3.4 of [PUS]).

Req-4.1.5.4 It shall be possible to choose periodic reporting of housekeeping data (7.3.6 of [PUS]).

Req-4.1.5.5 It shall be possible to choose reporting of filtered housekeeping data (7.3.6 of [PUS]).

Req-4.1.5.6 It shall be possible to get a report containing the masked housekeeping parameters (7.3.7 of [PUS]).

Req-4.1.5.7 It shall be possible to define and re-define diagnostics report contents (7.3.1 and 7.3.2 of [PUS]).

Req-4.1.5.8 It shall be possible to enable and disable the collection of diagnostics (7.3.3 of [PUS]).

Req-4.1.5.9 It shall be possible to get a report containing the diagnostics parameter definitions. (7.3.4 of [PUS]).

Req-4.1.5.10 It shall be possible to choose periodic reporting of diagnostics data (7.3.6 of [PUS]).

Req-4.1.5.11 It shall be possible to choose reporting of filtered diagnostics data (7.3.6 of [PUS]).

4.1.6 Telemetry Timing

Req-4.1.6.1 Telemetry timing shall report the satellite time which is based on a single master clock (TIMING-2 of [PUS]).

Req-4.1.6.2 All telemetry shall be time-stamped using satellite on-board time.

4.1.7 Memory Management

Req-4.1.7.1 It shall be possible to update the contents of any writable on-board memory on request.

Req-4.1.7.1.1 It shall be possible to load data using an absolute address (10.3.2 of [PUS]).

Req-4.1.7.2 It shall be possible to dump the contents of any on-board memory on request.

Req-4.1.7.2.1 It shall be possible to dump data using an absolute address (10.3.4 of [PUS]).

Req-4.1.7.3 It shall be possible to check the contents of any memory area.

Req-4.1.7.3.1 It shall be possible to check one or more memory areas using an absolute address (10.3.6 of [PUS]).

4.1.8 Function Management

Req-4.1.8.1 It shall be possible to request the execution of an activity of a software task (11.3.6 of [PUS]).

Req-4.1.8.2 It shall be possible to deactivate an application function (12.3.2 of [PUS]).

Req-4.1.8.3 It shall be possible to request the execution of an activity of an application function (12.3.3 of [PUS]).

4.1.9 On-board Scheduling

Req-4.1.9.1 The on-board software shall support the concept of interlocking commands in a macro-schedule (or sub-schedule).

Req-4.1.9.2 It shall be possible to enable the release of selected telecommands at the level of macro-schedules and sub-schedules with the optional indication of the destination application process identification (15.3.1.1 of [PUS]).

Req-4.1.9.3 It shall be possible to disable the release of selected telecommands at the level of macro-schedules and sub-schedules with the optional indication of the destination application process identification (15.3.1.1 of [PUS]).

- Req-4.1.9.4** It shall be possible to reset the command macro-schedule (15.3.1.2 of [PUS]).
- Req-4.1.9.5** It shall be possible to insert telecommands into the command macro-schedule or sub-schedule by identifying a macro-scheduling event and possibly an execution timeout (15.3.2 of [PUS]).
- Req-4.1.9.6** It shall be possible to delete a set of specified telecommands from a command macro-schedule (15.3.3.1 of [PUS]).
- Req-4.1.9.7** It shall be possible to delete telecommands from a command macro-schedule or sub-schedule based on sub-schedule and time interval information (15.3.3.2 of [PUS]).
- Req-4.1.9.8** It shall be possible to time-shift one or more specified telecommands in a macro-schedule (15.3.4.1 of [PUS]).
- Req-4.1.9.9** It shall be possible to time-shift one or more telecommands in a macro-schedule based on time interval information (15.3.4.2 of [PUS]).
- Req-4.1.9.10** It shall be possible to get a detailed report on the contents of the command macro-schedule and sub-schedules (15.3.5.1 of [PUS]).
- Req-4.1.9.11** It shall be possible to get a summary report on the contents of the command macro-schedule and sub-schedules (15.3.5.2 of [PUS]).

4.1.10 On-board Monitoring

The values of parameters can be monitored against limits specified by ground. The monitoring is based on a monitoring list that for each relevant parameter describes the checks to be performed, e.g. limits, deltas or expected value.

- Req-4.1.10.1** It shall be possible to enable the monitoring of specific parameters (16.3.1 of [PUS]).
- Req-4.1.10.2** It shall be possible to disable the monitoring of specific parameters (16.3.1 of [PUS]).
- Req-4.1.10.3** It shall be possible to add parameters to the monitoring list and specify the particular checks to be performed (16.3.4 of [PUS]).
- Req-4.1.10.4** It shall be possible to delete parameters from the monitoring list (16.3.5 of [PUS]).

Req-4.1.10.5 It shall be possible to clear the monitoring list (16.3.3 of [PUS]).

Req-4.1.10.6 It shall be possible to modify the parameter check information for a parameter in the monitoring list (16.3.6 of [PUS]).

Req-4.1.10.7 An out-of-limits report shall be generated when a monitored parameter fails to meet the checks specified by the active parameter check specification.

4.1.11 On-board Storage and Retrieval

The on-board software may contain a number of uniquely identified packet stores in which packets of various specified types (and subtypes) can be stored for later processing/transmission.

Req-4.1.11.1 It shall be possible to enable and disable the use of a named packet store (19.3.1 of [PUS]).

Req-4.1.11.2 It shall be possible to add packet types and subtypes to the definition of packets to be stored in a particular packet store (19.3.2 of [PUS]).

Req-4.1.11.3 It shall be possible to remove packet types and subtypes from the definition of packets to be stored in a particular packet store (19.3.2 of [PUS]).

Req-4.1.11.4 It shall be possible to get a list of the packet types and subtypes to be stored in a particular packet store (19.3.3 of [PUS]).

Req-4.1.11.5 It shall be possible to downlink a range of packets from a packet store (19.3.4 of [PUS]).

Req-4.1.11.6 It shall be possible to downlink packets containing a time stamp falling in a specified interval from a packet store (19.3.5 of [PUS]).

Req-4.1.11.7 It shall be possible to delete packets up to a specified packet from a packet store (19.3.6 of [PUS]).

Req-4.1.11.8 It shall be possible to delete packets containing a time stamp falling in a specified interval from a packet store (19.3.7 of [PUS]).

4.1.12 On-board Traffic Management

Req-4.1.12.1 Problems in re-assembling telecommands shall not be reported.

Req-4.1.12.2 Problems in packet distribution shall be reported (TRAFF-3 of [PUS]).

4.1.13 Packet Routing

Req-4.1.13.1 Packet routing shall be based on the application process identifier (ROUTE-1 of [PUS]).

Req-4.1.13.2 Packets to be routed within the data handling computer shall be represented in *internal onboard format* as defined in [Formats].

4.2 Performance Requirements

Req-4.2.0.1 The on-board software shall be able to handle any supported telecommand (from reception of TC to transmission of execution completion verification TM) within 2 minutes.

Req-4.2.0.2 The on-board software shall be able to handle any non-supported telecommand (from reception of TC to transmission of acceptance failure verification TM) within 30 seconds.

Req-4.2.0.3 The on-board software shall be able to perform monitoring of up to 100 parameters with a period of 1 second.

Req-4.2.0.4 The on-board software shall be able to produce out-of-limit reports (for up to 100 parameters) with a period of 10 seconds.

Req-4.2.0.5 The on-board software shall be able to produce housekeeping reports (for up to 100 parameters) with a period of 2 seconds.

Req-4.2.0.6 The on-board software shall be able to perform on-board scheduling with a period of 1 second, for a schedule with up to 50 telecommands.

4.3 External Interface Requirements

Req-4.3.0.1 The structure and layout of external telecommand and telemetry packets shall be as prescribed in [PUS] and referred to as *external ground format* in [Formats].

Req-4.3.0.2 Telecommand and telemetry packets passed between the data handling system computer and other physical subsystems, e.g. ICUs, shall be represented using the *external onboard format* as described in [Formats].

4.4 Operational Requirements

Req-4.4.0.1 The on-board software shall be able to handle incoming telecommands at the rate of one TC per second.

Req-4.4.0.2 To prevent bursts of TCs, any two commands placed on the on-board schedule shall have (actual) schedule times that differ by at least 1 second.

4.5 Resource Requirements

4.5.1 Hardware

Req-4.5.1.1 The on-board software shall run on an ERC32 processor ([ITT]).

Req-4.5.1.2 The on-board software shall be able to execute within 4 Mbyte of RAM (code + data).

4.5.2 Software

Req-4.5.2.1 The on-board software shall be developed in Ada as defined by [Ada-83].

Req-4.5.2.2 The Alsys Ada cross compiler for ERC32 shall be used ([ITT]).

4.6 Verification Requirements

Req-4.6.0.1 The verification of the on-board software shall be performed using a software validation facility (SVF). The detailed interfaces will be described later in the development.

4.7 Acceptance Testing Requirements

The acceptance testing will be based on a demonstration scenario that is still to be defined within the project.

4.8 Documentation Requirements

Req-4.8.0.1 The on-board software shall be documented in a reusers manual.

Req-4.8.0.2 The reusers manual shall be available in an electronic form that utilizes hyper-links.

Req-4.8.0.3 The documentation shall include a description of a demonstration data handling system constituting an instantiation of the reusable software.

Req-4.8.0.1 To support a schedulability analysis, the demonstration data handling system documentation shall include the corresponding ESF and UCF files, see [Sched] for details on the file formats.

4.9 Security Requirements

None.

4.10 Quality Requirements

None.

4.11 Reliability Requirements

None.

4.12 Maintainability Requirements

None.

4.13 Reusability Requirements

The guidelines of [Guidelines] lead to certain specific requirements related to reuse, in addition to those requirements stated elsewhere in this requirements document.

Req-4.13.0.1 Parameterised design objects and implementation objects shall be used to the extend feasible.

Req-4.13.0.2 External interfaces shall be isolated in designated components.

Req-4.13.0.3 The software design shall where feasible be organised in a layered structure.

Req-4.13.0.4 Parts of a system that are likely to change shall be isolated in designated components.

Req-4.13.0.5 Machine dependent code shall be isolated from the rest of the code.

Req-4.13.0.6 Properties of types shall be obtained by using attributes.

4.14 S/W Design and Programming Requirements

4.14.1 Design Standards

Req-4.14.1.1 The design shall be expressed in HRT-HOOD.

4.14.2 Programming Standards

Req-4.14.2.1 The mapping of the HRT-HOOD design into Ada shall use a tasking structure that only implements *cyclic*, *sporadic* and *protected* objects.

Req-4.14.2.2 *Protected* objects shall, if feasible, be implemented using 'passive' tasks as supported by the Alsys Ada compiler.