DEUTSCHE NORM

Dezember 2014

DIN EN 16603-70-32

DIN

ICS 49.140

Raumfahrttechnik –
Sprache für Test- und Bedienprozeduren;
Englische Fassung EN 16603-70-32:2014

Space engineering –
Test and operations procedure language;
English version EN 16603-70-32:2014

Ingénierie spatiale –
Language de procedure pour les essais et des operations;
Version anglaise EN 16603-70-32:2014

Gesamtumfang 143 Seiten

DIN-Normenausschuss Luft- und Raumfahrt (NL)

Documents
Document Sharing Hub

Full document details: DIN EN 16603-70-32:2014-12

Nationales Vorwort

Dieses Dokument (EN 16603-70-32:2014) wurde vom Technischen Komitee CEN/CLC/TC 5 „Raumfahrt“ erarbeitet, dessen Sekretariat vom DIN (Deutschland) gehalten wird.

Das zuständige deutsche Normungsgremium ist der Arbeitsausschuss NA 131-10-01 AA „Interoperabilität von Informations-, Kommunikations- und Navigationssystemen“ im DIN-Normenausschuss Luft- und Raumfahrt (NL).

Dieses Dokument (EN 16603-70-32:2014) basiert auf ECSS-E-ST-70-32C.

Dieses Dokument enthält unter Berücksichtigung des DIN-Präsidialbeschlusses 1/2004 nur die englische Originalfassung von EN 16603-70-32:2014.

Dieses Dokument wurde speziell zur Behandlung von Raumfahrtsystemen erarbeitet und hat daher Vorrang vor jeglicher Europäischer Norm, da es denselben Anwendungsbereich hat, jedoch über einen größeren Geltungsbereich (z. B. Luft- und Raumfahrt) verfügt.

Documents
Document Sharing Hub

Full document details: DIN EN 16603-70-32:2014-12

Nationaler Anhang NA
(informativ)

Begriffe, Symbole und Abkürzungen

3 Begriffe, Symbole und Abkürzungen

3.1 Begriffe aus anderen Normen

Für die Anwendung dieses Dokuments gelten die Begriffe nach ECSS-S-ST-00-01.

3.2 Für diese Norm spezifische Begriffe

3.2.1
Aktivität
Überwachungs- und Steuerfunktion des Raumfahrtsystems

3.2.2
Verbundparameter
Datensatz, der jede Abfolge der **Datenaufzeichnung** enthält, Felder der **Datenaufzeichnung** und Unterdatensätze, die zusammen ausgewertet werden

BEISPIEL Eine Aufzeichnung von Unregelmäßigkeiten, die vom Raumsegment erzeugt wurde, welches eine Identifizierung (ID) für die Aufzeichnung von Unregelmäßigkeiten enthält sowie eine Reihe damit verbundener **Parameter**.

3.2.3
Bestätigungsteil
Teil einer **Prozedur** (oder **Schritt**), dessen Zweck die Beurteilung ist, ob die Zielsetzung einer **Prozedur** (oder eines **Schrittes**) erreicht wurde

3.2.4
Fortsetzungsprüfung
Sprachkonstruktion, die verwendet wird, um festzulegen, wie der Ablauf einer **Prozedur** (oder eines **Schrittes**) nach einem einzelnen **Schritt** (oder einer **Aktivität**), weitergeführt wurde

3.2.5
Ereignis
Vorkommen eines Zustandes oder einer Reihe von Zuständen, die während eines Prüfzeitraums oder einer Missionsphase auftreten können

3.2.6
Initialisierung
Anforderung, um die Ausführung eines **Schrittes** oder einer **Aktivität** einzuleiten

3.2.7
Hauptteil
Teil einer **Prozedur** (oder eines **Schrittes**), der darauf ausgerichtet ist, die Zielsetzungen der **Prozedur** (oder des **Schrittes**) zu erreichen

3.2.8
Parameter
unterste Ebene von grundlegender Information, die für die Überwachung des Raumfahrtsystems von Bedeutung ist

AVA Documents
Document Sharing Hub

Full document details: DIN EN 16603-70-32:2014-12

3.2.9
Vorbedingungsteil
Teil einer **Prozedur**, die darauf ausgerichtet ist, sicherzustellen, dass die **Prozedur** nur dann abläuft, falls oder wenn vorgegebene Initialisierungsbedingungen erfüllt wurden

3.2.10
Prozedur
Interaktionsmittel mit dem Raumfahrtsystem, um eine vorgegebene Zielsetzung oder eine Reihe von Zielsetzungen zu erreichen

3.2.11
Datenaufzeichnung
Daten, die für die Beurteilung der Funktionsweise des Raumfahrtsystems verwendet werden

ANMERKUNG Datenaufzeichnungen können aus einem Parameter (Einzeltyp) oder einem Verbundparameter (Verbundtyp) bestehen.

3.2.12
Raumfahrtsystemmodell
Darstellung des Raumfahrtsystems hinsichtlich dessen Zerlegung in **Systemelemente**, die **Aktivitäten**, welche an diesen **Systemelementen** durchgeführt werden können, die **Datenaufzeichnung**, welche den Zustand dieser **Systemelemente** wiedergibt und die **Ereignisse**, die zur Steuerung dieser **Systemelemente**, **Aktivitäten** oder **Datenaufzeichnungen** herbeigeführt und gehandhabt werden können

3.2.13
Anweisung
Element der **prozeduralen Sprache**, das gemeinsam mit weiteren Elementen das Ziel einer **Prozedur** (oder eines **Schrittes**) vorgibt

3.2.14
Schritt
Bestandteil einer **Prozedur**, der ein eindeutig festgelegtes Teilziel erreicht

3.2.15
Systemelement
Darstellung eines Funktionselementes des Raumfahrtsystems innerhalb des **Raumfahrtsystemmodells**

3.2.16
Überwachungsteil
Teil einer **Prozedur** (oder eines **Schrittes**), der Notfallsituationen behandelt, die während der Ausführung einer **Prozedur** (oder eines **Schrittes**) auftreten können

3.2.17
Überwachungsschritt
Bestandteil des **Überwachungsteils**, der dazu vorgesehen ist, das Vorkommen eines besonderen Notfallzustands zu ermitteln und korrigierende Maßnahmen durchzuführen

Documents
Document Sharing Hub
Full document details: DIN EN 16603-70-32:2014-12

3.3 Abkürzungen

Für die Anwendung dieses Dokuments gelten die Abkürzungen nach ECSS-S-ST-00-01 sowie die folgenden Abkürzungen.

Abkürzung	Bedeutung
AIV	Baugruppe, Integration und Verifizierung (*en. assembly, integration and verification*)
EBNF	Erweiterte Backus-Naur-Form (*en. extended Backus-Naur form*)
EGSE	Elektrisches Bodendienstgerät (*en. electrical ground support equipment*)
EMCS	EGSE und Missionssteuersystem (*en. EGSE and mission control system*)
FCP	Flugkontrollverfahren (*en. flight control procedure*)
FOP	Flugbetriebsplan (*en. flight operation plan*)
MMI	Schnittstelle von Mensch und Maschine (*en. man-machine interface*)
PLUTO	Sprache für Anwender von Test- und Bedienprozeduren (*en. procedure language for users in test and operations*)
SCOE	Checkout-Anlage (*en. special check-out equipment*)
SSM	Raumfahrtsystemmodell (*en. space system model*)

Documents
Document Sharing Hub

Full document details: DIN EN 16603-70-32:2014-12

— Leerseite —

Documents
Document Sharing Hub
Full document details: DIN EN 16603-70-32:2014-12

EUROPEAN STANDARD

NORME EUROPÉENNE

EUROPÄISCHE NORM

EN 16603-70-32

September 2014

ICS 49.140

English version

Space engineering - Test and operations procedure language

Ingénierie spatiale - Language de procedure pour les essais et des operations

Raumfahrttechnik - Sprache für Test- und Bedienprozeduren

This European Standard was approved by CEN on 6 March 2014.

CEN and CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration. Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the CEN-CENELEC Management Centre or to any CEN and CENELEC member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CEN and CENELEC member into its own language and notified to the CEN-CENELEC Management Centre has the same status as the official versions.

CEN and CENELEC members are the national standards bodies and national electrotechnical committees of Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, Former Yugoslav Republic of Macedonia, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and United Kingdom.

CEN-CENELEC Management Centre:
Avenue Marnix 17, B-1000 Brussels

© 2014 CEN/CENELEC All rights of exploitation in any form and by any means reserved worldwide for CEN national Members and for CENELEC Members.

Ref. No. EN 16603-70-32:2014 E

Documents
Document Sharing Hub

Full document details: DIN EN 16603-70-32:2014-12

Table of contents

Foreword **5**

Introduction **6**

1 Scope **7**

2 Normative references **8**

3 Terms, definitions and abbreviated terms **9**

3.1 Terms from other standards 9

3.2 Terms specific to the present standard 9

3.3 Abbreviated terms 11

4 Context of the procedure language **12**

4.1 Introduction 12

4.1.1 The space system 12

4.1.2 Satellite testing 13

4.1.3 Mission operations 14

4.2 EGSE and mission control system (EMCS) 14

4.2.1 General 14

4.2.2 Space system model 14

5 Requirements to be satisfied by procedures **18**

5.1 Procedure structure 18

5.2 Language constructs 19

5.3 Language specification 22

Annex A (informative) The PLUTO language **23**

A.1 The structure of a procedure 23

A.1.1 Procedure definition 23

A.1.2 Procedure declaration body 24

A.1.3 Procedure preconditions body 24

A.1.4 Procedure main body 25

A.1.5 Procedure watchdog body 25

A.1.6 Procedure confirmation body 26

Documents
Document Sharing Hub

Full document details: DIN EN 16603-70-32:2014-12

A.1.7 Structure of a step 26
A.2 The behaviour of a procedure 28
A.2.1 Procedure execution flow 28
A.2.2 Step execution flow 31
A.2.3 Activity execution flow 33
A.2.4 Execution in parallel 34
A.2.5 Continuation following an “initiate and confirm” statement 35
A.3 PLUTO language definition 37
A.3.1 Conventions 37
A.3.2 Language case sensitivity 38
A.3.3 Comments 38
A.3.4 Keywords 38
A.3.5 Identifiers 39
A.3.6 Constants 40
A.3.7 Types 43
A.3.8 System interfaces 44
A.3.9 Language constructs 45
A.4 Extended Backus-Naur form (EBNF) representation of PLUTO language constructs 103
A.4.1 Conventions 103
A.4.2 PLUTO language constructs 105
A.5 Index of PLUTO language constructs 117

Annex B (informative) Engineering units 120
B.1 Introduction 120
B.2 Engineering units and symbols 120
B.3 Engineering units railroad diagrams 126
B.4 EBNF representation of the engineering units 129

Annex C (informative) Functions 131
C.1 Introduction 131
C.2 Mathematical functions 131
C.3 Time functions 134
C.4 String functions 135

Bibliography 137

Figures

Figure 4-1: Example of space system elements 13

Documents
Document Sharing Hub
Full document details: DIN EN 16603-70-32:2014-12

Figure 4-2: Example of a space system model 16

Figure A-1 : Example of a procedure and its elements 24
Figure A-2 : Execution states and transitions for a procedure 31
Figure A-3 : Execution states and transitions for a step 33
Figure A-4 : Execution states and transitions for an activity 34
Figure A-5 : Confirmation status and continuation action combinations for main body "initiate and confirm" statements 36
Figure A-6 : Confirmation status and continuation action combinations for watchdog "initiate and confirm" statements 36
Figure A-7 : Example railroad diagram 38

Tables

Table A-1 : Predefined types 43
Table A-2 : Activity and step operation requests 78
Table A-3 : Reporting data, variable and argument operation requests 78
Table A-4 : Predefined operators 97
Table A-5 : Activity and step property requests 101
Table A-6 : Reporting data, variable and argument property requests 102
Table A-7 : Event property requests 102
Table A-8 : EBNF symbols and meanings 104
Table B-1 : Simple engineering units 121
Table B-2 : Acceptable multiple and submultiple of engineering unit 123
Table B-3 : Acceptable multiples of binary engineering units 124
Table B-4 : Standard compound engineering units 124
Table C-1 : Mathematical functions 131
Table C-2 : Time functions 134
Table C-3 : String functions 135

Documents
Document Sharing Hub

Full document details: DIN EN 16603-70-32:2014-12

Foreword

This document (EN 16603-70-32:2014) has been prepared by Technical Committee CEN/CLC/TC 5 "Space", the secretariat of which is held by DIN.

This standard (EN 16603-70-32:2014) originates from ECSS-E-ST-70-32C.

This European Standard shall be given the status of a national standard, either by publication of an identical text or by endorsement, at the latest by March 2015, and conflicting national standards shall be withdrawn at the latest by March 2015.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CEN [and/or CENELEC] shall not be held responsible for identifying any or all such patent rights.

This document has been prepared under a mandate given to CEN by the European Commission and the European Free Trade Association.

This document has been developed to cover specifically space systems and has therefore precedence over any EN covering the same scope but with a wider domain of applicability (e.g. : aerospace).

According to the CEN-CENELEC Internal Regulations, the national standards organizations of the following countries are bound to implement this European Standard: Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, Former Yugoslav Republic of Macedonia, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and the United Kingdom.

Documents
Document Sharing Hub
Full document details: DIN EN 16603-70-32:2014-12

Introduction

The procedure is the principal mechanism employed by the end-user to control the space system during pre-launch functional testing and post-launch in-orbit operations.

This Standard identifies the requirements to be satisfied by any language used for the development of automated test and operation procedures.

It also defines a reference language that fulfils these requirements. This language is called the "procedure language for users in test and operations (PLUTO)".

Documents
Document Sharing Hub

Full document details: DIN EN 16603-70-32:2014-12

1 Scope

This Standard specifies:

- The capabilities of the language used for the definition of procedures for space system testing and operations.
- The PLUTO language.

Clause 4 defines the context in which procedures operate.

Clause 5 contains the requirements for the procedure language.

Annex A specifies the PLUTO language. This includes:

- The "building blocks" that constitute procedures and the role that each of these building blocks plays in achieving the overall objectives of the procedure.
- The dynamic aspects of procedures i.e. the execution logic of each building block and execution relationships between these blocks.
- The syntax and semantics of the language itself.

Annex B specifies the engineering units to be supported by the procedure language.

Annex C specifies the mathematical, time and string functions to be supported by the procedure language.

This standard may be tailored for the specific characteristics and constraints of a space project in conformance with ECSS-S-ST-00.

Documents
Document Sharing Hub
Full document details: DIN EN 16603-70-32:2014-12

2
Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this ECSS Standard. For dated references, subsequent amendments to, or revisions of any of these publications, do not apply. However, parties to agreements based on this ECSS Standard are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references the latest edition of the publication referred to applies.

EN reference	Reference in text	Title
EN 16601-00-01	ECSS-S-ST-00-01	ECSS system – Glossary of terms
	ISO/IEC 14977	Information technology - Syntactic metalanguage – Extended BNF

Documents
Document Sharing Hub

Full document details: DIN EN 16603-70-32:2014-12

3 Terms, definitions and abbreviated terms

3.1 Terms from other standards

For the purpose of this Standard, the terms and definitions from ECSS-S-ST-00-01 apply.

3.2 Terms specific to the present standard

3.2.1 activity

space system monitoring and control function

3.2.2 compound parameter

record comprised of any sequence of **reporting data**, arrays of **reporting data** and sub-records that are interpreted together

EXAMPLE An anomaly report generated by the space segment comprising an anomaly report ID and a set of associated **parameters**.

3.2.3 confirmation body

part of a **procedure** (or **step**) whose purpose is to assess whether or not the objective of the **procedure** (or **step**) has been achieved

3.2.4 continuation test

language construct used to define how the execution of a **procedure** (or **step**) proceeds after a constituent **step** (or **activity**) has been executed

3.2.5 event

occurrence of a condition or set of conditions that can arise during the course of a test session or a mission phase

3.2.6 initiation

act of requesting the execution of a **step** or an **activity**

3.2.7 main body

part of a **procedure** (or **step**) dedicated to achieving the objectives of the **procedure** (or **step**)

Documents
Document Sharing Hub
Full document details: DIN EN 16603-70-32:2014-12

3.2.8 parameter

lowest level of elementary information that has a meaning for monitoring the space system

3.2.9 preconditions body

part of a **procedure** dedicated to ensuring that the **procedure** only executes if or when pre-defined initial conditions are satisfied

3.2.10 procedure

means for interacting with the space system in order to achieve a given objective or sequence of objectives

3.2.11 reporting data

data used for assessing the functioning of the space system

NOTE Reporting data can consist of a parameter (a simple type) or a compound parameter (a complex type).

3.2.12 space system model

representation of the space system in terms of its decomposition into **system elements**, the **activities** that can be performed on these **system elements**, the **reporting data** that reflects the state of these **system elements** and the **events** that can be raised and handled for the control of these **system elements**, **activities** or **reporting data**

3.2.13 statement

element of the **procedure** language which, together with other elements, implements the goal of a **procedure** (or **step**)

3.2.14 step

component of a **procedure** that achieves a well-defined sub-goal

3.2.15 system element

representation within the **space system model** of a functional element of the space system

3.2.16 watchdog body

part of a **procedure** (or **step**) which manages contingency situations that can arise during the execution of the **procedure** (or **step**)

3.2.17 watchdog step

component of the **watchdog body** dedicated to detecting the occurrence of a particular contingency condition and executing corrective actions

Documents
Document Sharing Hub
Full document details: DIN EN 16603-70-32:2014-12

3.3 Abbreviated terms

For the purpose of this Standard, the abbreviated terms from ECSS-S-ST-00-01 and the following apply:

Abbreviation	Meaning
AIV	assembly, integration and verification
EBNF	extended Backus-Naur form
EGSE	electrical ground support equipment
EMCS	EGSE and mission control system
FCP	flight control procedure
FOP	flight operations plan
MMI	man-machine interface
PLUTO	procedure language for users in test and operations
SCOE	special check-out equipment
SSM	space system model

Documents
Document Sharing Hub
Full document details: DIN EN 16603-70-32:2014-12

4 Context of the procedure language

4.1 Introduction

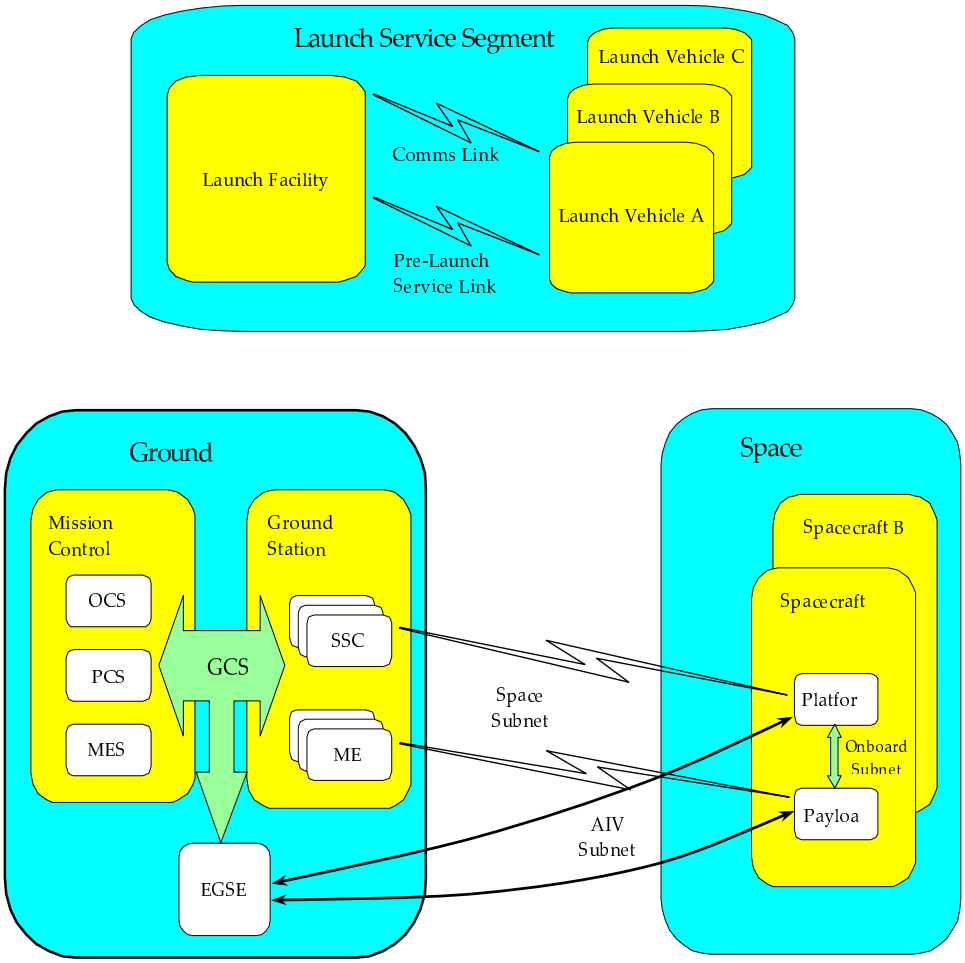
4.1.1 The space system

ECSS-S-ST-00 defines the overall space system as comprising a space segment, a ground segment and a launch service segment.

An example of the elements of a space system is shown in Figure 4-1. The space system elements shown in this figure are operational at different times:

- the electrical ground support equipment (EGSE) during the development phase;
- the launch service segment during the pre-launch and launch phases;
- the mission control and ground station systems during the mission operations phase.

Documents
Document Sharing Hub
Full document details: DIN EN 16603-70-32:2014-12



Key: OCS: Operation control system
PCS: Payload control system
MES: Mission exploitation system
AIV: Assembly, integration and verification
SSC: Space segment control station
ME: Mission exploitation station
GCS: Ground communications subnet

Figure 4-1: Example of space system elements

4.1.2 Satellite testing

ECSS-E-ST-10, ECSS-E-ST-10-02 and ECSS-E-ST-10-03 define the requirements for space system engineering, verification and testing.

This Standard does not prescribe the levels of integration and test at which procedures are used. This is considered to be a decision taken when the verification approach for a specific mission is defined. However, automated procedures are generally employed from the subsystem level upwards.

The re-use of test procedures at different levels of integration implies standardization of the functionality of the EGSE. Furthermore, the re-use of these procedures in the mission operations domain implies the harmonisation of the requirements for EGSE and mission control systems.

Documents
Document Sharing Hub
Full document details: DIN EN 16603-70-32:2014-12

4.1.3 Mission operations

ECSS-E-ST-70 identifies procedures as the primary mechanism for conducting mission operations and defines two types of flight control procedures (FCP):

- Nominal procedures

  These define the set of in-orbit operations of the space system to be used under nominal conditions. They constitute the building blocks from which the mission timelines and schedules of the flight operations plan (FOP) are constructed.

- Contingency procedures

  These define the recovery actions used to reconfigure the space system if pre-identified anomalies or failures occur.

Although FCPs have traditionally been executed under manual control, pressure to reduce manpower during routine mission operations implies more automation of routine tasks such as the execution of procedures.

4.2 EGSE and mission control system (EMCS)

4.2.1 General

In this Standard, the elements of the ground segment responsible for the monitoring and control of the overall space system, namely the EGSE and the mission control system are referred to jointly as the EMCS.

The procedure language and the procedure development and execution environments are an integral part of any EMCS. As such, they have direct access to other monitoring and control functions implemented within the overall EMCS.

4.2.2 Space system model

ECSS-E-ST-70-31 introduces the concept of a space system model (SSM) as a means for capturing mission knowledge used during AIV and operations. This knowledge is used by the different EMCS applications in order to interact with the space system and to process the dynamic data that is exchanged with it (i.e. space segment telemetry and telecommands, ranging data, ground segment commands and measurements).

The SSM consists of different types of object and the relationships between these objects. The objects of relevance for the procedure language are system elements, reporting data, activities and events.

System elements correspond to:

- the elements of the space segment resulting from the functional decomposition defined in ECSS-S-ST-00;
- the elements of the ground segment resulting from the functional decomposition defined in ECSS-E-ST-70.

Documents
Document Sharing Hub
Full document details: DIN EN 16603-70-32:2014-12

Reporting data and activities are associated with system elements:

- Reporting data comprises parameters and compound parameters. A parameter is the lowest level of elementary information that has a meaning for monitoring the space system. A compound parameter is a record comprised of any sequence of parameters, arrays of parameters and sub-records (see also ECSS-E-ST-70-41). For example, a complete telemetry packet, or part thereof, may be represented as a compound parameter. The parameters within a compound parameter are normally interpreted together (e.g. when interpreting the contents of an anomaly report). Reporting data can have different representations depending on its life cycle within the space system (e.g. an on-board measurement has an encoded value in telemetry and a raw or engineering value when presented on a ground segment display).
- An activity is a space system monitoring and control function. The term activity is introduced to refer generically to procedures, telecommands (either to the space segment or to the ground segment) and any function provided by the underlying EMCS (e.g. a printer request, sending an e-mail, transferring a file using ftp). A given mission can implement additional mission-specific activity types (e.g. conforming to a non-standard protocol).

Events are associated with system elements, reporting data and activities. An event is an occurrence of a condition or set of conditions that can arise during the course of a test session or a mission phase. It is used to trigger a monitoring and control function implemented within the space system.

An example of a space system model is shown in Figure 4-1.

Documents
Document Sharing Hub
Full document details: DIN EN 16603-70-32:2014-12

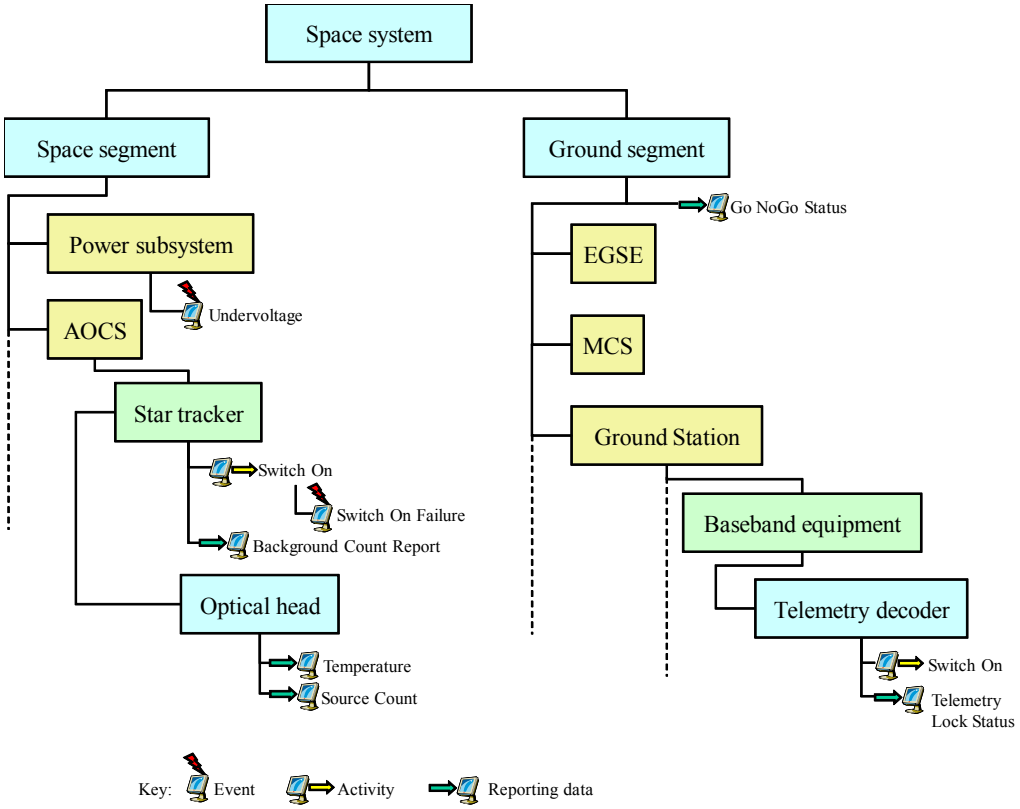


Figure 4-2: Example of a space system model

During the course of spacecraft testing and mission operations, the space system is configured in different ways, for example:

- to test elements of the space segment in a stand-alone manner;
- to test the space segment during integration when some elements are missing;
- to validate the ground segment with a simulated space segment.

The concepts of system elements, activities, reporting data and events provide a complete definition of the SSM independent of any specific space system configuration. However, for a given space system configuration, only a subset of the overall SSM is used.

In order to understand the scope of the procedure language, it is important to understand what is provided by the SSM. The SSM (as specified in ECSS-E-ST-70-31) provides the definition of space system configurations, system elements, activities, reporting data and events.

The procedure language provides the means to:

- refer to system elements, activities, reporting data and events but not to define them;
- define the procedural script.

For activities, the common set of data definitions in the SSM includes:

- name;
- description;

Documents
Document Sharing Hub
Full document details: DIN EN 16603-70-32:2014-12

- type (procedure, telecommand or operating system call);
- associated system element;
- version number and configuration history;
- validation status (i.e. information on the testing status of the activity);
- validity period, i.e. the earliest and the latest date on which the activity can be executed;
- expected (i.e. mean) duration;
- maximum and minimum durations;
- list of arguments, including for each argument:
  - name,
  - description,
  - engineering units,
  - data type, and
  - default value;
- allowed combinations of argument values;
- other attributes used for mission planning purposes such as the profile of resources that the activity utilizes (e.g. onboard power and downlink bandwidth).

In addition, the different types of activity have type-specific definitions; for procedures, this includes:

- default execution mode (manual or automatic);
- activation mode (i.e. whether the procedure is permanently active or is explicitly initiated);
- the procedural script.

Documents
Document Sharing Hub
Full document details: DIN EN 16603-70-32:2014-12

5 Requirements to be satisfied by procedures

5.1 Procedure structure

a. The capability shall be provided to construct a high-level, goal-oriented activity (namely a procedure) using elementary activities consisting of telecommands and operating system calls.

b. A procedure may include calls to execute other procedures.

c. The capability shall be provided to define self-contained sub-goals for a procedure.

d. A given sub-goal may be achieved by a single activity call or a sequence of activity calls.

e. An activity may have associated arguments whose values are passed to the activity at the time of initiation.

f. Where the achievement of a sub-goal involves complex logic (such as waiting for a period of time, waiting for a condition to become true or conditional branching), a step construct shall be provided to encapsulate the logic.

g. The capability shall be provided to execute procedure sub-goals in series or in parallel.

NOTE Parallel execution is used, for example, when two or more steps can be performed completely independently.

h. Preconditions may be specified for a procedure.

NOTE Preconditions are conditions to be fulfilled before the procedure can start.

i. Confirmation conditions may be specified for a procedure.

NOTE Confirmation conditions are those conditions that determine whether the goal of a procedure is met.

j. Preconditions and confirmation conditions may also be defined for steps.

k. In addition to the "nominal flow path" of a procedure (i.e. the flow designed to achieve its primary goal), contingency actions may also be defined.

l. A procedure contingency action may be executed upon:

Documents
Document Sharing Hub
Full document details: DIN EN 16603-70-32:2014-12

1. detection of space system anomalies;
2. detection of a failure in the execution of the nominal path;
3. other specified events or conditions.

m. In addition to the "nominal flow path" of a step, contingency actions may also be defined.

n. A step contingency action may be executed upon:

1. detection of a failure in the execution of the nominal path;
2. other specified events or conditions.

o. A contingency action should be taken at the level at which a failure occurs, i.e. if a failure occurs in the execution of a step, a contingency action should be taken at the level of the step.

p. A contingency action should rectify the detected anomaly or report it to the user and return control to the nominal flow path of the procedure (or step).

q. If an anomaly cannot be rectified, one of the following actions shall be performed:

1. generate an event to flag the problem at a higher level for resolution;
2. use other means to achieve the goal of the procedure (or sub-goal of a step) and then terminate the procedure (or step);
3. abort the procedure (or step).

r. If a contingency action is invoked, the body of the procedure (or step) shall be suspended until the contingency action is completed.

5.2 Language constructs

a. The capability shall be provided to request the execution of any space system activity.

b. The capability shall be provided to request the execution of an activity and proceed immediately with the execution of the next statement of the procedure.

c. The capability shall be provided to request the execution of an activity and wait for the confirmation of execution before continuing.

d. When a procedure waits for the confirmation of execution of an activity, the result of the execution of the activity shall be tested in order to determine the subsequent course of action.

e. The capability shall be provided to acquire the following data:

1. the execution status or confirmation status of an initiated activity;
2. the initiation time, start time, termination time or completion time of the last execution of an activity;
3. the value of reporting data;

Documents
Document Sharing Hub
Full document details: DIN EN 16603-70-32:2014-12

4. the validity status of a parameter;
5. the overall monitoring status or detailed monitoring status (limit-check, delta-check, expected status check or status consistency check) of a parameter;
6. the sampling time of reporting data;
7. the time of the last occurrence of an event.

f. When a given space system function is unavailable (e.g. during an early test phase), the capability shall be provided to replace the missing function. This includes setting:

1. the confirmation status of an activity;
2. the value of a parameter;
3. the validity status of a parameter;
4. the overall monitoring status or detailed monitoring status (limit-check, delta-check, expected status check or status consistency check) of a parameter;
5. the sampling time of reporting data.

g. To avoid the problem that can arise with asynchronous systems where a new instance of reporting data arrives between two successive procedure statements that use different properties of this object, the capability shall be provided to create a local copy of reporting data.

NOTE This ensures that the procedure uses properties of reporting data that have been sampled at the same instant in time.

h. To ensure that any service provided by the EMCS is accessible within a procedure, the following capabilities shall be provided:

1. to request any operation defined for any space system object;
2. to acquire the value of any property of any space system object.

i. The capability shall be provided to perform the following execution flow controls within a procedure:

1. simple conditional branching (i.e. if … then … else …);
2. multiple conditional branching where the path taken is dependent on the value of a specified parameter (or local variable, see j below);
3. repeated execution of a statement (or a group of statements) with the possibility of repeating the execution a specified number of times, repeating the execution indefinitely whilst a given condition holds true or repeating the execution until a given condition becomes true;
4. wait until a given absolute time;
5. wait for a given interval of time to elapse;
6. wait until a given condition becomes true;
7. wait until a given event occurs.

Documents
Document Sharing Hub
Full document details: DIN EN 16603-70-32:2014-12

j. Local variables used within a procedure shall be declared and their type explicitly stated.

k. The capability shall be provided to assign a value to a local variable within a procedure.

l. The capability shall be provided to raise a local event within a procedure (e.g. to trigger a contingency action).

m. Local events shall be defined as part of the procedure (or step) declarations.

n. Mathematical, time and string functions as specified in Annex C should be supported.

o. The capability shall be provided to construct expressions that operate on constants, space system parameters acquired from the EMCS, activity arguments (whose values are supplied at runtime) and local variables.

p. Engineering units shall be supported.

q. The capability to assign engineering units to constants shall be supported.

r. The capability to mix compatible units freely within an expression shall be supported

s. The automatic conversion between different, but compatible, units shall be supported.

t. Comments may be included within a procedure.

u. The capability shall be provided to generate a message for acknowledgement by the user.

v. The capability shall be provided to generate a message for entry in the procedure execution log.

w. The capability shall be provided to acquire inputs from the user.

x. The conditions specified for a procedure (or step) precondition or confirmation may be any combination of the following:

1. wait until a given absolute time;
2. wait for a given interval of time to elapse;
3. wait until a given condition becomes true;
4. wait until a given event occurs;
5. test whether a given condition is true;
6. request the user to specify the outcome.

Documents
Document Sharing Hub
Full document details: DIN EN 16603-70-32:2014-12

5.3 Language specification

a. Any language that complies with this Standard shall be formally specified using the ISO extended Backus-Naur form (EBNF), in conformance with ISO/IEC 14977.

b. The PLUTO procedure language, engineering units and functions, as specified in Annex A should be used for the automation of procedures in the development and exploitation phases of space systems.

NOTE Use of the PLUTO language ensures conformance with all requirements of this Standard. It also facilitates the transfer of procedure knowledge, acquired in the development phase during functional testing, to the mission operations phase (for a given mission) and encourages the use of a common procedure language across missions.

Documents
Document Sharing Hub
Full document details: DIN EN 16603-70-32:2014-12

Annex A (informative) The PLUTO language

A.1 The structure of a procedure

A.1.1 Procedure definition

A procedure comprises the following elements:

a. An optional declaration body, which declares the local events that can be raised within the procedure.

b. An optional preconditions body, which ensures that the procedure is only executed if (or when) pre-defined initial conditions are satisfied.

c. A mandatory main body, which fulfils the goal of the procedure. The main body can be composed of self-contained sub-goals fulfilled by activities or steps.

d. An optional watchdog body, which manages contingency situations that can arise during the execution of the procedure. The watchdog body is composed of one or more special steps, called watchdog steps, which are all initiated in parallel.

e. An optional confirmation body, which assesses whether the objectives of the procedure have been achieved or not.

An example of a procedure and its elements is shown in Figure A-1.

Documents
Document Sharing Hub
Full document details: DIN EN 16603-70-32:2014-12

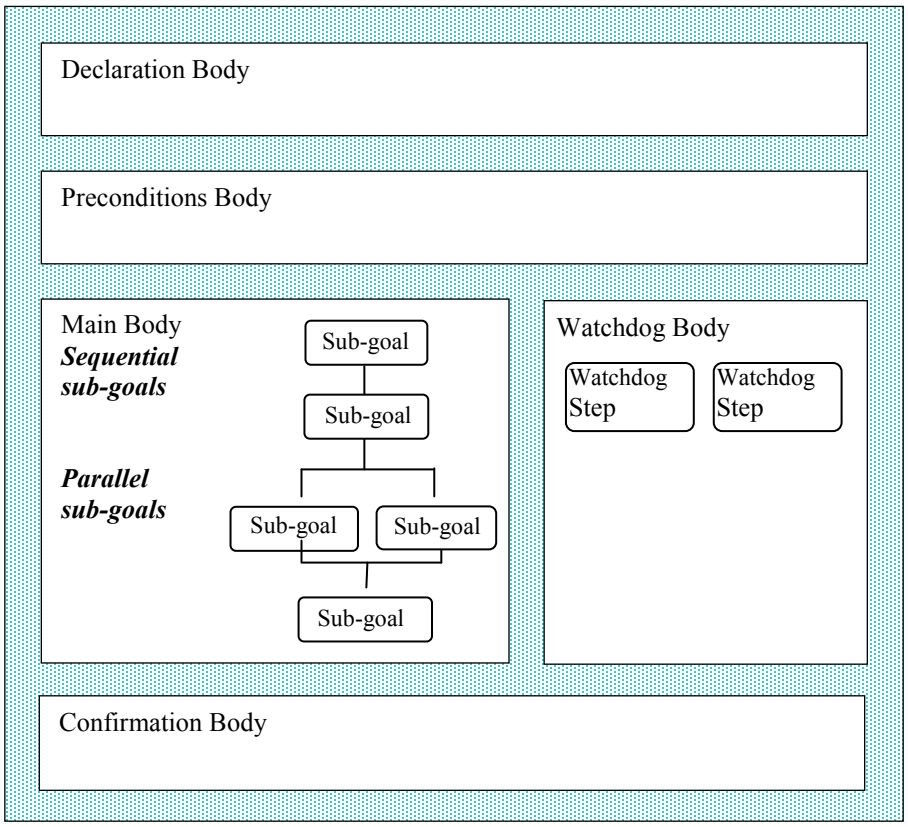


Figure A-1: Example of a procedure and its elements

A.1.2 Procedure declaration body

Local events that can be raised within the procedure, but which are accessible outside of the step in which they are raised (e.g. events that trigger a procedure-level watchdog, see clause A.1.5), are declared at procedure level.

Arguments may be passed to a procedure at initiation time by the entity calling the procedure. These arguments are defined externally (see clause 4.2.2) and not as part of the procedure declarations.

Arguments are similar to parameters in their definition and use.

Argument values cannot be changed within a procedure.

A.1.3 Procedure preconditions body

The preconditions body contains the conditions that define whether a procedure can start. They may be any combination of the following:

- wait until a given absolute time;
- wait for a given time interval to elapse;
- wait until a given condition becomes true;
- if the result of a logical expression (Boolean condition) is true;
- request a decision from the user.

Documents
Document Sharing Hub
Full document details: DIN EN 16603-70-32:2014-12

A.1.4 Procedure main body

The main body of a procedure consists of a sequence of statements of the following type:

- set procedure context statement;
- initiate in parallel;
- initiate and confirm step;
- initiate and confirm activity;
- initiate activity;
- inform user statement;
- log statement.

The "initiate in parallel" statement enables a combination of steps and activities to be executed in parallel with the subsequent behaviour being one of the following:

a. the set of parallel steps or activities ends when any one of the steps or activities has ended;

b. the set ends when all of the steps or activities have ended.

A.1.5 Procedure watchdog body

The watchdog body is composed of watchdog steps. Each watchdog step monitors for the occurrence of a particular contingency condition and executes corrective actions if the condition occurs.

The purpose of the procedure watchdog body is the following:

- Detection of, and reaction to, system-level events, for example space system anomalies. The procedure watchdog body need not rectify all anomalies internally; it can suspend or abort the procedure whilst another procedure handles the anomaly.
- Ensuring that procedure-level invariant conditions are sustained for the duration of the procedure.
- Management of errors (e.g. failures) in procedure execution that cannot be handled by the watchdog of the relevant procedure main body step (see clause A.1.7.5).

NOTE Error handling following execution of a procedure is managed by the calling entity.

Contingency situations monitored for by concurrently active watchdog steps are independent of each other. The reason for this is that when a watchdog step is triggered, it suspends the main body of the corresponding procedure whilst it takes the appropriate recovery action, but any other watchdog steps that are active continue their monitoring. Consequently, if there is any potential interaction between two or more contingency actions, the design of the procedure ensures that they are handled within the same watchdog step.

Documents
Document Sharing Hub
Full document details: DIN EN 16603-70-32:2014-12

A.1.6 Procedure confirmation body

Within the confirmation body, a procedure can be confirmed by any combination of the following conditions:

- wait until a given absolute time;
- wait for a given time interval to elapse;
- wait until a given condition becomes true;
- if the result of a logical expression (Boolean condition) is true;
- request a decision from the user.

A.1.7 Structure of a step

A.1.7.1. Step definition

The structure of a step is identical to the structure of a procedure: a step is composed of an optional declaration body, a preconditions body (optional for a main body step, mandatory for a watchdog step), a mandatory main body, an optional watchdog body and an optional confirmation body.

The declarations at step-level (see clause A.1.7.2) are different from those at procedure-level and the set of statements that can be used in a step main body (see clause A.1.7.4) is different from that which can be used in a procedure main body.

A step has a name, which is unique within a procedure. This name may be used to refer to the step.

Whilst steps within a procedure main body may have a watchdog body, watchdog steps do not have a watchdog body.

The preconditions body is mandatory for a watchdog step (since the preconditions body defines the contingency condition for which the watchdog step is monitoring).

A.1.7.2. Step declaration body

The following are declared at step level:

- Local variables: the objects defined for internal use within a step.

  Local variables are similar to parameters in their definition and use. They have an engineering value and a validity status, which is "**invalid**" until the variable is first assigned a value.

  Local variables can be used by steps contained within a step and also by watchdog steps within the step watchdog body.

- Local events: the events that can be raised within the step and which are only accessible from within that step.

Documents
Document Sharing Hub
Full document details: DIN EN 16603-70-32:2014-12

A.1.7.3. Step preconditions body

Within the preconditions body of a main body step, the conditions that define whether a step can start may be any combination of the following:

- wait until a given absolute time;
- wait for a given time interval to elapse;
- wait until a given condition becomes true;
- if the result of a logical expression (Boolean condition) is true;
- request a decision from the user.

However, the preconditions body of a watchdog step consists of a single "wait" condition.

A.1.7.4. Step main body

The main body of a step consists of a sequence of statements of the following type:

- set step context statement;
- assignment statement;
- flow control statements:
  - if statement;
  - case statement;
  - loop statements:
    - while statement;
    - for statement;
    - repeat statement;
- wait statement;
- object operation request statement;
- save context statement;
- initiate in parallel (steps or activities or both);
- initiate and confirm step;
- initiate and confirm activity;
- initiate activity;
- inform user statement;
- log statement.

Documents
Document Sharing Hub
Full document details: DIN EN 16603-70-32:2014-12

A.1.7.5. Step watchdog body

A step watchdog body manages the following:

- detection of, and reaction to, any failures that occur within the main body of the corresponding step;
- ensuring that step-level invariant conditions are sustained for the duration of the step.

NOTE Error handling following execution of a step is managed by the calling entity.

A.1.7.6. Step confirmation body

Within the confirmation body, a step can be confirmed by any combination of the following conditions:

- wait until a given absolute time;
- wait for a given time interval to elapse;
- wait until a given condition becomes true;
- if the result of a logical expression (Boolean condition) is true;
- request a decision from the user.

A.2 The behaviour of a procedure

A.2.1 Procedure execution flow

Procedures can be initiated by different mechanisms within the EMCS, e.g. by a user via an MMI or automatically as a call from another procedure.

During the execution of a procedure, its "execution status" changes to reflect the path that it follows ("**preconditions**", "**executing**", "**confirmation**" or "**completed**").

On completion, the "confirmation status" of a procedure changes to reflect its success or failure ("**confirmed**", "**not confirmed**" or "**aborted**"). Prior to completion, the confirmation status of the procedure is "**not available**".

The execution and confirmation statuses can be used by the calling entity to monitor and control the execution of the procedure.

a. When the procedure is initiated, the procedure preconditions body is executed (execution status: "**preconditions**", confirmation status: "**not available**"). The outcome is one of the following:

   — the preconditions are satisfied, resulting in the initiation of the watchdog body (i.e. the initiation of all watchdog step preconditions bodies) and then the initiation of the main body;

   — the preconditions are not satisfied, resulting in the aborting of the procedure (execution status: "**completed**", confirmation status: "**aborted**").

   If the procedure has no user-defined preconditions, the watchdog body and main body are started immediately.

Documents
Document Sharing Hub
Full document details: DIN EN 16603-70-32:2014-12

b. The procedure main body consists of a sequence of statements, which are of the type “initiate” (initiate activity, initiate and confirm step, initiate and confirm activity or initiate steps or activities in parallel), or are informative in nature (inform user or log), or define the context in which statements are executed.

When the main body is started, the first statement in the sequence is executed. Subsequent statements are executed according to the sequence logic (execution status: "executing", confirmation status: "not available").

The initiate and confirm statements (step or activity) include a continuation test, which determines how the main body continues based on the final value of the confirmation status. The outcome is one of the following:

— continue the execution of the main body;

— restart the current step or activity, i.e. restart from the preconditions body (if defined);

— raise a local event that is caught by a watchdog step;

— abort the procedure (execution status: "**completed**", confirmation status: "**aborted**").

c. The watchdog body comprises a number of (watchdog) initiate and confirm step statements. When the watchdog body is started, all of its step statements are initiated in parallel. Each watchdog step is designed to detect a particular contingency situation.

— If none of these contingency situations arises during the procedure execution, the watchdog body is automatically terminated when the procedure main body ends execution;

— If a contingency arises, the main body of the procedure is suspended whilst the contingency is handled by the relevant watchdog step (execution status: "**executing**", confirmation status: "**not available**").

Suspension of the main body means that the currently executing atomic statement is completed, but the subsequent action (if any) indicated as a result of that statement is not taken.

NOTE 1 An atomic statement is the lowest level of statement within the procedure language. All statements except “initiate and confirm step”, “initiate activity”, “initiate and confirm activity” and “initiate in parallel” are atomic in nature.

NOTE 2 In the case that the main body is executing an “initiate in parallel” statement, all executing statements are suspended when all currently executing atomic statements are completed.

The subsequent flow depends on the success or otherwise of the recovery and the “continuation action” arising within the watchdog step. The following cases can arise:

- The contingency is rectified and control is returned to the procedure main body (continuation action: "**resume**",

Documents
Document Sharing Hub
Full document details: DIN EN 16603-70-32:2014-12

execution status: "**executing**", confirmation status: "**not available**"). The execution of the main body is resumed at the end of the previously executing atomic statement. The watchdog step that handled the contingency is re-started.

- The contingency is handled by the watchdog by achieving the procedure goal through an alternative path. The procedure main body and the procedure watchdog body are then terminated and the procedure confirmation body is started (continuation action: "**terminate**", execution status: "**confirmation**", confirmation status: "**not available**").
- The contingency cannot be handled by this watchdog step, in which case either a local event is raised, which triggers another watchdog step (continuation action: "**raise event**"), or the procedure is aborted (continuation action: "**abort**", execution status: "**completed**", confirmation status: "**aborted**").

— If a second watchdog step is triggered whilst the first is still executing, the two watchdog steps are executed in parallel. Control is only returned to the procedure main body when both watchdog steps have completed execution. The watchdog steps are restarted as soon as they complete execution. If either watchdog step yields an "**abort**" continuation action, the procedure is aborted immediately.

d. When the main body has completed execution (i.e. when all initiated steps and activities have completed execution), the watchdog body is terminated and the confirmation body is executed (execution status: "**confirmation**", confirmation status: "**not available**").

NOTE This execution includes any parallel steps and activities that were initiated as part of an "initiate in parallel" statement with an "**until one completes**" termination condition.

e. When the confirmation body has completed execution, the procedure execution is completed. The outcome is one of the following:

— the confirmation conditions are fulfilled (execution status: "**completed**", confirmation status: "**confirmed**");

— the confirmation conditions are not fulfilled (execution status: "**completed**", confirmation status: "**not confirmed**").

If the procedure has no user-defined confirmation conditions, an implicit condition is used instead that depends on whether or not the confirmation body has been initiated by a watchdog step terminate action:

— If the confirmation body has been initiated by a "terminate" action of a watchdog step, the procedure confirmation status is set to the confirmation status of the watchdog step.

— Otherwise, the confirmation status is set to:

Documents
Document Sharing Hub
Full document details: DIN EN 16603-70-32:2014-12

- "**confirmed**" if all the activities and steps that have been initiated in the main body have been confirmed.
- "**not confirmed**" in all other cases.

Figure A-2 shows the different execution states through which a procedure can pass when it is executed and the allowed transitions between these states.

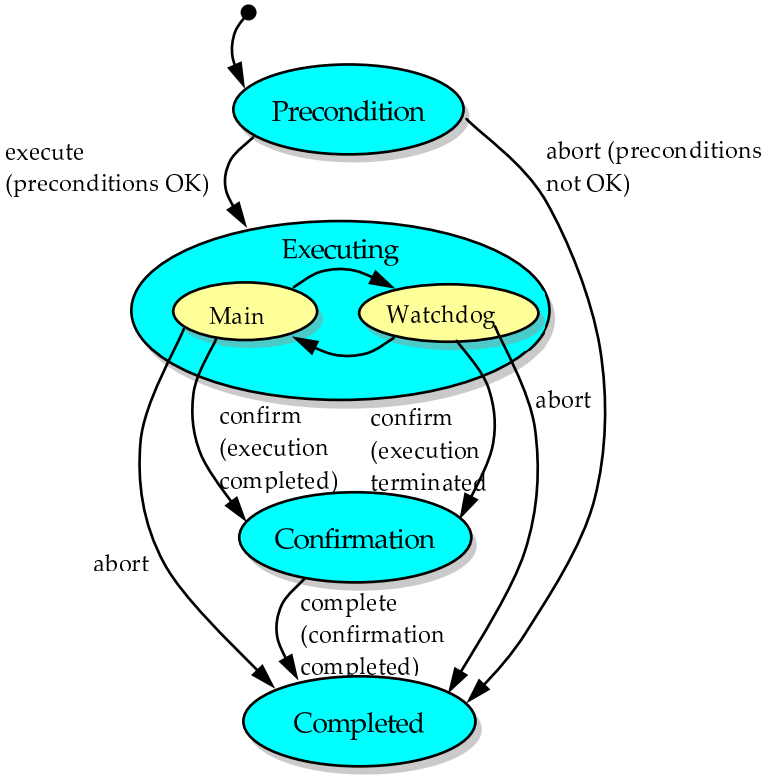


Figure A-2: Execution states and transitions for a procedure

A.2.2 Step execution flow

Steps are initiated by the execution of one of the following statements:

- within a procedure (or step) main body: “initiate and confirm step” or “initiate in parallel”;
- within a procedure (or step) watchdog body: “initiate and confirm step”.

The initiate statement implies both the execution of the step bodies and the “continuation action” derived from the execution result which determines the subsequent course of action.

a. The execution of the step preconditions body and main body proceeds in the same manner as described in clause A.2.1 for the procedure preconditions and main bodies.

b. The step watchdog body comprises a number of watchdog steps, each designed to detect a particular contingency:

Documents
Document Sharing Hub
Full document details: DIN EN 16603-70-32:2014-12

— If none of these contingency situations arises during the step execution, the step watchdog body is automatically terminated when the step main body ends execution.

— If a contingency arises, the main body of the step is suspended whilst the contingency is handled by the relevant watchdog step (execution status: "**executing**", confirmation status: "**not available**").

The subsequent flow depends on the success or otherwise of the recovery and the continuation action arising within the watchdog step. The following cases can arise:

- The contingency is rectified and control is returned to the step main body (continuation action: "**resume**", execution status: "**executing**", confirmation status: "**not available**"). The watchdog step that handled the contingency is re-started.
- The contingency is handled by the watchdog by achieving the step goal through an alternative path. The step main body and the step watchdog body are then terminated and the step confirmation body is started (continuation action: "**terminate**", execution status: "**confirmation**", confirmation status: "**not available**").
- The contingency cannot be handled by this watchdog step, in which case either a local event is raised, which triggers another watchdog step (continuation action: "**raise event**") or the step is aborted (continuation action: "**abort**", execution status: "**completed**", confirmation status: "**aborted**").

— If two watchdog steps are triggered at the same time, their subsequent behaviour is the same as described for a procedure.

c. When the step main body has completed execution, the step watchdog body is terminated and the execution of the step confirmation body is initiated (execution status: "**confirmation**", confirmation status: "**not available**").

d. When the confirmation body has completed execution, the step execution is completed (confirmation status: "**confirmed**" or "**not confirmed**").

e. For a step, the decision on how to proceed following execution is defined within the continuation test, depending on the value of the step confirmation status ("confirmed", "**not confirmed**" or "**aborted**"). A continuation action is defined for each confirmation status, either explicitly within the initiate step statement, or by default. This is unlike for a procedure, where the decision on how to proceed following execution remains with the calling entity.

If the outcome of the continuation test is "restart", the step restarts (execution status: "preconditions"). Otherwise, the step completes execution (execution status: "completed").

If there are no user-defined preconditions or confirmation conditions for a given step, default conditions are used. These default conditions are the same as defined in clause A.2.1 for a procedure.

Documents
Document Sharing Hub
Full document details: DIN EN 16603-70-32:2014-12

Figure A-3 shows the different execution states through which a step can pass when it is executed and the allowed transitions between these states.

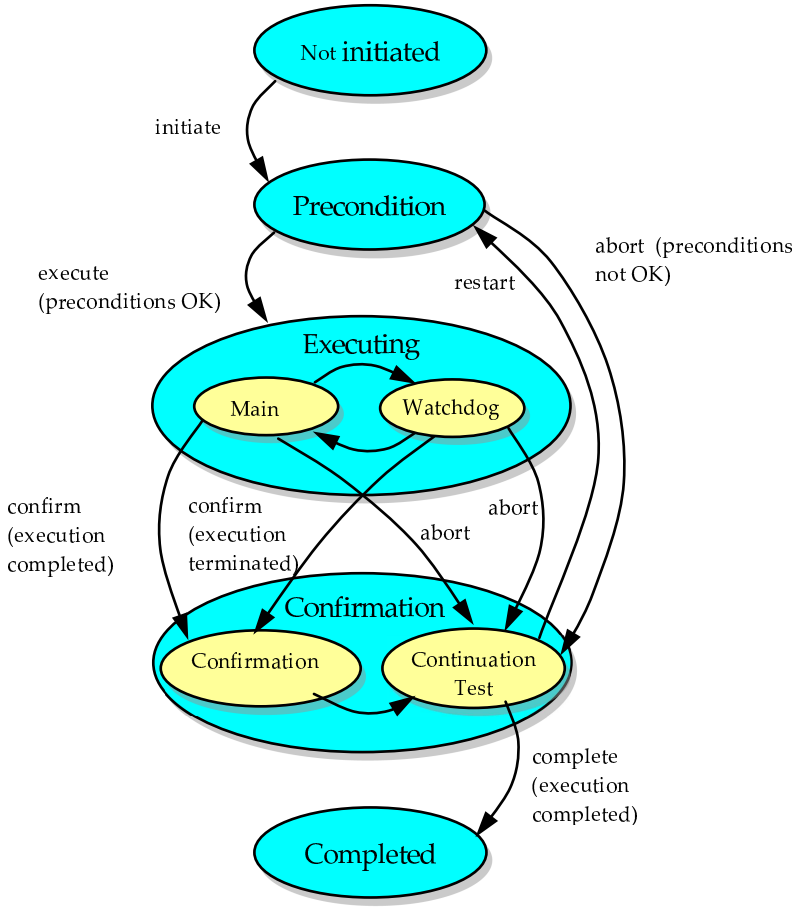


Figure A-3: Execution states and transitions for a step

A.2.3 Activity execution flow

Activities are initiated directly within a procedure or within a step by the execution of either an "initiate activity" or an "initiate and confirm activity" statement. The control of the execution of an activity is managed by the EMCS.

Figure A-4 shows the different executions states through which an activity can pass when it is executed and the allowed transitions between these states. Depending on how the activity is implemented, e.g. as a telecommand or a procedure, different notifications are reported by the EMCS for the monitoring of the progress of execution by the initiating entity. These notifications correspond to the state transitions, but also to sub-state transitions which are specific to different activity types (e.g. for telecommand, routing stages such as "released from EMCS", "reception by ground station" and "reception on-board").

a. In the case of an "initiate activity" statement, the calling entity (procedure or step) proceeds with the execution of the next statement without waiting for notification of completion of the activity.

Documents
Document Sharing Hub
Full document details: DIN EN 16603-70-32:2014-12

b. In the case of an “initiate and confirm activity”, the decision on how to proceed following execution is defined within the continuation test, depending on the value of the confirmation status ("**confirmed**", "**not confirmed**" or "**aborted**"). As for a step, the continuation action is defined for each confirmation status, either explicitly within the initiate statement, or by default.

If the outcome of the continuation test is "**restart**", the activity restarts (execution status: "**preconditions**"). Otherwise, the activity completes execution (execution status: "**completed**").

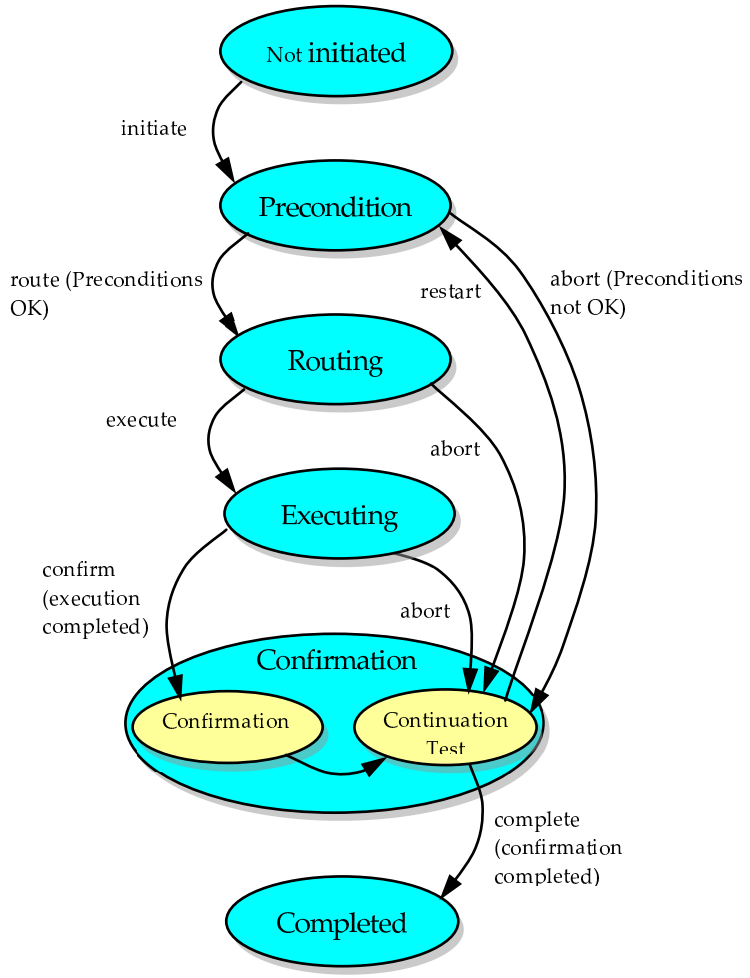


Figure A-4: Execution states and transitions for an activity

A.2.4 Execution in parallel

When an “initiate in parallel” statement is executed, all constituent steps and activities are initiated in parallel.

a. Where the termination condition of the “initiate in parallel” statement is defined as "until one completes", the first step or activity to complete initiates the execution of the next statement in the main body of the calling entity (procedure or step). All other parallel steps and activities

Documents
Document Sharing Hub
Full document details: DIN EN 16603-70-32:2014-12

continue to execute after the first one has completed. When these other parallel steps and activities complete, the continuation actions indicated in their respective continuation tests are taken, except if the specified action is "continue" when no further action is taken.

b. Where the termination condition of the “initiate in parallel” statement is defined as "until all complete", the continuation actions evaluated within the individual steps and activities (or input by the user, if "ask user" is the continuation action) can be different.

— When the action returned by a given step or activity is "**abort**", the procedure (or step) is immediately aborted (including aborting any other parallel steps and activities that are still executing).

— When the action returned by a given step or activity is "**restart**", the action is taken immediately and the initiated step or activity is restarted. Its execution status is reset to "**preconditions**" and its confirmation status is reset to "**not available**". The “initiate in parallel” statement cannot complete until the restarted step or activity has completed.

— When the action returned by a given step or activity is "**raise event**", the event is raised immediately, the event is then caught by a watchdog step which suspends its corresponding main body.

A.2.5 Continuation following an “initiate and confirm” statement

An “initiate and confirm” statement can have an explicit “continuation test” as part of its definition. This continuation test consists of up to three couplets of confirmation status and corresponding continuation action (one couplet for each confirmation status). In the event that no continuation test is specified (or less than three couplets are specified), default continuation actions are defined.

Figure A-5 presents the default, allowed and forbidden combinations of confirmation status and continuation action for “initiate and confirm” statements that are part of a main body.

NOTE The allowed combinations of confirmation status and continuation action can be extended for special testing scenarios. For example, a test can be designed to validate the rejection of a critical command, such as “deploy solar array”, where the successful outcome of the test is that the corresponding activity is "**aborted**". In this case, the continuation action "**continue**" can be defined for the confirmation status "**aborted**".

Figure A-6 presents the default, allowed and forbidden combinations of confirmation status and continuation action for “initiate and confirm” statements that are part of a watchdog.

Documents
Document Sharing Hub
Full document details: DIN EN 16603-70-32:2014-12

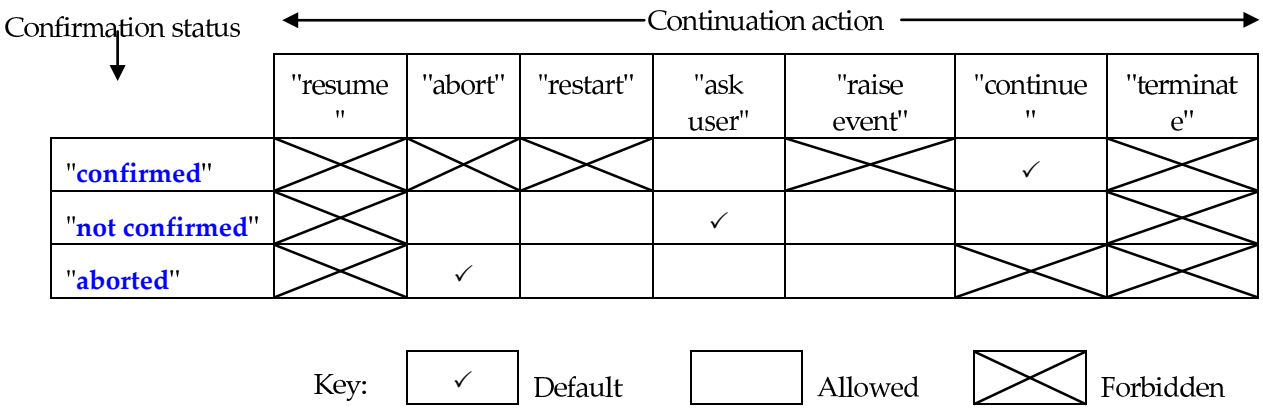


Figure A-5: Confirmation status and continuation action combinations for main body "initiate and confirm" statements

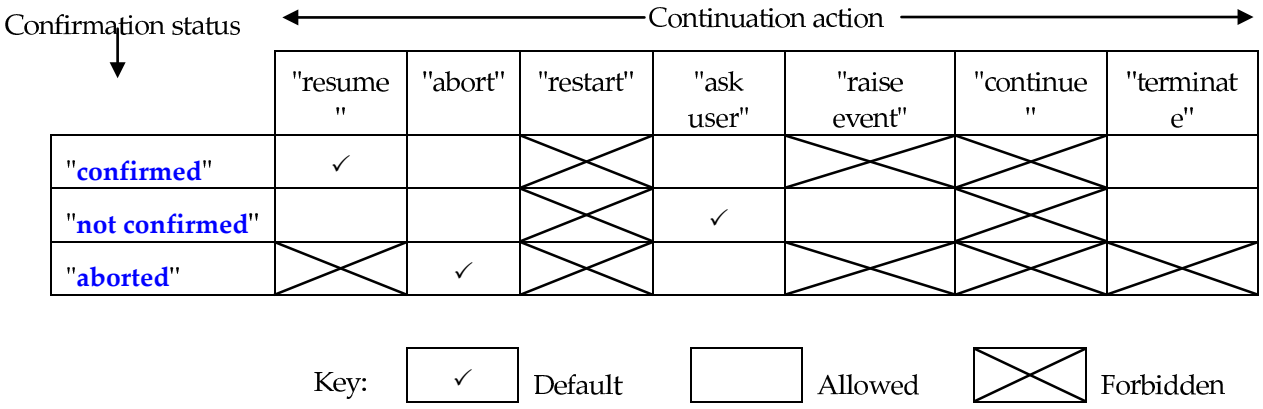


Figure A-6: Confirmation status and continuation action combinations for watchdog "initiate and confirm" statements

The meaning of the different continuation actions is as follows:

- "resume"

  Only used within the continuation test of a watchdog "initiate and confirm" statement. It resumes the execution of the main body of the procedure (or step) at the place at which it was suspended when the contingency was detected.

- "abort"

  Aborts the procedure (or step).

- "restart"

  Only used within the continuation test of a main body "initiate and confirm" statement and it restarts the corresponding "initiate and confirm" statement. To avoid a potentially indefinite loop, a timeout or a maximum number of iterations can be specified. The timeout is expressed as a relative time from the start of execution time of the

Documents
Document Sharing Hub
Full document details: DIN EN 16603-70-32:2014-12

statement. If the timeout or the maximum number of iterations is exceeded then either:

— an event is raised which terminates the statement and is then handled by a corresponding watchdog step, or

— the parent procedure (or step) is aborted.

- "ask user"

  Asks the user to specify how to proceed.

  For a main body “initiate and confirm” statement, the user may select any continuation action except "**resume**", "**ask user**" and "**terminate**". For a watchdog “initiate and confirm” statement, the user may select any continuation action except "**continue**" and "**ask user**".

- "raise event"

  Raises a local event, which is caught by a watchdog step of the procedure (or step). When the watchdog step has performed its function, the execution of the main body of the procedure (or step) resumes at the place at which it was suspended when the watchdog was triggered.

- "continue"

  Only used within a continuation test of a main body “initiate and confirm” statement. It implies that the procedure (or step) continues to be executed according to the content of the main body.

- "terminate"

  Only used within the continuation test of a watchdog “initiate and confirm” statement. It prematurely ends the execution of the procedure (or step) main body and starts the execution of the confirmation body.

In the case of a watchdog “initiate and confirm step” statement, the step itself is re-initiated unless the continuation action is "**abort**".

A.3 PLUTO language definition

A.3.1 Conventions

The terminology used in defining the syntax of the procedure language is compatible with ISO/IEC 14977.

The syntax of the procedure language is a set of rules, collectively known as a grammar.

Each rule defines a construct of the language, known as a “non-terminal symbol”. A non-terminal symbol is a combination of zero or more non-terminal symbols and “terminal symbols”. A terminal symbol is a sequence of one or more characters forming an irreducible element of the language.

A terminal symbol may consist of one or more words separated by one or more separators. The space, tab and end-of-line characters are separators.

Documents
Document Sharing Hub
Full document details: DIN EN 16603-70-32:2014-12