



Space engineering

Engineering design model data exchange (CDF)

**ECSS Secretariat
ESA-ESTEC
Requirements & Standards Division
Noordwijk, The Netherlands**

Foreword

This document is one of the series of ECSS Technical Memoranda. Its Technical Memorandum status indicates that it is a non-normative document providing useful information to the space systems developers' community on a specific subject. **It is made available to record and present non-normative data, which are not relevant for a Standard or a Handbook.** Note that these data are non-normative even if expressed in the language normally used for requirements.

Therefore, a Technical Memorandum is not considered by ECSS as suitable for direct use in Invitation To Tender (ITT) or business agreements for space systems development.

Disclaimer

ECSS does not provide any warranty whatsoever, whether expressed, implied, or statutory, including, but not limited to, any warranty of merchantability or fitness for a particular purpose or any warranty that the contents of the item are error-free. In no respect shall ECSS incur any liability for any damages, including, but not limited to, direct, indirect, special, or consequential damages arising out of, resulting from, or in any way connected to the use of this Technical Memorandum, whether or not based upon warranty, business agreement, tort, or otherwise; whether or not injury was sustained by persons or property or otherwise; and whether or not loss was sustained from, or arose out of, the results of, the item, or any services that may be provided by ECSS.

Published by: ESA Requirements and Standards Division
ESTEC, P.O. Box 299,
2200 AG Noordwijk
The Netherlands
Copyright: 2010 © by the European Space Agency for the members of ECSS

Change log

ECSS-TM-E-10-25A 20 October 2010	First issue
-------------------------------------	-------------

Table of contents

Change log	3
Introduction	6
1 Scope	7
2 Normative references	9
3 Terms, definitions and abbreviated terms	10
3.1 Terms defined in other ECSS documents	10
3.2 Terms specific to the Technical Memorandum.....	10
3.3 Abbreviated terms	11
4 Background and concepts	12
4.1 Background	12
4.2 Information model and reference data library.....	12
4.2.1 Introduction.....	12
4.2.2 Information model (SEIM).....	13
4.2.3 Reference data library (SERDL).....	18
4.3 Data exchange protocol	22
5 Requirements	23
5.1 Introduction.....	23
5.2 General requirements.....	23
Annex A (informative) SEIM-based data model for exchange	24
Annex B (informative) Space Engineering Reference Data Library	25
Annex C (informative) Web Services Interface	26
Annex D (informative) Margins and reference frames	27
D.1 Margins.....	27
D.2 Reference Frames.....	28
Bibliography	30

Figures

Figure 4-1 Top level view of the SEIM (UML package diagram)	13
Figure 4-2 SEIM main information object types (UML class diagram)	14
Figure 4-3 SEIM main information object types and relationships (informal UML class diagram).....	15
Figure 4-4 SEIM system decomposition and associated modes (UML class diagram)	16
Figure 4-5 SEIM representation of concurrent design parameters (UML class diagram)	17
Figure 4-6 SERDL CDRole and Discipline instances (UML class / object diagram)	18
Figure 4-7 SERDL LifeCyclePhase and CDActivityPhase instances (UML class / object diagram).....	19

Tables

Table 4-1 Examples of unit names and symbols for a selection of kinds of quantity from ECSS-E-ST-70-32 Annex B.....	21
---	----

Introduction

Concurrent design facilities – created following the model of the ESA Concurrent Design Facility (CDF) – will provide a capability to use engineering design models interactively for design, design iterations, and design reviews. Various space concurrent design centres exist and are evolving in the European space environment; a standard model based data exchange will allow more effective engineering design collaboration and iterations. Several cases have already required the exchange of design models and a common basis for exchange of model designs in early design phases is becoming essential to the space community.

This Technical Memorandum addresses this by separating the technical aspects of model data exchange from the issues connected with process, maintenance and transfer of such information.

1 Scope

This Technical Memorandum, under the ECSS-E-10 System Engineering in the engineering branch of ECSS series of documents, defines the recommendations for model based data exchange for the early phases of engineering design.

This Technical Memorandum facilitates and promotes common data definitions and exchange among partner Agencies, European space industry and institutes, which are interested to collaborate on concurrent design, sharing analysis and design outputs and related reviews. This comprises a decomposition of a system down to equipment level and related defined lists of parameters and disciplines. Further it provides the starting point of the space system life cycle defining the parameter sets needed to cover all project phases, although the present Technical Memorandum only addresses Phases 0 and A.

When viewed in a specific project context, the information provided in this Technical Memorandum should be adapted to match the specific need of a particular profile and circumstances of a project.

Ultimately, the objectives of the work can be summarised in three points:

- Creation of concurrent design facilities using a common / compatible data model;
- Enable effective data transfer across models belonging to different but compatible organizations / facilities;
- Enable real-time collaboration and joint activities among multiple organizations / facilities.

The main parts of this Technical Memorandum are:

- Clause 4 specifies general information to achieve the objectives.
- Clause 5 recommends a list of requirements for exchange of engineering design model data amongst concurrent design activities.
- Annex A defines the Space Engineering Information Model (SEIM) main objects, such as Space Engineering Organisation, Processes, Products.
- Annex B defines the Space Engineering Reference Data Library (SERDL). The SERDL contains names, descriptions, dimensions, units and unique codes for the set of parameters defining a Concurrent Engineering design.

This SEIM and SERDL are implemented in the ESA Open Concurrent Design Server (OCDS) project, which realizes an implementation of this data model.

This Technical Memorandum is intended to evolve into an ECSS Standard in the near future. For the time being, it is not yet possible to establish a standard that has the maturity and industrial validation required for application in new or running space projects. In conjunction with related development and validation activities, this Technical Memorandum should be regarded as a mechanism for reaching consensus prior to building the standard itself.

2

Normative references

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

ECSS- S-ST-00-01	ECSS System – Glossary of terms
ECSS-S-ST-00	ECSS System – Description, implementation and general requirements
ECSS-E-ST-10	Space engineering – System engineering general requirements
ECSS-M-ST-10	Space project management – Project planning and implementation

Terms, definitions and abbreviated terms

3.1 Terms defined in other ECSS documents

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

3.2 Terms specific to the Technical Memorandum

3.2.1 Concurrent Design Facility (CDF)

multidisciplinary design centre using the concurrent engineering approach for the assessment of potential future ESA missions

NOTE The CDF is located at ESA/ESTEC.

3.2.2 Integrated Design Model (IDM)

federation of information models and engineering tools supporting the concurrent design of a space mission in phases 0 and A

NOTE The IDM was developed at the CDF.

3.2.3 concurrent engineering approach

systematic approach to integrated multi-disciplinary system development that emphasises the response to customer expectations and embodies team values of co-operation, trust and sharing in such a manner that decision making is by consensus, involving all perspectives in parallel from the beginning of the system life-cycle

3.2.4 Open Concurrent Design Server (OCDS)

implementation of a database system that complies with this Technical Memorandum and related software applications to support concurrent engineering

NOTE The OCDS is initially developed for the CDF under ESA contract.

3.3 Abbreviated terms

The following abbreviations are defined and used within this technical memorandum:

Abbreviation	Meaning
CD	Concurrent Design
CDF	Concurrent Design Facility
CE	Concurrent Engineering
EIRP	Equivalent Isotropically Radiated Power
FDIR	Failure Detection, Isolation and Recovery
G/T	Gain to Noise
GS	Ground Station
HGA	High Gain Antenna
HK	Housekeeping
ICT	Information and Communication Technology
IDM	Integrated Design Model
ISO	International Organization for Standardization
LEO	Low Earth Orbit
LGA	Low Gain Antenna
MGA	Medium Gain Antenna
OCDS	Open Concurrent Design Server
OMG	Object Management Group
S/C	Spacecraft
S/S	Subsystem
SA	Solar Array
SEIM	Space Engineering Information Model
SERDL	Space Engineering Reference Data Library
TRL	Technology Readiness Level
UML	Unified Modeling Language (OMG)

Background and concepts

4.1 Background

Since 1996 ESA has been developing and using the concurrent engineering approach for life cycle phases 0 and A, realized in the Concurrent Design Facility (CDF). It was based on the concepts originally developed at NASA/JPL in their Team-X facility. In order to support a standard process the CDF developed the so-called Integrated Design Model (IDM), which defines a process and the data items that are needed to support concurrent design sessions.

The concurrent engineering approach has been so successful, that many organizations in the European space sector have set up their own concurrent engineering facilities. With multiple facilities now established and a natural desire to collaborate, the need arose to be able to exchange study data between facilities. Since each facility needs to optimize its operations to fulfil its specific needs, there are unavoidable differences in the supporting ICT. Therefore data exchange can only be successful if it is based on an open neutral data exchange protocol.

This Technical Memorandum defines a neutral data model and a set of predefined data definitions based on the IDM to enable data exchange and collaboration between concurrent engineering facilities for life cycle phases 0 and A.

4.2 Information model and reference data library

4.2.1 Introduction

In order to understand what data needs to be exchanged first an information model is given that describes the main processes and data items that are needed for concurrent engineering in phases 0 and A. This information model is the result of an information analysis activity on the basis of the IDM.

The information to be exchanged has basically the same structure for each concurrent design study. In addition there are many data items that describe and characterize the study and the system being studied: e.g. system element names and their composition, properties of parts and materials, design input parameters, analysis or simulation result parameters, names of participating disciplines.

In order to prepare for future evolution of the content of the data to be exchanged the actual data exchange protocol is split into two parts:

- a core data model that defines the main object types and the relationships between them, i.e. the structure of data to be exchanged;
- a reference data library that defines a set of predefined instances of the object types that are common to many different concurrent design studies.

The core data model is expected to be stable over relatively long periods of time and forms the basis for implementations of data exchange interfaces. The reference data library is expected to be made available through internet and to be extended at regular intervals when the need for additional data items (e.g. design parameters) arises.

4.2.2 Information model (SEIM)

4.2.2.1 Top level view – main partitioning

The information model is called the System Engineering Information Model (SEIM).

Figure 4-1 shows a top level view with the 5 packages that group the concepts in the model and the main relationships between them.

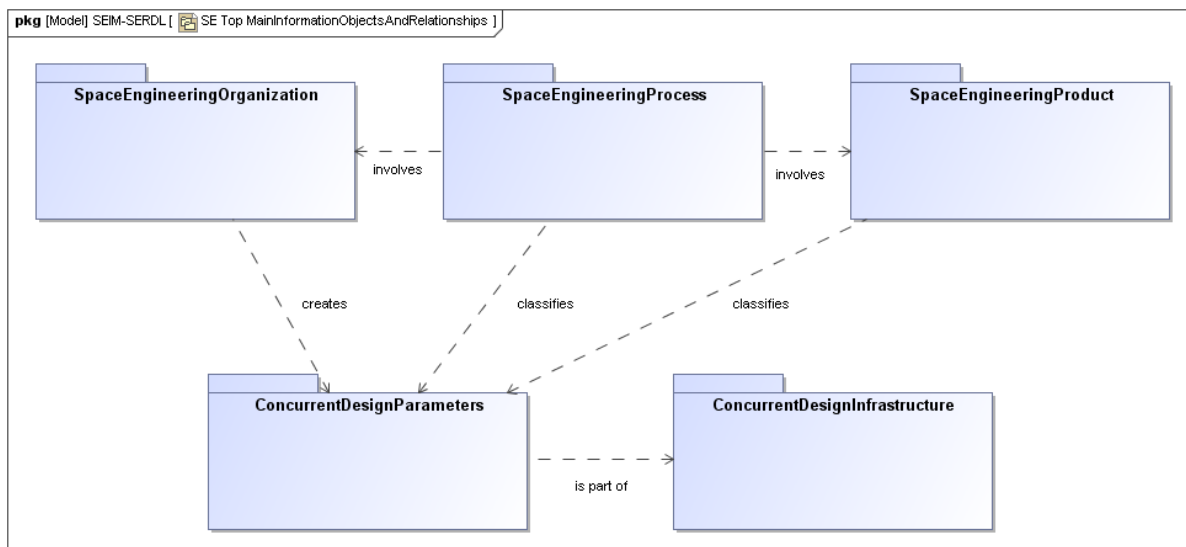


Figure 4-1 Top level view of the SEIM (UML package diagram)

NOTE All diagrams in this Technical Memorandum use UML2 notation.¹

The Space Engineering Process involves a Space Engineering Organization to carry out the process, and a Space Engineering Product to be designed. The (various Disciplines of the) Organization creates a set of Concurrent Design

¹ An introduction to UML (graphical) notation can be found in many books on UML or at: http://en.wikipedia.org/wiki/Unified_Modeling_Language#Diagrams_overview

Parameters (with values and meta-data) which are classified in terms of the process steps and/or product part they describe. The Concurrent Design Parameters are stored and managed in a Concurrent Design Infrastructure (e.g. OCDS).

4.2.2.2 Main information object types

Each of the packages contains a number of concepts that represent the main information objects needed to capture a concurrent design study. Figure 4-2 identifies the main object types in each of the packages. The object types are grouped by the following classifiers (denoted as UML stereotypes):

- organisationClass
- processClass
- objectClass
- dataClass
- facilityClass

These classifiers can later be used for mapping onto reference data library concepts.

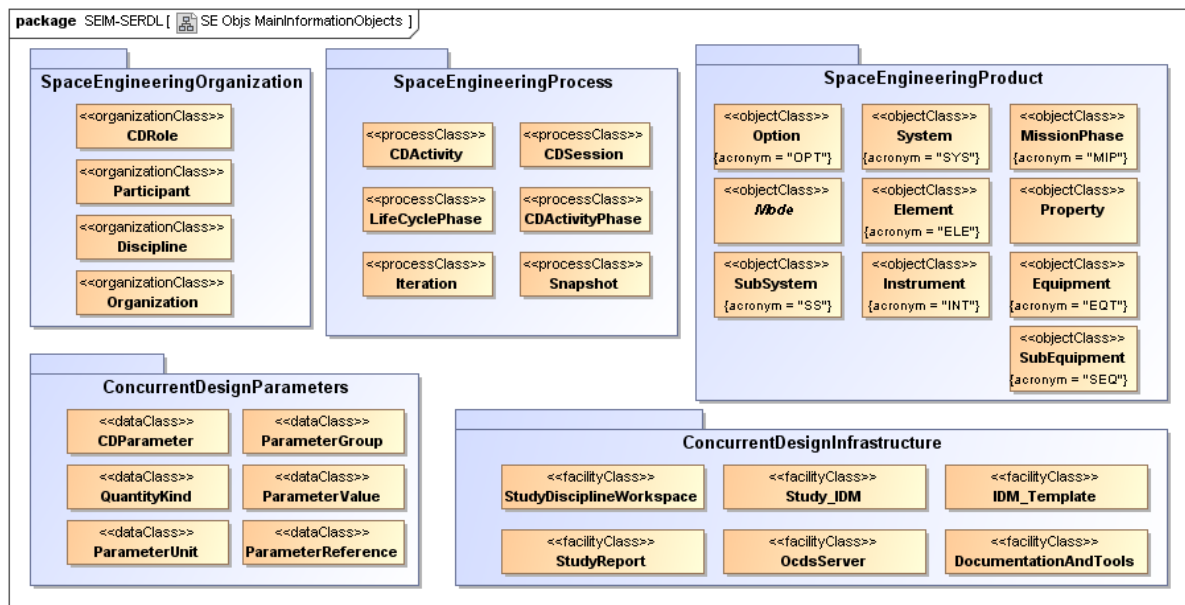


Figure 4-2 SEIM main information object types (UML class diagram)

The object types need to be completed with information on how they are related. Figure 4-3 therefore shows the main relationships. The object types and the relationships together provide a high level informal view of the SEIM.

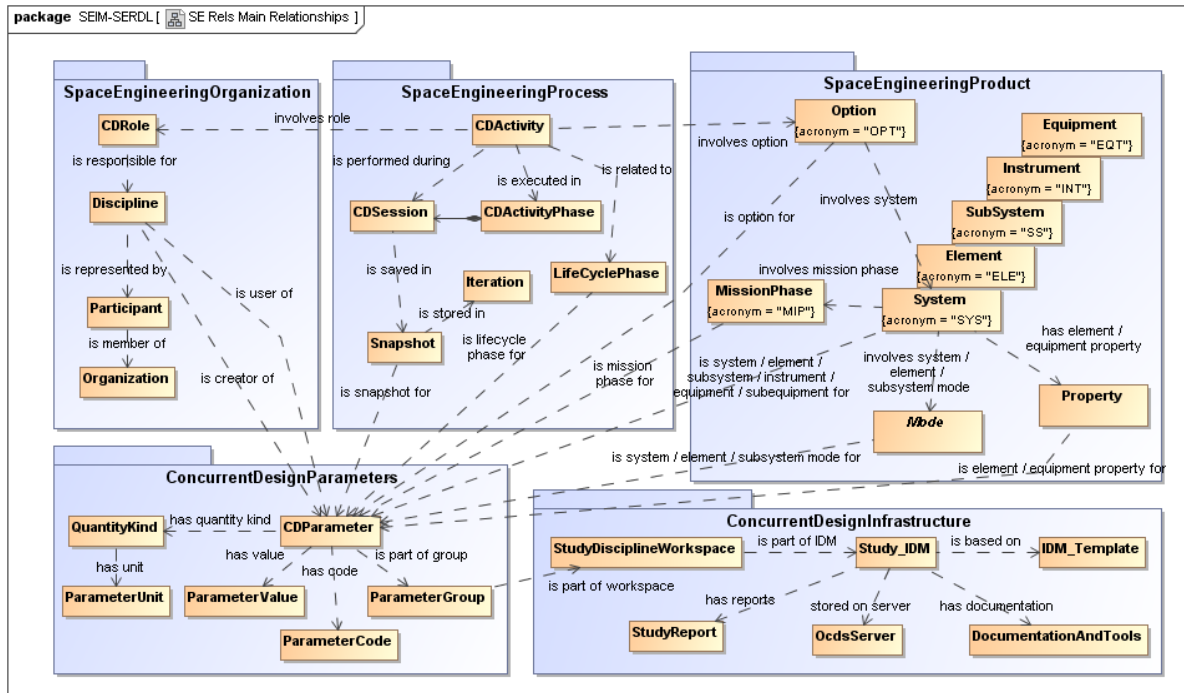


Figure 4-3 SEIM main information object types and relationships (informal UML class diagram)

4.2.2.3 SEIM representation of system decomposition

In order to manage complexity and to allow design work taking place concurrently on different aspects or parts of the system under study, hierarchical decomposition of the system is needed. This is well established engineering practice for complex systems. Figure 4-4 shows how a system can be decomposed in the SEIM. This is a limited way of decomposing systems that is considered adequate for the conceptual design in phases 0 and A:

- An *Option* represents a certain design option considered in the study. There may be more than one *Option* for a given *System*.
- A *System* represents a top level system. A *System* may be decomposed into a number of child systems (through the *hasSystem* relationship), in order to enable a simplistic system of systems representation.
- At the next lower level a *System* may contain a number of *Element*.
- At the next lower level each *Element* may contain a number of *Equipment*. Additionally (or alternatively) an *Element* may contain a number of *Instrument*, which in turn may also contain a number of *Equipment*, but an *Equipment* may only be part of (i.e. owned by) either an *Element* or an *Instrument*, not both.
- An *Element* may also contain a number of *SubSystem*. A *SubSystem* is different from the other decomposition object types in the sense that it is a *logical* grouping of *Equipment*, in contrast with the other decomposition object types that constitute a *physical* decomposition. Therefore a *SubSystem* may refer to (but not own) a number of *Equipment*.
- Finally an *Equipment* may contain a number of *SubEquipment*.

Furthermore, Figure 4-4 shows how different operational modes may be associated with a system at the levels of *System*, *Element* and *SubSystem*.

For each of the object types an acronym is defined that will later be used to construct parameter short names (also known as parameter codes).

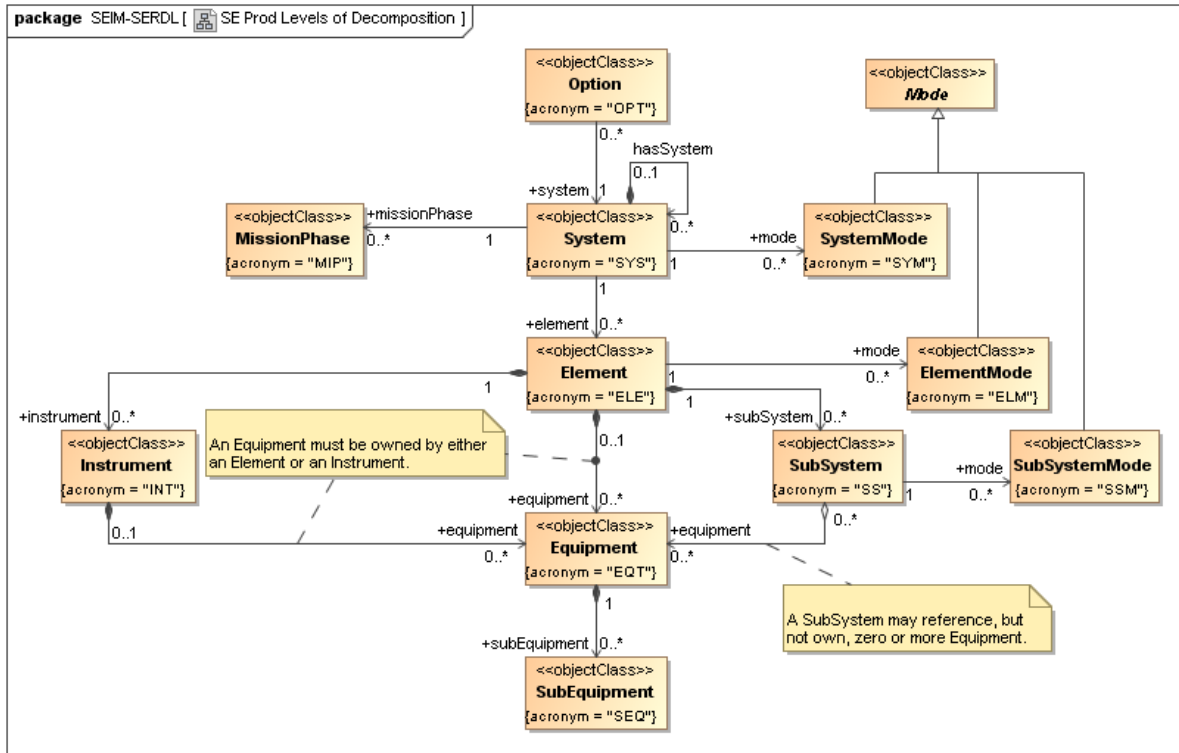


Figure 4-4 SEIM system decomposition and associated modes (UML class diagram)

4.2.2.4 SEIM representation of concurrent design parameters

Parameters are essential to capture the conceptual design in a concise and flexible way. Design parameters with their values form the basis of capturing the state of a concurrent design, and communicating this state between all participating disciplines. Figure 4-5 shows how concurrent design parameters are represented:

- The central object type is *CDParameter* which represents a concurrent design parameter and has attributes and relationships to other object types as listed hereafter. A *CDParameter* has a *name* and *description*.
- *ParameterGroup* may be used to group a set of parameters in useful ways, e.g. by system decomposition level.
- A *CDParameter* has a *QuantityKind* which specifies what kind of quantity the parameter represents. A *QuantityKind* may be e.g. *Length*, *Mass*, *Time*, *Energy*, *Power*, etc., as well as the *quantity of dimension one*, also known as the *dimensionless quantity*.

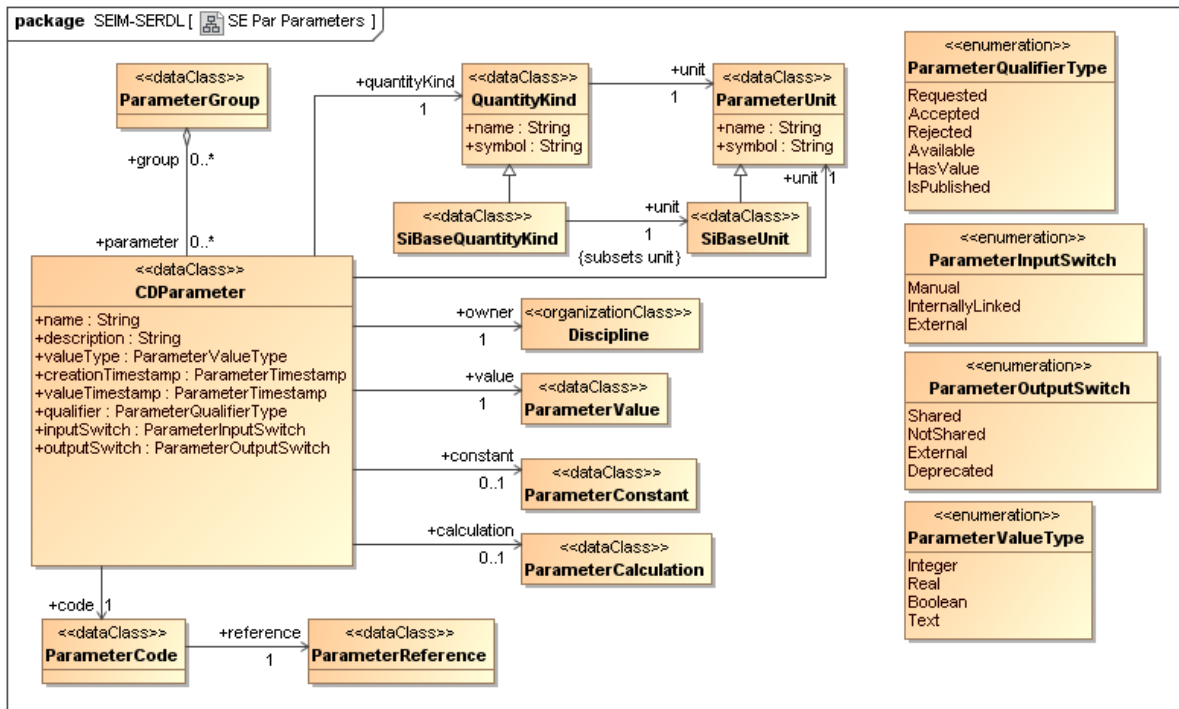


Figure 4-5 SEIM representation of concurrent design parameters
(UML class diagram)

- A *CDParameter* also has a *Unit*, which specifies the measurement unit in which the parameter value is expressed. The *Unit* may be *1*, also known as the *dimensionless unit*.

NOTE For detailed information on standard definitions concerning quantities, units and dimensions see [VIM].

- A *CDParameter* has a *ParameterValueType* which specifies in what kind of computer data type its value is expressed. The possible data types are *Integer*, *Real*, *Boolean* and *Text*. *Boolean* and *Text* valued parameters have by definition a *QuantityKind* that is the *dimensionless quantity* and *Unit* that is the *dimensionless unit*.
- A *CDParameter* has a *ParameterCode* which is a coded short name to refer to the parameter. Clause 0 explains the naming convention that is used for the codes.
- A *CDParameter* may have an associated *ParameterConstant* or a *ParameterCalculation*, but not both. A *ParameterConstant* defines a constant value. A *ParameterCalculation* defines a mathematical expression that prescribes the value of the parameter dependent on other parameters, as in a spreadsheet formula.
- A *CDParameter* has an *owner* which is the *Discipline* (see Figure 4-3) that is the first responsible for the parameter.
- Finally, a *CDParameter* has some housekeeping attributes:
 - *creationTimestamp*, which specifies the date and time when the parameter was created.

- *valueTimestamp*, which specifies the last date and time when the parameter value was set.
- *qualifier*, which specifies the state of the parameter in the concurrent design process
- *inputSwitch*, which specifies how the parameter is treated as an input parameter for the *owner Discipline*.
- *outputSwitch*, which specifies how the parameter is treated as an output parameter for the *owner Discipline*.

4.2.3 Reference data library (SERDL)

4.2.3.1 Purpose

The reference data library is called the System Engineering Reference Data Library (SERDL). The SERDL defines a set of predefined instances of the SEIM object types that are common to many different concurrent design studies. The SERDL will be made available through internet and can be extended at regular intervals when the need for additional data items (e.g. design parameters) arises.

4.2.3.2 Organization roles and disciplines

The SERDL defines the *CDRoles* and *Disciplines* as in Figure 4-6 below.

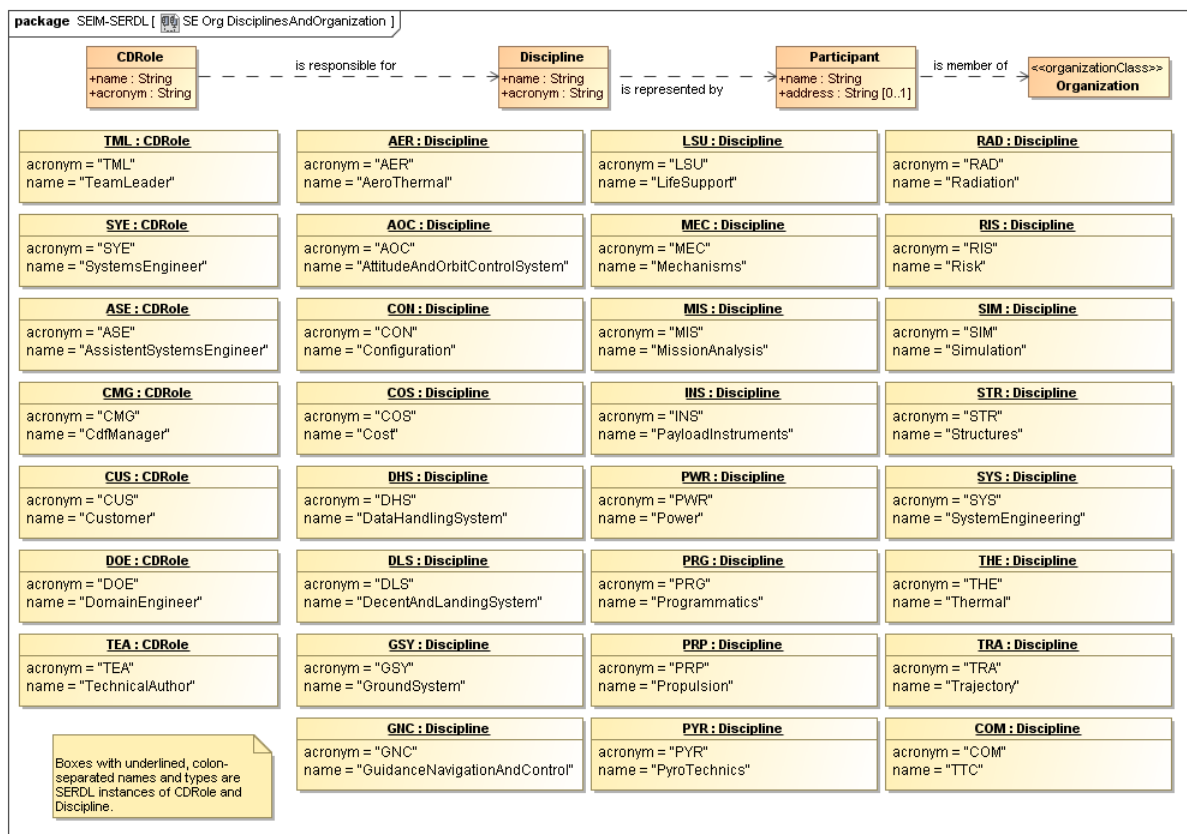


Figure 4-6 SERDL CDRole and Discipline instances (UML class / object diagram)

4.2.3.3 Process related definitions

The SERDL defines the process related instances of *LifeCyclePhase* and *CDActivityPhase* as in Figure 4-6 below.

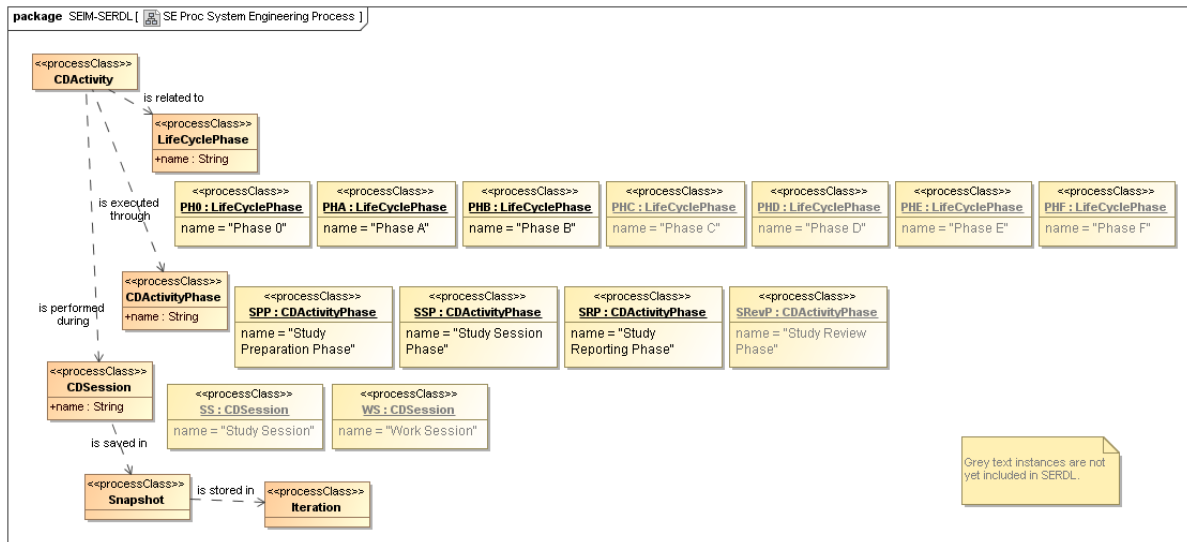


Figure 4-7 SERDL LifeCyclePhase and CDActivityPhase instances (UML class / object diagram)

4.2.3.4 ParameterCode naming convention

Parameters are named according to the data structure hierarchy. Generally the parameter is named according to the origin of the parameter or a calculation associated with that result. For example, some parameters require a name linked to a particular discipline including mass, the number of units or a mass margin. In this case the subsystem to which the parameter belongs is provided. To ensure concurrence between different organisations, the source of the parameter calculation should be declared and agreed. For example, the propellant budget could be calculated at *System* level or by the owner discipline "Propulsion" as part of the "Propulsion" *SubSystem*.

Some parameters are provided by configuration, for example CoG (centre of gravity) is linked to the location and position of equipments noting that for multiple equipments this can be different.

The SERDL system decomposition hierarchy has the following levels, that are given in the order in which they appear in the code naming convention:

1. Option
2. System
3. MissionPhase
4. SystemMode
5. Element
6. ElementMode
7. ElementProperty

8. Subsystem
9. SubsystemMode
10. Instrument
11. Equipment
12. EquipmentProperty
13. SubEquipment

The codes for instances in the decomposition hierarchy are built by concatenating the acronyms for each of the system decomposition levels with numbers (denoted by #) and *ParameterReference* names (denoted by PRname), joined by the underscore character. The following list enumerates a subset of valid codes:

- OPT#_SYS#_PRname
- OPT#_SYS#_MIP#_PRname
- OPT#_SYS#_MIP#_SYM#_PRname
- OPT#_SYS#_MIP#_SYM#_ELE#_PRname
- OPT#_SYS#_MIP#_SYM#_ELE#_ELM#_PRname
- OPT#_SYS#_MIP#_SYM#_ELE#_ELP#_PRname
- OPT#_SYS#_MIP#_SYM#_ELE#_ss_PRname
- OPT#_SYS#_MIP#_SYM#_ELE#_ss_SSM#_PRname
- OPT#_SYS#_MIP#_SYM#_ELE#_INT#_PRname
- OPT#_SYS#_MIP#_SYM#_ELE#_INT#_EQT#_PRname
- OPT#_SYS#_MIP#_SYM#_ELE#_EQT#_PRname
- OPT#_SYS#_MIP#_SYM#_ELE#_EQT#_EQP#_PRname
- OPT#_SYS#_MIP#_SYM#_ELE#_EQT#_SEQ#_PRname

Examples and notes:

NOTE 1 Thus, for example, the mass of Element 1 gets the code "ELE1_mass".

NOTE 2 Decomposition levels that apply equally to all instances may be dropped from the code. Typically, OPT# and SYS# is often dropped in the names/codes of most *Element* and *Equipment* level parameters.

NOTE 3 The "SS_" is not replaced by relevant Discipline. Instead the *ParameterReference* instance has *Discipline* as creator.

NOTE 4 Eigenfrequencies are treated as subtypes of *ElementProperty* "ELP#".

NOTE 5 Ground Station, Solar Array, Battery and Antenna are treated as subtypes of *Equipment* "EQT#".

NOTE 6 Coverglass Thickness and Dose Depth are treated as subtypes of *EquipmentProperty* "EQP#".

4.2.3.5 Unit names and symbols

The names and symbols of units (*Unit* instances) comply with the International System of Units (SI) as defined in [ISO/IEC80000]. In order to be compatible with a wide range of software applications the unit symbols are encoded in ASCII character strings complying with the specification in Annex B of ECSS-E-ST-70-32. Table 4-1 shows examples for some units.

Table 4-1 Examples of unit names and symbols for a selection of kinds of quantity from ECSS-E-ST-70-32 Annex B

quantity name	unit name	unit symbol
acceleration	metre per square second	m/s ²
angular momentum	kilogram square metre per second	kg.m ² /s
angular velocity	degree per second	deg/s
angular velocity	radian per second	rad/s
area	square metre	m ²
coefficient of heat transfer	watt per square metre kelvin	W/(m ² .K)
density, mass density	kilogram per cubic metre	kg/m ³
moment of inertia	kilogram square metre	kg.m ²
rotational frequency	reciprocal second	s ⁻¹
thermal conductivity	watt per metre kelvin	W/(m.K)
transmission rate	bit per second	bit/s
velocity, speed	metre per second	m/s
volume	cubic metre	m ³
wavenumber	reciprocal metre	m ⁻¹

4.3 Data exchange protocol

Building on top of the SEIM and the SERDL the actual exchange of engineering design model data will be implemented according to the following protocol. There are two potential modes of operation:

1. Off-line exchange of a dataset that comprises the data (in whole or in part) of a concurrent design activity (*CDA*ctivity). This is a classical file export / import scenario.
2. On-line exchange of data between two concurrent design facilities through a web services connection. This is a more dynamic exchange scenario where datasets are exchanged on demand through messages between software applications.

For both modes of operation the structure of the dataset and its content is defined in the data model specified in Annex A and the reference data library specified in Annex B.

Additionally, for on-line exchange the web services are defined in Annex C.

5 Requirements

5.1 Introduction

Clause 5 contains a non-exhaustive list of requirements for the Engineering design model data exchange. This list will be elaborated in the future standard.

These requirements should be complied with when two or more parties want to exchange engineering design model data for a concurrent design activity.

Annex D provides guidelines on the definition and usage of margins and reference frames.

5.2 General requirements

- a. The parties shall agree on the disciplines (*Discipline*) that participate in the concurrent design activity.
- b. The parties shall agree on the roles (*CDRole*) that participate in the concurrent design activity.
- c. The parties shall agree on the system decomposition to be used, including the names at all decomposition levels.
- d. The parties shall agree on the mandatory set of parameters to be present in the reference data library or exchange dataset, starting from the minimal set of parameters specified in Annex B.
- e. The parties shall agree on the owner discipline for each parameter, starting from the default ownerships specified in Annex B.
- f. The parties shall agree on the unit in which the value for each parameter will be expressed, starting from the default units specified in Annex B.
- g. The parties shall agree on off-line or on-line mode of data exchange, or both.
- h. If on-line data exchange is selected, the parties shall provide each other with details on how to connect to each other's web services implementations.
- i. The parties shall agree on the specific data format to be used from the options specified in Annex A.

Annex A (informative)

SEIM-based data model for exchange

The dataset for exchange of design model data should comply with the data model defined at:

<http://www.purl.org/ecss/ecss-e-tm-10-25/annex-a>

For convenience the data model is provided in three different data modelling technologies:

1. UML model – annotated class diagrams for overview and easy readability.
2. ISO 10303-11 EXPRESS SCHEMA specification.
3. XML Schema (XSD) specification, automatically generated from the EXPRESS SCHEMA.

The exchange dataset may be encoded in one of the following formats:

1. ISO 10303-21 file, also known as "Part 21" or "STEP Physical File" format.
2. XML file, in compliance with the XML Schema specification.

Annex B (informative)

Space Engineering Reference Data Library

The Space Engineering Reference Data Library (SERDL) for exchange of design model can be found from <http://www.purl.org/ecss/ecss-e-tm-10-25/annex-b>

Annex C (informative) Web Services Interface

This annex defines the web services interface to be used to exchange engineering design model data between two concurrent design facilities that are connected on-line via a computer network.

The web services interface is specified in WSDL at the following URL:

<http://www.purl.org/ecss/ecss-e-tm-10-25/annex-c>

Annex D (informative)

Margins and reference frames

D.1 Margins

- a. Appropriate and consistent design margins shall be defined for the spacecraft system and components.
- b. Design margins are recommended as follows during the analysis and design processes as follows:
 - 5 % for fully developed items
 - 10 % for items to be modified
 - 20 % for items to be developed

In principle any margin percentage is acceptable, but an explanation shall be provided whenever the selected value differs from one of the possibilities given above. The subsystem margin is the weighted average of the equipment margins computed as:

$$margin_{subsystem} = \frac{\sum_{all} X_{equipment} * margin_{equipment}}{\sum_{all} X_{equipment}}$$

where

$X_{equipment}$ is the related parameter (Mass or Power) of each equipment without margin.

If detail at equipment level is not available, overall sub-system masses can be given. In this case margins are defined according to the same criteria as for the equipments.

- a. During analysis and design a 20 % system margin shall be provided for the Dry Mass of the Spacecraft System to be launched. This covers uncertainties and future changes in requirements.
- b. A Propellant Margin computed against the Spacecraft system Dry Mass including the System Margin being 5 % for manoeuvres and 100 % for the AOCS thruster propellant where applicable.
- c. A Launcher margin is advised being that the final design wet mass of the spacecraft system to be launched must be smaller or equal to the launcher capabilities, a 10 % margin is recommended.

- d. During conceptual design a Delta-V margin of at least 5 % is recommended (higher if gravity losses are not accounted for).
- e. During conceptual design a data processing margin of 50 % for mass memory and 100 % for computing power is recommended.
- f. During conceptual design a 3dB margin in the communications link budget is recommended.
- g. During conceptual design a $\pm 10^{\circ}\text{C}$ margin in equipment temperatures is recommended.
- h. During conceptual design a 20 % margin in the total required spacecraft system power budget is recommended.
- i. For each Equipment the Peak transient power consumption must be given (see EQUIPMENT in annex tables). In contrast to the mass, this must take into account the quantity/number of equipments of the Equipment. See example below:
 - Equipment Name: EPC
 - Number of equipments: 3
 - Peak transient power consumption for one EPC: 400W.

If all 3 are working at the same time (no redundancy) then the Peak power consumption should be given as $3 * 400\text{W} = 1200\text{W}$

- In case of hot redundancy, the number will be the same: 1200W
- In case of cold redundancy (if 3 equipments, typically 1 is redundant and 2 working): $2 * 400 = 800\text{W}$.

For each equipment (e.g. Equipment 2) and for each mode, Power consumption (on), Power consumption (standby) and Duty Cycle must all be specified. These parameters also have to take into account the quantity/number of equipments of the Equipment. Continuing the EPC example above:

- Power Consumption (on) for one EPC: 200 W

If all 3 are working at the same time (no redundancy) then the Power consumption (on) becomes $3 * 200\text{W} = 600\text{W}$

- In case of hot redundancy, the number will be the same: 600W
- In case of cold redundancy (if 3 equipments, typically 1 is redundant and 2 working): $2 * 200 = 400\text{W}$.

The Duty cycle is calculated with respect to the Mode duration, and the on-time of the Equipment during the mode. If the Equipment is on during the whole mode, the Duty cycle is 100 % and if it is off all the time the Duty Cycle is 0 %.

D.2 Reference Frames

In general reference frame definition shall comply with ECSS-E-ST-10-09.

- a. Reference frames shall be defined for spacecraft system design.
- b. A reference frame for mission phases and operations modes.

- c. A reference frame for mechanical design.
- d. A reference frame for control design.
- e. Typical project reference frames include the following:
 - 1. Local orbital reference frame.
 - 2. Spacecraft reference frame.
 - 3. Spacecraft centre of gravity reference frame.
 - 4. Spacecraft Sensors reference frames.
 - 5. Spacecraft attitude reference frame.
 - 6. Payload reference frame.
 - 7. Payload instruments reference frames.

Bibliography

ECSS-E-TM-10-21	System engineering — System modelling and simulation
ECSS-E-TM-10-23	System engineering — Engineering database
ECSS-E-ST-10-09	System engineering – Reference coordinate system
ECSS-E-ST-70-32	System engineering – Test and operations procedure language
ECSS-S-ST-00	ECSS system – Description, implementation and general requirements
IEC 60027-2	Letter symbols to be used in electrical technology – Part 2: Telecommunications and electronics (Third edition 2005-08) Specifies the electrical and telecommunications quantities and units in addition to SI.
ISO 31	Quantities and units (Third edition 1992-08-01) Specifies the international system of units (SI) in 14 parts.
ISO/IEC 80000	Quantities and units 15 parts, some published, some still in progress. This is the harmonized replacement of ISO 31 and IEC 60027, the new international system of quantities and units.
ISO 10303-11	Industrial automation systems and integration – Product data representation and exchange – Part 11: The EXPRESS language reference manual (second edition, 2004)
ISO 10303-21	Industrial automation systems and integration – Product data representation and exchange – Part 21: Implementation methods: Clear text encoding of the exchange structure (second edition, 2002)
OCDS-DNV-D1.1.2	OCDS-DNV-D1.1.2-Ver4.1 Space Engineering Information Model
OCDS-DNV-D1.1.3	OCDS-DNV-D1.1.3-Ver4.1 Space Engineering Reference Data Library
SysML	System Modeling Language (SysML), OMG http://www.omg.sysml.org
UML	Unified Modeling Language, version 2, OMG, http://www.uml.org

VIM	JCGM 200:2008, International Vocabulary of Metrology – Basic and General Concepts and Associated Terms (VIM), 3rd edition, 2008, BIPM, Paris, France. Available in PDF format from BIPM’s website http://www.bipm.org
WSDL	Web Services Description Language, World Wide Web Consortium (W3C) http://www.w3.org/2002/ws/desc
XML	Extensible Markup Language, World Wide Web Consortium (W3C) http://www.w3.org/XML
XSD	XML Schema, World Wide Web Consortium (W3C) http://www.w3.org/XML/Schema