Rapid Prototyping of Application-Specific Signal Processors (RASSP)

BUILD 2

APPLICATION INTERPRETED MODEL REPORT

Authorization: Subcontract No. P.O. TTM 748357

_

Responsible Organization:

Information Requirements and Analysis

Advanced Information Engineering

Approved By:

F. L. Bsharah J. F. Willis (Concurrence)

Project Engineer Manager

Information Requirements and Analysis Advanced Information Engineering

Advanced Information Engineering

Distribution: Authorized only to U.S. Government agencies and private individuals or enterprises eligible to obtain export controlled technical data in accordance with regulations implementing 10 U.S.C. 140c as of 31 March 1987. Other requests may be referred to the Director, Defense Advanced Research Projects Agency.

Warning: This document may contain technical data whose export is restricted by the Arms Export Control Act (Title U.S.C. Sec. 2751 et seq.) or Executive Order 12470. Violators of these export control laws are subject to severe criminal penalties.

Destruction Notice: Destroy by any method that will prevent disclosure of contents or reconstruction of the document.

Date: 20 March 1996

No. of Pages: x + 104

Revision/Change Log

Revision	Change	Date	Driver (CR/DR)	Description
Baseline		23 December 1994		Initial Release, Version 1
A	Added application interpreted objects and assertion per SCRA DM2 requirements.	20 March 1996		Build 2 Release

Contents

Section Page

3.1.49 discrepant_product

3.1.50 document

3.1.51 document_reference

3.1.52 drawing

3.1.53 document_type

3.1.54 enhancement_product

3.1.55 enterprise

3.1.56 file_folder

3.1.57 group

3.1.58 hardware_software

3.1.59 local_time

3.1.60 lot_effectivity

3.1.61 make_from_usage_option

3.1.62 measure_with_unit

3.1.63 named_unit

3.1.64 next_assembly_usage_occurrence

3.1.65 node_location

3.1.66 ordered_action

3.1.67 ordinal_date

3.1.68 organization

3.1.69 organizational_address

3.1.70 organizational_project

3.1.71 part

3.1.72 person

3.1.73 person_and_organization

3.1.74 personal_address

3.1.75 physical_unit

Foreword

This document has been prepared by the Information Requirements and Analysis (IR&A) group of Rockwell's Advanced Information Engineering (AIE) organization.

This is the second edition of this document.

This document identifies and defines the application interpreted information requirements of Build 2 for the Rapid Prototyping of Application Specific Signal Processors (RASSP) Enterprise Framework. The RASSP Application Interpreted Model (AIM) information model is documented in this report.

Based upon this model, the future iterrations of the RASSP Implementation Data Model will be developed.

Introduction

The Build 2 RASSP Application Interpreted Model Report identifies and defines the information requirements for the RASSP Enterprise Framework. The information requirements are represented in the RASSP Application Interpreted Model (AIM). Within the AIM, a mapping table is constructed to capture the information and business constructs from the enterprise level to the application reference level. The EXPRESS and EXPRESS-G modeling language will be used to document the AIM. For Build 1, the AIM represents configuration management principles.

Industries have a need to communicate to their suppliers, customers, clients, and contractors any product problems or anomalies, the corrections for these problems and any resulting corrective actions or changes. The products supported by the RASSP configuration management process are those for which RASSP wishes to maintain a change history such as files, discrete parts or components, assemblies, documents, and signal processors designs.

The application interpreted requirements represented by the AIM are specified in clause 1 using the enterprise level (STEP community) terminology with specializations in the RASSP Enterprise Framework terminology. A graphical representation of the application interpreted requirements, referred to as the RASSP Application Interpreted Model, is given in annex A. The mappings between the ARM and the REDM that produce the AIM, are given in Annex B.

1 Scope

This report specifies the use of the information resources, as defined by the user and enterprise, necessary for the scope and information requirements for the configuration management process. Configuration management for an item (product or file) includes the identification of the reason for a change, its cause, the approval and performance of the resulting changes to the item, and the authorization of corrective actions to prevent reoccurrence. The identified information provides configuration management support throughout the life cycle of an application specific signal processor. This support includes areas such as design, manufacture, production, and technical publication generation.

The following are within the scope of the application reference model:

— Identification of the item requiring change;

> — The classification of each item requiring change as either discrepant or needing enhancement;

— The identification of an anomaly in the form of a flaw or an issue that results from corrective, perfecting, adaptive, or preventative needs. An identified anomaly applies to a product or one or more versions of a file that requires a change;

— The specification of the tasks required implementation of a change and inspection of that changed product to verify that the change requirements have been properly implemented.

The following are outside the scope of this report:

— The usage of the change information in planning and administration functions;

— The scope of management concern.

2 Definitions and Abbreviations

This report makes use of the following terms and definitions:

anomaly: a description of either a product problem or enhancement that may result in a change requirement. The product problems are deviations from the expected product characteristics. A product enhancement identifies the need for new and or improved product characteristics. Product characteristics are the form, fit, and function properties of a product as well as any other descriptive traits.

authorization: the decision making mechanism through which the appropriate level of permission is granted to proceed with the execution of planned actions or resource allocations. A commitment or acknowledgment to perform a particular process step or series of steps.

change management: a procedure used by the functional organizations within an enterprise. The purpose is to determine which functions are impacted by a change activity and coordinate the tasks that will be involved throughout the entire change procedure.

The management of a change process is conducted in two parts: 1.) the design activity which involves all the design and administrative activities involved in the disposition of a change need and 2.) the actual implementation activity which involves the actual change process to an item requiring change. Change management includes the conceptual design, final design process, testing procedures and final delivery.

corrective action: an action taken to prevent a product anomaly from reoccurring. Corrective action may include any or all of the following steps; localization, isolation, disassembly, re-assembly, alignment, and checkout.

change requirements: the reason for the condition of changing, altering or modifying, transformation, replacing of one thing for another substitution and a transition from one state, condition, phase to another of that which is required.

product data: a single article or unit included in a collection, enumeration, or series that collectively defines a product and is specified separately from the product.

updates applicable to either product improvements and/or major modifications: update reviews should be initiated as a result of discrepancies reported on previous reviews, to provide an audit trail for follow up improvements and corrective actions. The update review should assess the present status of the fielded system against the baseline established by the previous fielded history review.

product functionality: a description of the requirement that is satisfied by the product.

related change: a change to a product that is required because of a problem, enhancement need, or corrective action associated with a related anomaly.

support resource: a product required to design, build, operate, and maintain another product. A resource may be a facility, tool, person, or documentation.

process step: a unit of specific work behavior with a clear beginning and ending. The process step describes the performance of a meaningful function.

Unit of Functionality: a grouping of objects (entities, attributes, enumerations, etc.). The Unit of Functionality (UoF) is used to organize and summarize one or more concepts of operation into reusable capabilities.

3 Application Interpreted Information Requirements

This clause specifies the application interpreted information required for the configuration management of a RASSP product (application specific signal processor).

The Application Interpreted Model (AIM) is an information model that captures the user's information requirements with respect to the RASSP Enterprise Model. The Build 2 AIM was based on the the REDM from Build 2, the Application Reference Model (ARM) from Build 1, and SCRA document MMC-RASSP-2.01.00, STEP Configuration Management Suitability Report.

The information requirements are specified as application objects and application assertions. These assertions pertain to individual application objects and to relationships between application objects.

NOTES

1 - A graphical representation of the information requirements is given in annex A.

3.1 Application Interpreted Objects

3.1.1 identifier

An identifier is an alphnumeric string which allows an individual thing to be identified. It may not provide natural language meaning.

EXAMPLE - A part_number would be an identifier.

3.1.2 label

A label is the term by which something may be referred to. It is a string which represents the human-interpretable name of something and shall have a natural language meaning.

> EXAMPLE - "Smith", "Widget Inc.", and "Materials Test Laboratory" are examples of labels.

3.1.3 text

A text is an alphanumeric string of characters which is intended to be read and understood by a human being. It is for information purposes only.

3.1.4 year_number

A year_number is the year as defined by the Gregorian Calendar.

3.1.5 day_in_month_number

A day_in_month_number is the position of the specified day in a month.

3.1.6 day_in_week_number

A day_in_week_number is the value of the day as defined in ISO 8601 (clause 5.2.3).

NOTE - Monday is day number 1, Tuesday is day number 2, Wednesday is day number 3, Thursday is day number 4, Friday is day number 5, Saturday is day number 6, and Sunday is day number 7.

Formal propositions:

WR1: the value of the integer shall be between 0 and 7.

3.1.7 day_in_year_number

A day_in_year_number is the position of the specified day in a year.

EXAMPLE - The 27th day of March is day 86 in years that are not leap years and day 87 in leap years.

3.1.8 month_in_year_number

A month_in_year_number is the position of the specified month in a year as defined in ISO 8601 (clause 5.2.1).

NOTE - January is month number 1, February is month number 2, March is month number 3, April is month number 4, May is month number 5, June is month number 6, July is month number 7, August is month number 8, September is month number 9, October is month number 10, November is month number 11, and December is month number 12.

3.1.9 week_in_year_number

A week_in_year_number is the value of the calendar week as defined in ISO 8601 (clause 3.1.7).

Formal propositions:

WR1: the value of the integer shall be between 1 and 53.

3.1.10 hour_in_day

A hour_in_day is the hour element of a specified time on a 24 hour clock.

EXAMPLE - The hour of 3 o'clock in the afternoon is 15.

Formal propositions:

WR1: the value of the integer shall be between 0 and 23.

3.1.11 minute_in_hour

A minute_in_hour is the minute element of a specified time.

Formal propositions:

WR1: the value of the integer shall be between 0 and 59.

3.1.12 second_in_minute

A second_in_minute is the second element of a specified time.

Formal propositions:

WR1: the value of the integer shall be between 0 and 59.

3.1.13 ahead_or_behind

An ahead_or_behind type is used to specify whether a given time is ahead of or behind coordinated universal time.

3.1.14 date_time_select

A time_date_select type allows a date and/or local_time to be referenced.

3.1.15 person_organization_select

The person_organization_select type allows a person and/or organization to be referenced.

3.1.16 support_resource_select

The support_resource_select type allows a piece of equipment, person and/or organization to be referenced with respect to their supporting role in an action_execution.

The support_resource_select recommends or requires the facilitating design, production, training, operation, and/or maintenance of a product_version. A support_resource may be personnel, support_equipment, or organization.

3.1.17 unit

A unit is a physical quantity, with a value of one, which is used as a standard in terms of which other quantities are expressed.

3.1.18 action

An action is the specific effort to realize a specific result. An action is a type of product.

— method.

3.1.18.1 method

The method is the procedure used to carry out the action.

3.1.19 action_assignment

An action_assignment is an association of an action with product data.

— assigned_action.

3.1.19.1 assigned_action

The assigned_action is the action which is to be associated with the product data.

3.1.20 action_execution

An action_execution is an action which has been carried out.

— order.

3.1.20.1 order

An order is the action_order against which the action_execution was made.

3.1.21 action_execution_support_resource

The action_execution_support_resource is the actual support_resources used/consumed in each execution of an action.

— executed_action;

— supporting_resource;

3.1.21.1 executed_action

The executed_action is the execution of an action that is performed by a support_resource.

3.1.21.2 supporting_resource

The supporting_resource is the support resource (person or organization) that is executing the action.

3.1.22 action_item

An action_item is the association of an action to a product_version.

— items.

3.1.22.1 items

Items are a set of product_versions which are associated to particular actions that are or are to be carried out.

3.1.23 action_method

An action_method is a potential means of satisfying the requirements that are highlighted in a requested_action.

— consequence;

— purpose;

— requests.

3.1.23.1 consequence

A consequence is an informal description of the effects of the action_method.

3.1.23.2 purpose

The purpose is an informal description of the rationale behind the action_method.

3.1.23.3 requests

The requests is requested_actions which could be satisfied by this action_method.

3.1.24 action_status

An action_status is the ranking which gives an indication of the state of an action.

> EXAMPLE - Effectivity from a particular date or across specific batches are examples of action_statuses.

— status;

— assigned_action.

3.1.24.1 status

The status of the action in terms of what state the action is in.

3.1.24.2 assigned_action

The assigned_action is the action_execution that has an assigned status..

3.1.25 address

An address is the place where people and organizations may be reached.

— mail_stop;

— postal_box;

— street;

— street_number;

— town;

— region;

— postal_code;

— country;

— facsimile_number;

— telephone_number;

— electronic_mail_address;

— telex_number.

3.1.25.1 mail_stop

The mail_stop is an organization defined address for internal mail delivery.

3.1.25.2 postal_box

The postal_box: is the number of a postal box.

3.1.25.3 street

The street is the name of a street.

3.1.25.4 street_number

The street_number is the number of a building on a street.

3.1.25.5 town

The town is the name of a town.

3.1.25.6 region

The region is the name of a region.

EXAMPLE - The counties of Great Britain and the states of North America are examples of regions.

3.1.25.7 postal_code

The postal_code is the code that is used by the country's postal service.

3.1.25.8 country

The country is the name of a country.

3.1.25.9 facsimile_number

The facsimile_number is the number at which facsimiles may be received.

3.1.25.10 telephone_number

The telephone_number is the number at which telephone calls may be received.

3.1.25.11 electronic_mail_address

The electronic_mail_address is the electronic address at which electronic mail may be received.

3.1.25.12 telex_number

The telex_number is the number at which telex calls may be received.

Formal propositions:

WR1: at least one of the attributes shall have a value.

3.1.26 approval

An approval is a confirmation of the quality of the product data which it is related to.

— status;

— level.

EXAMPLE - One possible level of approval is "released for production"; this explicitly identifies the approved usage. Another possible level is "preliminary design completed"; this only implies the approved usage which will depend upon company—specific procedures.

3.1.26.1 status

The status of the approval in terms of whether or not that approval has been granted.

3.1.26.2 level

The level is the type or level of approval in terms of the usage that the approval is for. This usage may be implied rather than explicit.

3.1.27 approval_assignment

An approval_assignment is an association of a approval with product data.

— assigned_approval.

3.1.27.1 assigned_approval

The assigned_approval is the approval which is to be associated with the product data.

3.1.28 approval_date_time

A approval_date_time is the association of a date and/or time with an approval.

— date_time;

— dated_approval.

3.1.28.1 date_time

The date and/or time which is to be associated with the approval.

3.1.28.2 dated_approval

The approval which is to be associated with the date and/or time.

3.1.29 approval_person_organization

A approval_person_organization is an association of a person and/or organization with an approval.

- person_organization;
- authorized_approval.

3.1.29.1 person_organization

The person_organization is the person and/or organization which authorizes the approval.

3.1.29.2 authorized_approval

The authorized_approval is the approval which is authorized by the person and/or organization.

3.1.30 approval_role

An approval_role is is a function performed with respect to an approval.

- role.

3.1.30.1 role

The role is the name of the performed function.

3.1.31 approval_status

An approval_status is is the ranking which gives an indication of the state of an approval.

EXAMLPE - 'Approved' and 'disapproved' are examples of approval_statuses.

— name.

3.1.31.1 name

The name is the ranking of the approval.

3.1.32 approved_item

A approved_item assigns an approval to a particular product_version.

— items.

3.1.32.1 items

Items are a set of approved_items which identify the versions of particular products to which the approval is assigned.

3.1.33 assembly_component_usage

The assembly component usage relates a constituent to its assembly. The assembly_component_usage entity is a subtype of the product_definition_usage entity that establishes a relationship between product_definitions within one of the following three product structures:

— bill—of—material (BOM) structure;

— parts list structure;

— promissory use structure.

The assembly_component_usage entity has four subtypes:

— The quantified_assembly_component_usage;

— The next_assembly_usage_occurrence;

— The specified_higher_usage_occurrence;

— promissory_usage_occurrence.

The quantified_assembly_component_usage represents the relationship between a constituent and an assembly where, for discrete constituents, several occurrences of the constituent are represented by the single constituent and a quantity representing the number of occurrences of it. The quantity represents a unit of measure other than a unitless number for non—discrete constituents. The next_assembly_usage_occurrence represents a relationship between a component and its immediate assembly in a product structure. The specified_higher_usage_occurrence shall be used to represent the explicit relationship between a descendent component and any ancestor higher level assembly. The promissory_usage_occurrence shall be used to represent intended relationships between a lower—level constituent and a higher level assembly, when intermediate constituents and their relationships are yet undetermined.

In a BOM graph structure, product_definition entities represent nodes and next_assembly_usage_occurrence or quantified_assembly_component_usage entities represent links.

In a parts list tree structure, a product_definition entity represents the root node. Next_assembly_usage_occurrence entities represent nodes at each intermediate level of the structure. The specified_higher_usage_occurrence entities enable links to higher levels of the structure.

In a promissory use graph structure, product_definition entities represent nodes, and promissory_usage_occurrence entities represent links between the nodes.

— reference_designator;

— product_definition_relationship.relating_product_definition;

— product_definition_relationship.related_product_definition.

3.1.33.1 reference_designator

The reference_designator is the identifier for the assembly_component_usage, in addition to the id attribute inherited from the product_definition_usage.

NOTE — The reference designator attribute may be constrained to be unique by an application protocol.

3.1.33.2 product_definition_relationship.relating_product_definition

The product_definition_relationship_relating_product_definition is an assembly for which the related_product_definition is its constituent.

3.1.33.3 product_definition_relationship.related_product_definition

The product_definition_relationship.related_product_definition is a constituent for which the relating_product_definition is its parent assembly.

3.1.34 assembly_component_usage_substitute

The assembly_component_usage_substitute specifies that one constituent can be used as a substitute for another within a given assembly context.

The instance of the substitute constituent does not require the same spatial relationship or the same quantity. A substitute constituent does not require equivalent form, fit, and function of the constituent for which it is a substitute.

This entity defines one-way substitution only. Within a given context, if A is specified as a substitute for B, B is not assumed to be a substitute for A, unless explicitly stated so in another instance of the entity.

The assembly_component_usage_substitute entity establishes an exclusive relationship between the referenced and substitute constituents.

The assembly_component_usage_substitute entity may be used to eliminate the re-identification of all higher level assemblies when a new version of a lower level constituent is created.

— base;

— substitute.

Formal Propositions:

UR1: The combination of the base and substitute attributes shall be unique.

WR1: The value of the relating_product_definition attribute of both the base and the substitute attributes shall be the same; i.e., they should refer to the same assembly product_definition.

WR2: The base and substitute attributes shall not be the same.

3.1.34.1 base

The base is an assembly_component_usage for which the substitute may be used.

3.1.34.2 substitute

The substitute is an assembly_component_usage which may be used for the base.

3.1.35 cage

The cage is a code used to uniquely identifies a commercial or government entity and/or enterprise.

— cage_code.

3.1.35.1 cage_code

The cage_code is the unique and alternate identifier of an organization.

3.1.36 calendar_date

A calendar_date is a date which is identified by a day in a month of a year.

- day_component;
- month_component.

3.1.36.1 day_component

The day_component is the day element of the date.

3.1.36.2 month_component

The month_component is the month element of the date.

3.1.37 classified_item

A classified_item applies security_classification to a particular product_version.

- items.

3.1.37.1 items

Items are a set of classified_items which identify the versions of particular products to which the security_classification_is assigned.

3.1.38 configuration_design

The configuration design relates a configuration controlled item and a product design intended to implement that item. Thus, the configuration_design entity shall represent the association of a configuration_item with a product_version to specify that the corresponding design is for the specific configuration_item.

> NOTE - organizations establish this association before any actual units are planned and before any details of the design have been established.

— configuration;

— design.

Formal propositions:

UR1: The combination of the value of the configuration attribute and the value of the design attribute shall be unique.

3.1.38.1 configuration

A configuration_item which specifies a product_version as a candidate for manufacturing actual units associated with the configuration_item.

3.1.38.2 design

A product_version representing a design which is a candidate for use in manufacturing actual units associated with the configuration attribute.

3.1.39 configuration_item

A configuration_item is used to manage the composition of constituents for actual units of manufacture.

All configuration management within an organization is done using these configuration_items.

Configuration management is the identification of a product_version that realizes the configuration_item.

The product that is planned for manufacture is referred to as the configuration_item. It is usually visible to customers of the organization that does the configuration management. A configuration_item may be an entire product_concept or some portion thereof.

A configuration_item can be established prior to the existence of a corresponding product_version.

The association between a configuration_item and a corresponding product_version is established using a configuration_design.

A configuration_item is associated with a single product_concept.

An organization determines which products are to be under its configuration management control. These products become the configuration items of the organization. These are high level functional elements which act as the focal points for managing the effectivity of constituent lower level parts and assemblies.

— item_concept;

— purpose.

Formal propositions:

UR1: The value of the identification attribute shall be unique.

3.1.39.1 item_concept

A product_concept associated with the configuration_item.

3.1.39.1 purpose

A descriptive label providing a reason to create the item_concept.

3.1.40 coordinated_universal_time_offset

A coordinated_universal_time_offset is used to relate a time to coordinated universal time by an offset (specified in hours and minutes) and a direction.

— hour_offset;

— minute_offset;

— sense.

3.1.40.1 hour_offset

The number of hours by which a time is offset from coordinated universal time.

3.1.40.2 minute_offset

The number of minutes by which a time is offset from coordinated universal time.

3.1.40.3 sense

The direction of the offset.

3.1.41 correspondence

A type of document which is used for communication between parties.

3.1.42 data_template

A type of product which defines in a skeleton manner, the makeup and format of a technical report, document, input/output screen or any set of desired information.

3.1.43 date

A date is the identification of a moment in time occuring between midnight of one day and midnight of the day following.

— year_component.

3.1.43.1 year_component

The year in which the date occurs.

3.1.44 date_and_time

A date_and_time is a moment of time on a particular day.

— date_component;

— time_component.

3.1.44.1 date_component

The date element of the date time combination.

3.1.44.2 time_component

The time element of the date time combination.

3.1.45 dated_effectivity

The dated effectivity specifies that a product_definition_usage is effective for a series of actual units produced during a given time period.

— effectivity_start_date;

— effectivity_end_date.

3.1.45.1 effectivity_start_date

The date and time at which the product_definition_usage identified by the design_usage attribute becomes effective.

3.1.45.2 effectivity_end_date

The date and time at which the product_definition_usage identified by the design_usage attribute is no longer effective. If no value is given the end date for the effectivity is not yet determined.

3.1.46 derived_unit

A derived_unit is an expression of units.

— elements.

Formal propositions:

WR1: there shall be either more than one member in the elements set or the value of the exponent of the single element of the elements set shall not be equal to one.

3.1.46.1 elements

The group of units and their exponents that define the derived_unit.

3.1.47 derived_unit_element

A derived_unit_element is one of the unit quantities which makes up a derived_unit.

> EXAMPLE - Newtons per square millimetre is a derived unit. It has two elements, Newton whose exponent has a value of 1 and millimeter whose exponent is —2.

— unit;

— exponent.

3.1.47.1 unit

The fixed quantity which is used as the mathematical factor.

3.1.47.2 exponent

The power that is applied to the unit attribute.

3.1.48 dimensional_exponents

The dimensionality of any quantity can be expressed as a product of powers of the dimensions of base quantities. The dimensional_exponents entity defines defines the powers of the dimensions of the base quantities. All the physical quantities are founded on seven base quantities.

> NOTE - Length, mass, time, electric current, thermodynamic temperature, amount of substance, and luminous intensity are the seven base quantities.

EXAMPLE - A length of 2 millimetres has a length exponent of 1. The remaining exponents are equal to 0. A velocity of 2 millimetres per second has a length exponent of 1 and a time exponent of —1. The remaining exponents are equal to 0.

— length_exponent;

— mass_exponent;

— time_exponent;

— electric_current_exponent;

— thermodynamic_temperature_exponent;

— amount_of_substance_exponent;

— luminous_intensity_exponent.

3.1.48.1 length_exponent

The power of the length base quantity.

3.1.48.2 mass_exponent

The power of the mass base quantity.

3.1.48.3 time_exponent

The power of the time base quantity.

3.1.48.4 electric_current_exponent

The power of the electric current base quantity.

3.1.48.5 thermodynamic_temperature_exponent

The power of the thermodynamic temperature base quantity.

3.1.48.6 amount_of_substance_exponent

The power of the amount of substance base quantity.

3.1.48.7 luminous_intensity_exponent

The power of the luminous intensity base quantity.

3.1.49 discrepant_product

Identifies a product_version that fails to satisfy design nominal criteria.

— failure_rate.

3.1.49.1 failure_rate

The failure_rate is the number of failures a product has failed to operate correctly.

3.1.50 document

A document is an unambiguous reference to a formal standard or document. A document is a type of product.

- kind;
- size.

3.1.50.1 kind

The sort of data that the document describes.

3.1.50.2 size

The size is the relative measurement of the document.

EXAMPLE - 12,345 bytes or 8 pages are examples of a size of a document.

3.1.51 document_reference

A document_reference is an association of a document with product data.

— assigned_document.

3.1.51.1 assigned_document

The document which is to be associated with the product data.

3.1.52 drawing

A drawing is a type of document which graphical represents a product.

3.1.53 document_type

A document_type is the sort of data that the formal standards or documents are being used to describe in a particular context..

EXAMPLE - 'Material', surface finish', and 'heat treatment process' are all pieces of data that can be described implicitly, by reference to other documents (such as DIN documents), rather than explicitly every time they are used.

— product_data_type.

3.1.53.1 product_data_type

The product_data_type is the name of the sort of data that the document is being used to describe.

3.1.54 enhancement_product

An enhancement_product is the identification of a need for new or improved product functionality.

3.1.55 enterprise

A type of organization that identifies a supplier/manufacturer/consumer of a product_version (in-house or external).

3.1.56 file_folder

The association of a product_version to a file or folder.

— representative_product;

— file_type.

3.1.56.1 representative_product

The representative_product is the product which is represented by the file or folder.

3.1.56.2 file_type

The file_type defines whether the file_folder instance is a file or a folder.

3.1.57 group

A group is a collection of element.

- group_name.

3.1.57.1 group_name

The word, or group of words, by which the group is referred to.

3.1.58 hardware_software

A hardware_software is a type of system. It defines a physical implementation of a computer system architecture.

3.1.59 local_time

A local_time is a moment of occurence measured by hour, minute, and second. It represents one instant of time on a 24 hour clock.

> NOTE - This construct is used to represent a moment in time whereas time measures represent amounts of time.

> EXAMPLE - 1500 hours is an instant in time whereas 15 hours is an amount of time.

- hour_component;

- minute_component;

- second_component.

3.1.59.1 hour_component

The hour_component is the number of hours.

3.1.59.2 minute_component

The minute_component is the number of minutes.

3.1.59.3 second_component

The second_component is the number of seconds.

Formal propositions:

WR1: the seconds attribut shall only exist if the minute attribute exists.

3.1.60 lot_effectivity

This lot effectivity specifies that a product_definition_usage is effective for a specific quantity of actual products. The product lot has an identifier. If the lot contains individual units, these need not be identified.

— effectivity_lot_id;

— effectivity_lot_size.

3.1.60.1 effectivity_lot_id

The effectivity_lot_id is an identifier for the lot of the actual product. The product_definition_usage identified by the inherited design_usage attribute is effective for this lot.

3.1.60.2 effectivity_lot_size

The effectivity_lot_size is a measure of the size of the effective lot.

3.1.61 make_from_usage_option

The make from usage option identifies that a product is made from another product through machining or some other unspecified process.

In situations in which a product is made from another product using a sequence of processes, the intermediate products will be related using the make_from_usage_option entity.

A product to be modified can be an assembly.

> NOTE 1 - Generally, the assembly_component_usage differs from the make_from_usage_option in that the constituents of an assembly are used in the assembly without any change.

The make_from_usage_option represents the fact that any actual unit of one design can be manufactured by consuming or modifying an actual unit of another design;

NOTE 2 - Typically the consumed product is referred to as stock or raw material.

The make_from_usage_option_group is used to represent one specific combination of products that can be made from a single product;

NOTE 3 - Typically the single product is referred to as stock or raw material.

The relationship concept represented by the make_from_usage_option applies to designs, represented by product_definitions, rather than the actual units of the designs. A make_from_usage_option relationship is independent of any specific manufactured instances of actual units, and is represented by the attribute references, inherited from the supertype entity, to the relating_product_definition and related_product_definition.

A product_definition may be the relating_product_definition of many make_from_usage_option relationships, and a product_definition may be the related_product_definition of many make_from_usage_option relationships. Further, there may be multiple make_from_usage_option instances referencing the same relating_product_definition and related_product_definition pair of product_definitions.

EXAMPLE 6 - Consider the case of a shaft which can be machined from either a casting or a forging. All three, the shaft, the forging and the casting, are represented by separate instances of product_definitions. Two instances of the make_from_usage_option entity exist, one between the relating_product_definition shaft and the related_product_definition forging, the other between the relating_product_definition shaft and the related_product_definition casting.

— ranking;

— ranking_rationale;

— quantity;

— product_definition_relationship.relating_product_definition;

— product_definition_relationship.related_product_definition.

Formal propositions:

WR1: The value of ranking shall be positive.

WR2: The value of quantity shall be positive.

3.1.61.1 ranking

The ranking is an integer which ranks the preference for use of the related_product_definition input product_definition among all make_from_usage_option instances with the same value for the inherited relating_product_definition attribute. This is a positive integer value that only has meaning when comparing it with corresponding values for make_from_usage_options sharing the same relating_product_definition product_definition. It is a relative ranking value, not an absolute ranking. A lower value indicates a higher preference for the related_product_definition product_definition, and a higher value indicates a lower preference.

> NOTE - Special care is required when assigning these values. If different organizations use different ranges of values, and if populated data sets from these organizations are merged, and multiple make_from_usage_— options from both organizations then exist in the merged file for a single relating_product_definition product_definition, then non—comparable values for this attribute may result.

3.1.61.2 ranking_rationale

The ranking_rationale is the text which describes the rationale used for the ranking.

> EXAMPLE 7 - Examples of ranking_rationale are cost and long lead time.

3.1.61.3 quantity

The quantity is the number of physical instances of the relating_product_definition product_definition that can be made from one unit of a related_product_definition product_definition.

3.1.61.4 product_definition_relationship.relating_product_definition

A product_definition_relationship.relating_product_definition is a product_definition made from the related_product_definition product.

3.1.61.5 product_definition_relationship.related_product_definition

A product_definition_relationship.related_product_definition is a product_definition from which the relating_product_definition is made.

3.1.62 measure_with_unit

A measure_with_unit is the specification of a physical quantity.

— value_component;

— unit_component.

3.1.62.1 value_component

The value of the physical quantity when expressed in the specified units.

3.1.62.2 unit_component

The unit in which the physical quantity is expressed.

Formal propositions:

WR1: the unit shall be a valid unit for the kind of measure.

3.1.63 named_unit

A named_unit is a unit quantity associated with the word, or group of words, by which the unit is identified.

— dimensions.

3.1.63.1 dimensions

dimensions: the exponents of the base properties by which the named_unit is defined.

3.1.64 next_assembly_usage_occurrence

The next_assembly_usage_occurrence is the relationship between a child constituent and its immediate parent assembly in a product structure. It represents the use of individual occurrences of constituents. The use of the same constituent may be represented by another distinct next_assembly_usage_occurrence instance for the purpose of assigning a position and orientation for the constituent.

> NOTE - An application algorithm can derive an indented parts list for a product by sequentially tracing through a structure of next_assembly_usage_occurrence instances. A similar algorithm can be used to calculate the position and orientation of each occurrence of

every constituent relative to its higher level assemblies within a BOM.

— product_definition_relationship.relating_product_definition;

— product_definition_relationship.related_product_definition.

3.1.64.1 product_definition_relationship.relating_product_definition

The product_definition_relationship.relating_product_definition is an assembly for which the related_product_definition is its immediate constituent.

3.1.64.2 product_definition_relationship.related_product_definition

The product_definition_relationship.related_product_definition is a constituent for which the relating_product_definition is its immediate parent assembly.

3.1.65 node_location

A node_location is the network node where a hardware_software system resides.

— system;

— protocol;

— node_address.

3.1.65.1 system

The system is the hardware_software system that reside at the location.

3.1.65.2 protocol

The protocol is convention used to define the address of the hardware_software system.

3.1.65.3 node_address

The node_address is the addres location of the hardware_software system.

3.1.66 ordered_action

An ordered_action is the formal notification that authority has been given to perform an action. An action_order is the result of the processing of requested_actions.

> NOTE - The distinction between a requested_action and an ordered_action is the level of authority that is associated with it. Anyone can submit a requested_action whereas only authorized people or organizations can submit ordered_actions that are to be acted upon. A request asks for action whereas an order demands action.

— name;

— description;

— analysis;

— comment;

— requests.

3.1.66.1 name

A name is the word, or group of words, by which the ordered_action is referred to.

3.1.66.2 description

The description is the text that relates the nature of the ordered_action.

3.1.66.3 analysis

The analysis is an informal description of the results of the analysis that was carried out on the elements of the requests set.

> EXAMPLE - The fact that two different requests are asking for the same effect could be recorded in this attribute.

3.1.66.4 comment

The comment is an informal description of any other pertinent information.

3.1.66.5 requests

The requests are the requested_action that this ordered_action relates to.

3.1.67 ordinal_date

An ordinal_date is a date which is identified by a day of a year.

— day_component.

Formal propositions:

WR1: the day_component shall be between 1 and 365 if the year_component is not a leap year; otherwise the day_component shall be between 1 and 366.

3.1.67.1 day_component

The day_component is the day element of the date.

3.1.68 organization

An organization is an administrative structure.

— cage_code.

3.1.68.1 cage_code

The cage_code is the unique and alternate identifier of an organization.

3.1.69 organizational_address

A organizational_address is an address where organizations are located.

- — organizations.

3.1.69.1 organizations

The organizations are the organizations located at the address.

3.1.70 organizational_project

An organizational_project is project for which one or more organizations are responsible.

- — name;
- — description;

- — responsible_organization.

3.1.70.1 name

The name is the word, or group of words, by which the organizational_project is referred to.

3.1.70.2 description

The description is the text that relates the nature of the organizational_project.

3.1.70.3 responsible_organization

The responsible_organization is the organizations which are responsible for the project.

3.1.71 part

A part is a product that is intended to be produced or employed in a production process. A part is the type of product that is a discrete product of the organization.

- — part_type;
- — part_function_type;
- — part_configuration_identifier.

3.1.71.1 part_type

The part_type is the further classification of a part.

3.1.71.2 part_function_type

The part_fucntion_type is the further functional classification of a part.

3.1.71.3 part_configuration_identifier

The part_configuration_identifier is the identification of the configuration of the part.

3.1.72 person

A person is an individual human being.

— last_name;

— first_name;

— middle_name;

— prefix_titles;

— suffix_titles.

Formal propositions:

WR1: either the last_name or the first_name shall be defined.

3.1.72.1 last_name

The last_name is the person's surname.

3.1.72.2 first_name

The first_name is the first element of the person's list of forenames.

3.1.72.3 middle_name

The middle_name is the person's other forenames, if there are any.

3.1.72.4 prefix_titles

The prefix_titles is the word, or group of words, which specify the person's social and/or professional standing and appear before his/her names.

3.1.72.5 suffix_titles

The suffix_titles is the word, or group of words, which specify the person's social and/or professional standing and appear after his/her names.

3.1.73 person_and_organization

A person_and_organization is a person in an organization.

— the_person;

— the_organization.

3.1.73.1 the_person

The the_person is the person who is related to the organization.

3.1.73.2 the_organization

The the_organization is the organization to which the person is related.

3.1.74 personal_address

A personal_address is an address where a person resides.

— people.

3.1.74.1 people

The people are the people who reside at the address.

3.1.75 physical_unit

A uniquely identifiable physical manifestation of a product_version design. A tracked instance of a product_version (that is, a serialized unit or lot).

— configuration.

3.1.75.1 configuration

The configuration is the configuration_design which is is associated to a physical instantiation of a product_version.

3.1.76 planned_effectivity

The planned effectivity defines common effectivity attributes for items under configuration control. The planned_effectivity entity is used by an organization to specify effectivity of product_definition_usages.

> EXAMPLE 13 - A user may want to specify that certain product_definition_usages are to be effective for a configuration_item. A 200 HP engine is to be effective starting on a certain date. This information is captured prior to any production plans exist for the 200 HP engine in a planned_effectivity entity.

Configuration management is the association of the appropriate versions of a product to build a configuration_item. This association is referred to as planned_effectivity.

There are three ways to apply planned_effectivity. They are:

a) serial_numbered_effectivity, where the planned_effectivity is based on serial numbered instances of manufactured products.

b) dated_effectivity, where the planned_effectivity is based on dates when instances of the product are manufactured.

c) lot_effectivity, where the planned_effectivity is based on instances of lots of products manufactured.

The subtypes of this entity represent different situations in which the specified design_usage is effective for actual units of a configuration_item.

- configuration;
- design_usage;
- identification.

Formal propositions:

UR1: The combination of the value of the configuration attribute, the value of the design_usage attribute,and the value of the identification attribute shall be unique.

WR1: The design_usage shall refer to a constituent of the product_version referenced by the configuration_design.

3.1.76.1 configuration

The configuration is a configuration_design whose product_version is contained in the set of product_definition_usages that constitute the configuration_item of the configuration_design.

3.1.76.2 design_usage

A design_usage is a product_definition_usage instance which the planned_effectivity entity specifies as being effective.

3.1.76.3 identification

The identification is an identifier for the planned_effectivity.

3.1.77 procedure

A procedure is a type of document that describes procedures to be followed.

3.1.78 product

A product is a physically realizable object that is produced by a natural process or manufacture.

> EXAMPLE - Production, construction, manufacture, and fabrication are all examples of processes.

> EXAMPLE - The ball-point pen, its cap, and the assembly of the cap and the ball-point pen are all physically realizable objects.

- — id;
- — name;
- — description;
- — frame_of_reference.

Formal propositions:

UR1: every product's identification shall be unique.

3.1.78.1 id

The id is the identification of the product.

> EXAMPLE — Part numbers and stock item numbers are examples of product identifiers.

3.1.78.2 name

The name is the word, or group of words, by which the product is referred to.

EXAMPLE — ``Ball—point pen", ``cap", and ``nib" are examples of product.names.

3.1.78.3 description

The description is the text that relates the nature of the product.

3.1.78.4 frame_of_reference

The frame_of_reference is the context within which the product was defined.

3.1.79 product_anomaly

The product_anomaly is the identification of a nonconformance or a deviation from design nominal conditions for a product.

— anomaly_cause;

— anomaly_type;

— detection_method;

— product_anomaly_description;

— product_anomaly_identifier.

3.1.79.1 anomaly_cause

An anomaly_cause specifies a narrative identifying the reason why the nonconformance occurred.

3.1.79.2 anomaly_type

An anomaly_type specifies the type of product_anomaly as being either an product_issue, product_concern, or an product_flaw.

3.1.79.3 detection_method

The detection_method specifies the procedure that a system, sub-system or assembly was evaluated and determined to be nonconforming.

3.1.79.4 product_anomaly_description

The description specifies a narrative account describing the nonconformance.

3.1.79.5 product_anomaly_identifier

An identifier specifies the unique identification of a product_issue, product_concern, or a product_flaw that is associated with a product.

3.1.80 product_anomaly_disposition

The product_anomaly_disposition is the actual resolution applied to a product_anomaly.

— anomalized_product;

— disposition_actions.

3.1.80.1 anomalized_product

The anomalized_product specifies the identification of a product_anomaly.

3.1.80.2 disposition_actions

The disposition_actions specifies the performance of an action_execution for answering the disposition of a product to the satisfaction of the controlling interest.

3.1.81 product_change

An product_change is the creation of a new product that results from an anomaly or concern about a baseline product.

> NOTE - This entity identifies the new product as well as the baseline product that the new version was based upon, due to an anomaly or concern as well as the authorization that accounts for the product_change.

— baseline_product;

— baseline_product_disposition;

— reasons;

— resulting_product.

3.1.81.1 baseline_product

The baseline_product specifies the product that undergoes a change process and results in a new product.

3.1.81.2 baseline_product_disposition

The baseline_product_disposition specifies the resolution that is being applied to a baseline_product to satisfy an anomaly.

3.1.81.3 reasons

The reason specifies the rationale of why a product_change took place.

3.1.81.4 resulting_product

The resulting_product specifies the product that results from a change process.

3.1.82 product_classification

A product_classification is a type of classified item that associates a security_classification with a product.

— items.

3.1.82.1 items

The items is the product which is assigned to a security_classification.

3.1.83 product_concept

The product_concept is the idea of a product as defined by customer needs. The product_concept and its features may be identified as configuration items to control their manufacture. A product concept may exist before a product has been defined. A product concept identifies a selection of product features or capabilities.

A product concept identifies a deliverable product as perceived by the customer. A product concept is often used to identify a selection of product features or capabilities.

A product concept may be composed of several configuration items.

> Note - A product_concept will often correspond to the highest level item(s) manufactured by an organization for a customer. It may be characterized by a set of product features identified by the customers or derived from customers' needs. The definition of product concepts is often driven by marketing.

> EXAMPLE - If an organization manufactures cars and engines for cars, the cars will be represented by product_concept instances. If another organization manufactures engines for cars, then the engines will be represented as product_concept in that organization.

— product_concept_context.

3.1.83.1 product_concept_context

The product_concept_context is a market context in which the product_concept is defined.

3.1.84 product_concern

The product_concern is a type of product_anomaly that expresses a concern for a particular product.

3.1.85 product_definition

A product definition is the identification of a characterization of a product_version in a particular application context.

> NOTE - A product_definition is characterized by properties which refer to it.

> EXAMPLE - A product's physical design may be one product_definition whilst the functional design of the same product may be a different product_definition. Both product_definitions would be related to the same product_version but would be used in different application contexts.

— description;

— version;

— frame_of_reference.

3.1.85.1 description

The description is the text that relates the nature of the product_definition.

3.1.85.2 version

The version is the product_version to which the product_definition relates.

3.1.85.3 frame_of_reference

The frame_of_reference is the product_definition_context in which the product_definition or product_definition data is used.

3.1.86 product_definition_relationship

A product_definition_relationship is an association between two product_definitions. An association may exist between product_definitions that relate to different products or between different definitions of the same product.

> EXAMPLE - The relationships within a bill of materials structure are examples of product_definition_relationships that associate different products. The relationship between a sketch and a detailed design is an example of a product_definitionrelationship that associates different definitions of a single product.

A single product_definition may be used more than once within the description of a product.

> NOTE - The same component could be used more than once in the same assembly. Each usage of the component would be specified as an instance of the product_definition_relationship entity.

— id;

— name;

— description;

— relating_product_definition;

— related_product_definition.

3.1.86.1 id

The id is the identification of the product_definition_relationship .

3.1.86.2 name

The name is the word, or group of words, by which the product_definition_relationship is referref by.

3.1.86.3 description

The description is the text that relates the nature of the product_definition_relationship.

3.1.86.4 relating_product_definition

The relating_product_definition is one of the product_definitions which is a part of the relationship.

> EXAMPLE - If the product_definition_relationship is an assembly component relationship the relating_product_definition may be the assembly.

3.1.86.5 related_product_definition

The related_product_definition is the other product_definition which is a part of the relationship.

EXAMPLE - In an assembly the related_product_definition may be the product_definition that is an element of the assembly.

3.1.87 product_definition_usage

The product_definition_usage is a subtype of the product_definition_relationship entity for use within the context of product structure definition and management. This subtype adds meaning to the two attributes: relating_product_definition, related_product_definition.

The subtypes of this entity represent different kinds of product structure relationships between the referenced pair of product_definitions. One subtype, make_from_usage_option, represents the relationship between a product and another product, where one product is made from the other. The other subtype, assembly_component_usage, represents the relationship between an assembly and one of its constituents.

— product_definition_relationship.id;

— product_definition_relationship.relating_product_definition;

— product_definition_relationship.related_product_definition.

Formal propositions:

UR1: The inherited id, relating_product_definition and related_product_definition, uniquely identifies an instance of product_definition_usage.

WR1: The graph structure of product_definition nodes and product_definition_usage links shall be acyclic. Each product_definition shall not be a descendant of itself in the graph structure.

3.1.87.1 product_definition_relationship.id

The product_definition_relationship.id is an identifier for a usage of a product_definition. It is used to distinguish between two instances of product_definition_usage where the pair of product_definition attributes are the same

> EXAMPLE 5 - If four identical bolts are used to attach two plates, there may be a need to identify one specific bolt for some purpose. It needs to be torqued to a greater degree than the rest. The id attribute then is used to identify this specific bolt's requirement, even though all four bolt product_definition_usages will have the same attribute pair of product_definitions.

3.1.87.2 product_definition_relationship.relating_product_definition

The product_definition_relationship.relating_product_definition is a product_definition that is made from or serves as the assembly for the related_product_definition.

3.1.87.3 product_definition_relationship.related_product_definition

The product_definition_relationship.related_product_definition is a product_definition from which the relating_product_definition is made or which is the component in the relating_product_definition assembly.

3.1.88 product_flaw

The description of a nonconformance or flaw in, on or about a product_version.

— product_flaw_type.

3.1.88.1 product_flaw_type

The product_flaw_type is the further classification of the type of flaw that is associated to a product.

3.1.89 product_issue

The identification of special issues or concerns that are not flaws but may require further action.

3.1.90 product_process_step

The product_process_step is a type of action_assignment where an action is associated to a product.

— products.

3.1.90.1 products

The product which is to be associated with the process_step.

3.1.91 product_requiring_change

Identifies a product_version that does not satisfy a particular requirement. A product_requiring_change is a product_version that is changed because of the identification of a flaw or need for capability enhancement .

— requiring_change_product;

— anomalized_products;

— product_change_requirement_type.

3.1.91.1 requiring_change_product

The requiring_change_product specifies the action_execution that will satisfy the change requirement.

3.1.91.2 anomalized_products

The anomalized_products specifies the product_anomaly that that wil be addressed by the change.

3.1.91.3 product_change_requirement_type

The product_change_requirement_type specifies whether the reason for a product change is either a discrepancy or enhancement.

3.1.92 product_responsibility

The product_responsibility specifies the association of a organizational_project to a product.

— project;

— product.

3.1.92.1 project

The project is the organizational_project that is associated to the product.

3.1.92.2 product

The product specifies the product that is associated to an organizational_project.

3.1.93 product_state

The product_state specifies the lifecycle state of a product.

- state_name;
- product;
- action_transition.

3.1.93.1 state_name

The state_name is the word, or group of words, by which the product_state is referred to.

3.1.93.2 product

The product specifies the product_version that has the associated lifecycle state.

3.1.93.3 action_transition

The action_transition specifies the action_execution that transitioned the product_version to a given lifecycle state.

3.1.94 product_version

A product_version is an identified version of a product that differs from other versions in some significant way. However, it is insufficiently different to be regarded as a different product.

> NOTE - At any given time there may be multiple active and obsolete versions for the same product.

— version_id;

— description;

— of_product.

Formal propositions:

UR1: the version_id of each product_version that is related a single product (through their ofproduct attributes) shall be unique within the collection of product_versions which are related to that product.

3.1.94.1 version_id

The version_id is the unique identification of the product_version in the context of the product that it relates to.

> EXAMPLE — Part version number is an example of a product_version identifier.

3.1.94.2 description

The description is the text the relates the nature of the product_version.

NOTE - The descriptions of different versions of a single product could identify differences in the purpose and function of each version.

3.1.94.3 of_product

The of_product is the product that the product_version is a version of.

NOTE - A product is associated with one or more product_versions through the inverse of this relationship.

3.1.95 product_version_group

A product_version_group is the association of a product_version to a group.

— group;

— version.

3.1.95.1 group

The group is the group in which a product belongs to.

3.1.95.2 version

The version is the product_version which belongs to the group.

3.1.96 program

A type of organization denoting a particular organized thrust or development effort.

EXAMPLE - The B-1B Aircraft Program is an example of a program.

3.1.97 promissory_usage_occurrence

The promissory usage occurrence is the intention to use constituent product_definition in an assembly product_definition. It is used when the product structure is not completely defined. In such a situation, it is still possible to relate an assembly to a constituent to capture the intent that the constituent will be eventually used. The promissory_usage_occurrence represents the relationship between a constituent and an ancestor assembly within an overall product structure without any specification of the intermediate assemblies being represented.

— product_definition_relationship.relating_product_definition;

— product_definition_relationship.related_product_definition.

3.1.97.1 product_definition_relationship.relating_product_definition

The product_definition_relationship.relating_product_definition is an assembly for which the related_product_definition is a constituent, and the details of the product structure are not completely defined.

3.1.97.2 product_definition_relationship.related_product_definition

The product_definition_relationship.related_product_definition is a constituent for which the relating_product_definition is an assembly, and the details of the product structure are not completely defined.

3.1.98 publication

A publication is a type of document that published for distribution.

3.1.99 quantified_assembly_component_usage

The quantified_assembly_component_usage establishes the relationship between an assembly and one of its constituents, when there is a need to specify the quantity of the child constituent used in the assembly.

> NOTE - Generally for production planning or material planning purposes several occurrences of a constituent are lumped together and a quantity is specified to account for the several occurrences. A typical example would be the specifying of an occurrence of a rivet used for joining airplane structures and denoting the number of such rivets used on the entire plane. If each of the occurrences of the rivets used is to be specified, then the next_assembly_usage_occurrence entity may be used. As many instances of the next_assembly_usage_occurrence as the number of occurrences of the rivets will exist.

— quantity;

— product_definition_relationship.relating_product_definition;

— product_definition_relationship.related_product_definition.

3.1.99.1 quantity

The quantity is a measure of how many or how much of the constituent is used in the assembly.

3.1.99.2 product_definition_relationship.relating_product_definition

The product_definition_relationship.relating_product_definition is an assembly for which the related_product_definition is its constituent, and where the quantity of the constituent may be specified.

3.1.99.3 product_definition_relationship.related_product_definition

The product_definition_relationship.related_product_definition is an assembly for which the relating_product_definition is its parent assembly, and where the quantity of the constituent may be specified.

3.1.100 recommended_support_resource

A support_resource that is recommended/required to assist, accomodate/facilitate, the performance of an action_item such as design, production, training, operation, and/or maintenance.

— recommended_action;

— supporting_resource.

3.1.100.1 recommended_action

The recommended_action is the recommendation of an action to be performed on a product_version by a support_resource.

3.1.100.2 supporting_resource

The supporting_resource is the support resource (person or organization) that is recommended to perform the action.

3.1.101 related_change

A related_change is a type of product_requiring_change that identifies a product_requiring_change due to an anomaly with another product_requiring_change .

— anomalized_product;

— related_change_product.

3.1.101.1 anomalized_product

An anomalized_product specifies the identification of a product anomaly that has identified an additional product_requiring_change.

3.1.101.2 related_change_product

The related_change_product is a product that has been identified as needing to be changed due to the change of another product.

3.1.102 requested_action

A requested_action is a formal notification of a desire for action to be taken.

— id;

— version;

— purpose;

— description.

3.1.102.1 id

The id is the means of identification of the requested_action.

3.1.102.2 version

The version is the identification of the version of the requested_action.

3.1.102.3 purpose

The purpose is an informal description of the reason for the requested_action.

3.1.102.4 description

The description is an informal definition of the requested_action.

3.1.103 reuse_part

A is a type of part and it is a Martin Marietta part that may be reused for different signal processor designs.

3.1.104 role

A role is the context in which a user will perform a given action on a product.

— role_name.

3.1.104.1 role_name

The role_name is the nomenclature used to describe the role that a user plays in the performance of an action. A role_name may be considered a user job classification.

EXAMPLE - Examples of role_names are "designer", "manager", and "checker".

3.1.105 security_classification

A security classification is the level of confidentiality that is required for the purpose of product data protection.

— name;

— purpose;

— security_level.

3.1.105.1 name

The name is the word, or group of words, by which the security classification is referred to.

3.1.105.2 purpose

The purpose is an informal description of the intent of the security_classification.

3.1.105.3 security_level

The security_level is the category of the security_classification.

3.1.106 security_classification_assignment

A security_classification_assignment is an associaton of a security_classification with product data.

- — assigned_security_classification.

3.1.106.1 assigned_security_classification

The assigned_security_classification is the security_classification which is to be associated with the product data.

3.1.107 security_classification_level

A security_classification_level is a category of security.

EXAMPLE - 'Confidential', 'secret', 'and top secret' are examples of security_classification_levels.

— name.

3.1.107.1 name

The name is the word, or group of words, by which the security_classification_level is spoken of or referred to.

3.1.108 serial_numbered_effectivity

This serial numbered effectivity specifies that a product_definition_usage is effective for one or more actual units that result from a production planning activity, where each such actual unit has its own individual identifier. These identifiers are used to define a range. It is assumed that these identifiers are assigned during actual manufacturing of a product and have a well defined ordering algorithm.

— effectivity_start_id;

— effectivity_end_id.

3.1.108.1 effectivity_start_id

The first of one or more actual units to result from a production planning activity. The product_definition_usage identified by the design_usage attribute is effective for these actual units.

3.1.108.2 effectivity_end_id

The ending identifier of a bounded sequence of actual units. If no value is given the range of the serial_numbered_effectivity is open. If the values of the effectivity_start_id and effectivity_end_id are the same, the serial_numbered_effectivity applies to a single actual unit.

3.1.109 signal_processor_design

A signal_processor_design is a type of configuration_item. It is the focus of configuration management for the signal processor design process.

3.1.110 software_application

A software_application is a type of part. It is programming code that may be idenified by a part number and is written in a specific software programming language.

3.1.110.1 software_language

The software_language is the word, or group of words, which identify the programming language which was utilized to produce the software_application.

3.1.111 specification

A specification is a type of document that defines a product in detail and/or its requirements.

3.1.112 specified_higher_usage_occurrence

The specified_higher_usage_occurrence represents the relationship between a specific use of a constituent with respect to a non-immediate/non-parent ancestor assembly within the product structure; For a general product structure, in order to identify the usage of any constituent within an assembled product, it is necessary to identify the path between the assembled product and the constituent. The specified_higher_usage_occurrence entity provides this capability.

The specified_higher_usage_occurrence specifies the relationship between a constituent and an assembly where the assembly is not the immediate parent for the constituent.

If a specified_higher_usage_occurrence is specified, the entire path between the constituent and the assembly is also identified using successive instances of specified_higher_usage_occurrence. Successive instances of specified_higher_usage_occurrence identify all the intermediate constituent and assembly relationships that exist between the assembly and its constituent specified by the primary specified_higher_usage_occurrence.

The relationship between the constituent and the assembly of the specified_higher_usage_ occurrence to be specified is captured by the relationship of the inherited attributes relating_product_definition and related_product_definition.

The two attributes (upper_usage and next_usage) within the primary instance of the entity specified_higher_usage_occurrence will respectively specify the next_assembly_usage_occurrence and an assembly_component_usage which together will provide the definition of the path from the constituent to the assembly for which the specified_higher_usage_occurrence is being specified. To ensure that the next_assembly_usage_occurrence and the assembly_component_usage together constitute the entire path desired for the specified_higher_usage_occurrence, it is essential that the instance of the related_product_definition attribute of the assembly_component_usage entity referenced by the upper_usage be the same as the instance of the relating_product_definition attribute of the next_assembly_usage_occurrence entity referenced by next_usage. The attribute related_product_definition of the next_assembly_usage_occurrence shall be the same instance of the attribute related_product_definition of the specified_higher_usage_occurrence being specified. The attribute relating_product_definition of the assembly_component_usage entity referenced by the attribute upper_usage shall be the same instance as the attribute relating_product_definition of the specified_higher_usage_occurrence being specified.

If the assembly_component_usage referenced by the attribute upper_usage is not a next_assembly_usage_occurrence it will be a specified_higher_usage_occurrence. This specified_higher_usage_occurrence shall have its attributes upper_usage and next_usage defined as described in the previous paragraph to specify further the path of the primary specified_higher_usage_occurrence. This recursive specification shall continue until the attribute upper_usage references an assembly_component_usage entity that is a next_assembly_usage_occurrence. At this point, the primary specified_higher_usage_occurrence is fully specified both in terms of its constituents/assembly relationship and the entire path between them.

In order to be able to completely specify a specified_higher_usage_occurrence all the necessary assembly_component_usage instances shall have been defined.

The specified_higher_usage_occurrence entity supports the representation of parts list tree structures. Typically, it is used to define portions of parts lists that contain a specific constituent within an assembly for which certain properties are to be associated.

— upper_usage;

— next_usage;

— product_definition_relationship.relating_product_definition;

— product_definition_relationship.related_product_definition.

Formal propositions:

UR1: The combination of the upper_usage and next_usage attributes shall be unique.

WR1: The instance of specified_higher_usage_occurrence shall not be the same as the instance of upper_usage.

WR2: The relating_product_definition (i.e., assembly) of the specified_higher_usage_occurrence shall be the same instance product definition as relating_product_definition (i.e., assembly) for the upper_usage.

WR3: The related_product_definition (i.e., constituent) of the specified_higher_usage_occurrence shall be the same instance of product definition as the related_product_definition for the next_usage.

WR4: The related_product_definition (i.e., component) for the upper_usage shall be the same instance of product_definition as the relating_product_definition (i.e., assembly) for the next_usage.

WR5: The type of the upper_usage attribute cannot be the promissory_usage_occurrence type.

3.1.112.1 upper_usage

The upper_usage is an assembly_component_usage that has the same instance of the attribute relating_product_definition as this specified_higher_usage_occurrence and the same instance of the attribute related_product_definition as the relating_product_definition of the next_assembly_usage_ occurrence referenced by the attribute next_usage.

3.1.112.2 next_usage

The next_usage is a next_assembly_usage_occurrence that has the same instance of the attribute related_product_definition as this specified_higher_usage_occurrence and the same instance of the product definition referenced by the attribute relating_product_definition as the product definition referenced by the attribute related_product_definition of the attribute upper_usage.

3.1.112.3 product_definition_relationship.relating_product_definition

The product_definition_relationship.relating_product_definition is the inherited attribute for the assembly product definition of the specified_higher_usage_occurrence.

3.1.112.4 product_definition_relationship.related_product_definition

The product_definition_relationship.related_product_definition is the inherited attribute for the constituent product definition of the specified_higher_usage_occurrence.

3.1.113 specified_item

A specified_item assigns a document to a particular product_version.

— items.

3.1.113.1 items

Items are a set of specified_items which identify the versions of particular products to which the document is assigned.

3.1.114 supplier

A supplier is a type of organization. In the supplier capacity, the organization provides a product, on a given date, to the enterprise.

3.1.114.1 source_product

The product is the product that is supplied.

3.1.114.2 source_date

The source_date is the date in which the product was supplied by the organization.

3.1.115 support_equipment

A device recommended/required to facilitate design, production, training, operation, and/or maintenance of a product_version.

— name.

3.1.115.1 name

The name is the word, or group of words, by which the support_equipment is referred to.

3.1.116 system

A type of product_version that is a regularly interacting or interdependent group of products forming a unified whole under the influence of related forces.

3.1.117 system_user

A system_user is a type of person where the person is associated to a hardware_software system.

— system;

— user_id;

— password;

— access_level.

3.1.117.1 system

The system is the hardware_software system that is associated to the person.

3.1.117.2 user_id

The user_id is the identification of the person that is recognizable by the hardware_software system.

3.1.117.3 password

The password is the hardware_software system recognizable confirmation of the user_id.

3.1.117.4 access_level

The access_level is an indication of what the person (with a successful login of user_id and password) can perform in the hardware_software system..

3.1.118 week_of_year_and_day_date

A week_of_year_and_day_date is a date which is identified by a day in a week of a year.

— week_component;

— day_component.

Informal propositions:

valid_year_and_day: the combination of the day_component and the week_component shall be between 1 and 365 if the year_component is not a leap year, otherwise the combination of the day_component and the week_component shall be between 1 and 366.

3.1.118.1 week_component

The week_component is the week element of the date.

3.1.118.2 day_component

The day_component is the day element of the date.

3.2 Application Interpreted Assertions

This subclause specifies the application specific object assertions for the RASSP Application Interpreted Model. Object assertions specify the relationships between application specific objects, the cardinality of the relationships, and the rules required for the integrity and validity of the application specific objects. The application specific assertions and their definitions are given below.

3.2.1 action to action_method

Each instance of an action defines the method of zero, one, or many action_method instances.

3.2.2 action_assignment to action

Each instance of an action defines the association to zero, one, or many action_assignment instances.

3.2.3 action_execution to ordered_action

Each instance of an ordered_action authorizes zero, one, or many action_execution instances.

3.2.4 action_execution_support_resource to action_execution

Each instance of a action_execution is the executed action for zero, one or many action_execution_support_resource instances.

3.2.5 action_item to product_version

Each instance of an action_item defines a set of one or more product_version instances.

3.2.6 action_method to requested_action

Each instance of an action_method requests a set of one or more requested_action instances.

3.2.7 action_status to action_execution

Each instance of an action_execution has a status defined by a zero, one or more action_status instances.

3.2.8 approval to approval_status

Each instance of approval_status is the status for exactly one approval.

3.2.9 approval_assignment to approval

Each instance of approval is assigned to zero, one or many approval instances.

3.2.10 approval_date_time to approval

Each instance of approval shall be referenced by exactly one approval_date_time. This enforces the requirement for every approval to have a date on which the approval obtained its specific status.

3.2.11 approved_item to product_version

Each instance of an approved_item is for a set of one or more product_version instances.

3.2.12 approval_person_organization to approval

Each instance of approval shall have one or more approval_user_organization referencing it. This rule enforces the requirement for an approval to be authorized by one or more people within their organizations.

3.2.13 approval_person_organization to approval_role

Each instance of an approval_role is the role for zero, one or more approval_person_organization instances.

3.2.14 assembly_component_usage_substitute to assembly_component_usage

Each instance of a assembly_component_usage may the base for zero, one, or more assembly_component_usage_substitute. Each instance of a assembly_component_usage may the substitute for zero, one, or more assembly_component_usage_substitute.

3.2.15 classified_item to product_version

Each instance of a classified_item classifies a set of one or more product_version instances.

3.2.16 configuration_design to configuration_item

Each instance of a configuration_item defines the configuration for zero, one, or many configuration_design instances.

3.2.17 configuration_design to product

Each instance of a product is the design for zero, one, or many configuration_design instances.

3.2.18 configuration_item to product_concept

Each instance of a product_concept is the item concept for zero, one, or many configuration_item instances.

3.2.19 date_and_time to date

Each instance of a date is the component for zero, one, or many date_and_time instances.

3.2.20 date_and_time to local_time

Each instance of a local_time is the component for zero, one, or many date_and_time.

3.2.21 dated_effectivity to date_and_time

Each instance of a date_and_time defines the effectivity_start_date for zero, one, or more dated_effectivity instances. Each instance of a date_and_time may define the effectivity_end_date for zero, one, or more dated_effectivity instances.

3.2.22 derived_unit to derived_unit_element

Each instance of derived unit requires a set of one or more derived_unit_elements.

3.2.23 derived_unit_element to named_unit

Each instance of a named_unit is the unit for zero, one, or many derived_unit_element instances.

3.2.24 document to document_type

Each instance of a document_type is the kind for zero, one, or many document instances.

3.2.25 document_reference to document

Each instance of a document is assigned to zero, one, or many document_reference instances.

3.2.26 file_folder to product_version

Each product_version is electronically represented by zero, one or many associated file_folder instances.

3.2.27 lot_effectivity to measure_with_unit

Each instance of a measure_with_unit defines the lot size of zero, one, or many lot_effectivity instances.

3.2.28 make_from_usage_option to measure_with_unit

Each instance of a measure_with_unit defines the quantity of zero, one, or many make_from_usage_option instances.

3.2.29 named_unit to dimensional_exponents

Each instance of a dimensional_exponents defines the dimensions of zero, one or more named_unit instances.

3.2.30 node_location to hardware_software

Each instance of a hardware_software is defined by the location of zero, one or more node_location instances.

3.2.31 ordered_action to requested_action

Each instance of an ordered_action authorizes a set of one or more requested_action instances.

3.2.32 organization to cage

Each instance of a cage defines the cage code for zero, one or more organization instances.

3.2.33 organizational_address to organization

Each instance of an organizational_address defines the location for a set of one or more organization instances. Each instance of a organization is located at zero, one, or many organizational_address instances.

3.2.34 organizational_project to organization

Each instance of a organizational_project is the responsibility of a set of one or many organization instances.

3.2.35 person_and_organization to organization

Each instance of an organization defines zero, one, or many person_and_organization instances.

3.2.36 person_and_organization to person

Each instance of a person defines zero, one, or many person_and_organization instances.

3.2.37 personal_address to person

Each instance of an personal_address defines the location for a set of one or more person instances. Each instance of a person is located at zero, one, or many personal_address instances.

3.2.38 physical_unit to configuration_design

Each instance of a configuration_design defines the configuration for zero, one, or many physical_unit instances.

3.2.39 planned_effectivity to configuration_design

Each instance of a configuration_design defines the configuration for zero, one, or many planned_effectivity instances.

3.2.40 planned_effectivity to product_definition_usage

Each instance of a product_definition_usage defines the design_usage for zero, one, or many planned_effectivity instances.

3.2.41 product_anomaly_disposition to action_execution

Each instance of a product_anomaly_disposition is dispositioned by a set of one or more action_execution instances.

3.2.42 product_anomaly_disposition to product_anomaly

Each instance of a product_anomaly is resolved by zero, one or many product_anomaly_disposition instances.

3.2.43 product_change to product_anomaly_disposition

Each product_anomaly_disposition defines the baseline product disposition for zero, one or many product_change instances.

3.2.44 product_change to product_requiring_change

Each instance of a product_requiring_change defines the baseline product for zero, one, or many product_change instances.

3.2.45 product_change to product_version

Each instance of a product_version defines the resulting product for zero, one, or many product_change instances.

3.2.46 product_classification to product

Each instance of a product_classification defines a set of one or more product instances.

3.2.47 product_definition to product_version

Each product_version is characterized by zero, one, or many product_definition instances.

3.2.48 product_definition_relationship to product_definition

Each product_definition_relationship is the related_product_definition for zero, one, or many product_defintion instances. Each product_definition_relationship is the relating_product_definition for zero, one, or many product_definition instances.

3.2.49 product_process_step to product

Each instance of product process step requires a set of one or more products.

3.2.50 product_requiring_change to action_execution

Each action_execution requires zero, one, or many product_requiring_change instances.

3.2.51 product_requiring_change to product_anomaly

Each product_requiring_change requires a set of one or many product_anomaly instances. Each product_anomaly defines a set of one or many product_requiring_change instances.

3.2.52 product_responsibility to organizational_project

Each organizational_project defines the project for zero, one or more product_responsibility instances.

3.2.53 product_responsibility to product

Each product defines the product for zero, one or more product_responsibility instances.

3.2.54 product_state to action_execution

Each action_execution defines the action_transition of zero, one or more product_state instances.

3.2.55 product_state to product_version

Each product_version has a lifecycle state defined by zero, one or more product_state instances.

3.2.56 product_version_group to group

Each group is the container for zero, one or more product_version_group instances.

3.2.57 product_version_group to product_version

Each product_version is grouped by (belongs to) zero, one or more product_version_group instances.

3.2.58 product_version to product

Each product is versioned by zero, one or more product_version instances.

3.2.59 quantified_assembly_component_usage to measure_with_unit

Each instance of a measure_with_unit defines the quantity of zero, one, or many quantified_assembly_component_usage instances.

3.2.60 recommended_support_resource to action_item

Each instance of a action_item is the recommended action for zero, one or many recommended_support_resource instances.

3.2.61 related_change to product_anomaly

Each instance of a product_anomaly references zero, one or many related_change instances.

3.2.62 related_change to product_requiring_change

Each instance of a product_requiring_change defines the related product that is changing for zero, one or many related_change instances.

3.2.63 security_classification to security_classification_level

Each instance of a security_classification_level is categorized by zero, one, or many security_classification instances.

3.2.64 security_classification_assignment to security_classification

Each instance of a security_classification is assigned to zero, one, or many security_classification_assignment instances.

3.2.65 serial_numbered_effectivity to physical_unit

Each instance of a physical_unit defines the effectivity start unit for zero, one, or many serial_numbered_effectivity instances. Each instance of a physical_unit may define the effectivity end unit for zero, one, or many serial_numbered_effectivity instances.

3.2.66 specific_higher_usage_occurrence to assembly_component_usage

Each instance of an assembly_component_usage defines the upper usage for zero, one or many specific_higher_usage_occurrence instances.

3.2.67 specific_higher_usage_occurrence to next_assembly_component_usage

Each instance of an next_assembly_component_usage defines the next usage for zero, one or many specific_higher_usage_occurrence instances.

3.2.68 specified_item to product_version

Each instance of a specified_item defines the reference of a set of one or more product_version instances.

3.2.69 supplier to date

Each instance of a date defines the source date for zero, one or many supplier instances.

3.2.70 supplier to product

Each instance of a product defines the product supplied by zero, one or many supplier instances.

3.2.71 system_user to hardware_software

Each instance of a hardware_software defines the association of zero, one or many system_user instances.

Annex A Application Interpreted Model

A.1 RASSP Application Interpreted Model EXPRESS

```
SCHEMA RASSP-Application_Interpreted_Model;

TYPE day_in_month_number = INTEGER;
END_TYPE;

TYPE day_in_week_number = INTEGER;
END_TYPE;

TYPE day_in_year_number = INTEGER;
END_TYPE;
```

```
TYPE hour_in_day = INTEGER;
END_TYPE;

TYPE identifier = STRING;
END_TYPE;

TYPE identifier = STRING;
END_TYPE;

TYPE identifier = ;
END_TYPE;

TYPE identifier = STRING;
END_TYPE;

TYPE identifier = STRING;
END_TYPE;

TYPE identifier = STRING;
END_TYPE;

TYPE identifier = STRING;
END_TYPE;

TYPE identifier = STRING;
END_TYPE;

TYPE identifier = STRING;
END_TYPE;
```

```
TYPE integer = INTEGER;
END_TYPE;

TYPE label = STRING;
END_TYPE;

TYPE label = STRING;
END_TYPE;

TYPE label = STRING;
END_TYPE;

TYPE label = ;
END_TYPE;

TYPE label = ;
END_TYPE;

TYPE label = ;
END_TYPE;

TYPE label = ;
END_TYPE;

TYPE label = ;
END_TYPE;

TYPE label = ;
```

```
END_TYPE;

TYPE label = ;
END_TYPE;

TYPE label = ;
END_TYPE;

TYPE label = STRING;
END_TYPE;

TYPE label = STRING;
END_TYPE;

TYPE label = ;
END_TYPE;

TYPE level = ;
END_TYPE;

TYPE minute_in_hour = ;
END_TYPE;

TYPE month_in_year_number = INTEGER;
END_TYPE;

TYPE second_in_minute = INTEGER;
END_TYPE;
```

```
TYPE text = STRING;
END_TYPE;

TYPE text = STRING;
END_TYPE;

TYPE text = STRING;
END_TYPE;

TYPE text = ;
END_TYPE;

TYPE text = ;
END_TYPE;

TYPE text = ;
END_TYPE;

TYPE text = STRING;
END_TYPE;

TYPE text = STRING;
END_TYPE;

TYPE text = STRING;
END_TYPE;

TYPE text = ;
END_TYPE;
```

```
TYPE text = STRING;

END_TYPE;


TYPE text = STRING;

END_TYPE;


TYPE text = STRING;

END_TYPE;


TYPE text = ;

END_TYPE;


TYPE text = ;

END_TYPE;


TYPE week_in_year_number = INTEGER;

END_TYPE;


TYPE year_number = INTEGER;

END_TYPE;


TYPE ahead_or_behind = ENUMERATION OF

(ahead,

behind);

END_TYPE;


TYPE date_time_select = SELECT

(date,
date_and_time,

local_time);

END_TYPE;


TYPE person_organization_select = SELECT

(person_and_organization,

organization,

person);

END_TYPE;


TYPE support_resource_select = SELECT

(person,

organization,

support_equipment);

END_TYPE;


TYPE unit = SELECT

(derived_unit,

named_unit);

END_TYPE;


ENTITY action

SUPERTYPE OF (action_execution)

SUBTYPE OF (product);

method : action_method;

END_ENTITY;


ENTITY action_assignment

ABSTRACT SUPERTYPE OF (action_item ANDOR product_process_step);
```

```
assigned_action : action;

END_ENTITY;


ENTITY action_execution

SUBTYPE OF (action);

order : ordered_action;

END_ENTITY;


ENTITY action_execution_support_resource;

executed_action : action_execution;

supporting_resources : support_resource_select;

END_ENTITY;


ENTITY action_item

SUBTYPE OF (action_assignment);

items : SET [1:?] OF product_version;

END_ENTITY;


ENTITY action_method

SUBTYPE OF (product);

requests : SET [1:?] OF requested_action;

purpose : text;

consequence : text;

END_ENTITY;


ENTITY action_status;

assigned_action : action_execution;

status : label;

END_ENTITY;
```

```
ENTITY address
SUPERTYPE OF (personal_address ANDOR organizational_address);
telex_number : OPTIONAL label;
electronic_mail_address : OPTIONAL label;
telephone_number : OPTIONAL label;
facsimile_number : OPTIONAL label;
country : OPTIONAL label;
postal_code : OPTIONAL label;
region : OPTIONAL label;
town : OPTIONAL label;
postal_box : OPTIONAL label;
street : OPTIONAL label;
street_number : OPTIONAL label;
mail_stop : OPTIONAL label;
END_ENTITY;

ENTITY approval;
status : approval_status;
level : label;
END_ENTITY;

ENTITY approval_assignment
ABSTRACT SUPERTYPE OF (approved_item);
assigned_approval : approval;
END_ENTITY;

ENTITY approval_date_time;
dated_approval : approval;
```

```
date_time : date_time_select;

END_ENTITY;


ENTITY approval_person_organization;

authorized_approval : approval;

role : approval_role;

person_organization : person_organization_select;

END_ENTITY;


ENTITY approval_role;

role : label;

END_ENTITY;


ENTITY approval_status;

name : label;

END_ENTITY;


ENTITY approved_item

SUBTYPE OF (approval_assignment);

items : SET [1:?] OF product_version;

END_ENTITY;


ENTITY assembly_component_usage

SUPERTYPE OF (quantified_assembly_component_usage ANDOR
ONEOF(promissory_usage_occurence,specified_higher_usage_occurence,next_assembly_

SUBTYPE OF (product_definition_usage);

reference_designator : OPTIONAL identifier;

END_ENTITY;
```

```
ENTITY assembly_component_usage_substitute;
base : assembly_component_usage;
substitute : assembly_component_usage;
UNIQUE
UR1: base,substitute;
END_ENTITY;

ENTITY cage;
cage_code : identifier;
END_ENTITY;

ENTITY calendar_date
SUBTYPE OF (date);
day_component : day_in_month_number;
month_component : month_in_year_number;
END_ENTITY;

ENTITY classified_item
SUBTYPE OF (security_classification_assignment);
items : SET [1:?] OF product_version;
END_ENTITY;

ENTITY configuration_design;
design : product_version;
configuration : configuration_item;
UNIQUE
UR1: configuration,design;
END_ENTITY;
```

```
ENTITY configuration_item
SUPERTYPE OF (signal_processor_design)
SUBTYPE OF (product);
item_concept : product_concept;
purpose : label;
UNIQUE
UR1: identification;
END_ENTITY;

ENTITY coordinated_universal_time_offset;
sense : ahead_or_behind;
hour_offset : hour_in_day;
minute_offset : OPTIONAL minute_in_hour;
END_ENTITY;

ENTITY correspondence
SUBTYPE OF (document);
END_ENTITY;

ENTITY data_template
SUBTYPE OF (product);
END_ENTITY;

ENTITY date
SUPERTYPE OF (ONEOF(ordinal_date,calendar_date,week_of_year_and_day_date));
year_component : year_number;
END_ENTITY;

ENTITY date_and_time;
```

```
date_component : date;

time_component : local_time;

END_ENTITY;


ENTITY dated_effectivity

SUBTYPE OF (planned_effectivity);

effectivity_end_date : OPTIONAL date_and_time;

effectivity_start_date : date_and_time;

END_ENTITY;


ENTITY derived_unit;

elements : SET [1:?] OF derived_unit_element;

END_ENTITY;


ENTITY derived_unit_element;

unit : named_unit;

exponent : REAL;

END_ENTITY;


ENTITY dimensional_exponents;

length_exponent : REAL;

mass_exponent : REAL;

time_exponent : REAL;

electric_current_exponent : REAL;

thermodynamic_temperature_exponent : REAL;

amount_of_substance_exponent : REAL;

luminous_intensity_exponent : REAL;

END_ENTITY;
```

```
ENTITY discrepant_product

SUBTYPE OF (product_requiring_change);

failure_rate : SET [1:?] OF REAL;

END_ENTITY;


ENTITY document

SUPERTYPE OF (specification ANDOR correspondence ANDOR drawing ANDOR
procedure ANDOR publication)

SUBTYPE OF (product);

kind : document_type;

size : integer;

UNIQUE

UR1: id;

END_ENTITY;


ENTITY document_reference

ABSTRACT SUPERTYPE OF (specified_item);

assigned_document : document;

END_ENTITY;


ENTITY document_type;

product_data_type : label;

END_ENTITY;


ENTITY drawing

SUBTYPE OF (document);

END_ENTITY;


ENTITY enhancement_product

SUBTYPE OF (product_requiring_change);
```

```
END_ENTITY;

ENTITY enterprise
SUBTYPE OF (organization);
END_ENTITY;

ENTITY file_folder
SUBTYPE OF (physical_unit);
file_type : label;
representative_product : product_version;
END_ENTITY;

ENTITY group;
group_name : text;
END_ENTITY;

ENTITY hardware_software
SUBTYPE OF (system);
END_ENTITY;

ENTITY local_time;
zone : coordinated_universal_time_offset;
hour_component : hour_in_day;
minute_component : OPTIONAL minute_in_hour;
second_component : OPTIONAL second_in_minute;
END_ENTITY;

ENTITY lot_effectivity
SUBTYPE OF (planned_effectivity);
```

```
effectivity_lot_size : measure_with_unit;
effectivity_lot_id : identifier;
END_ENTITY;

ENTITY make_from_usage_option
SUBTYPE OF (product_definition_usage);
quantity : measure_with_unit;
ranking_rationale : text;
ranking : INTEGER;
END_ENTITY;

ENTITY measure_with_unit;
unit_component : unit;
value_component : REAL;
END_ENTITY;

ENTITY named_unit;
dimensions : dimensional_exponents;
END_ENTITY;

ENTITY next_assembly_usage_occurence
SUBTYPE OF (assembly_component_usage);
END_ENTITY;

ENTITY node_location;
system : hardware_software;
protocol : text;
node_address : text;
END_ENTITY;
```

```
ENTITY ordered_action;
requests : SET [1:?] OF requested_action;
name : label;
description : text;
comment : text;
analysis : text;
END_ENTITY;

ENTITY ordinal_date
SUBTYPE OF (date);
day_component : day_in_year_number;
END_ENTITY;

ENTITY organization
SUPERTYPE OF (ONEOF(enterprise,program) ANDOR supplier)
SUBTYPE OF (product);
cage_code : cage;
END_ENTITY;

ENTITY organizational_address
SUBTYPE OF (address);
organizations : SET [1:?] OF organization;
END_ENTITY;

ENTITY organizational_project;
responsible_organizations : SET [1:?] OF organization;
description : text;
name : label;
```

```
END_ENTITY;

ENTITY part
SUPERTYPE OF (software_application ANDOR reuse_part)
SUBTYPE OF (product);
part_configuration_identifier : identifier;
part_function_type : text;
part_type : text;
END_ENTITY;

ENTITY person
SUPERTYPE OF (system_user)
SUBTYPE OF (product);
last_name : label;
first_name : label;
suffix_titles : OPTIONAL SET [1:?] OF label;
prefix_titles : OPTIONAL SET [1:?] OF label;
middle_names : OPTIONAL SET [1:?] OF label;
UNIQUE
UR1: id;
END_ENTITY;

ENTITY person_and_organization;
the_person : person;
the_organization : organization;
END_ENTITY;

ENTITY personal_address
SUBTYPE OF (address);
```

```
people : SET [1:?] OF person;

END_ENTITY;



ENTITY physical_unit

SUPERTYPE OF (file_folder)

SUBTYPE OF (product_version);

configuration : configuration_design;

UNIQUE

UR1: configuration;

END_ENTITY;



ENTITY planned_effectivity

SUPERTYPE OF
(ONEOF(serial_numbered_effectivity,lot_effectivity,dated_effectivity));

configuration : configuration_design;

design_usage : product_definition_usage;

identification : identifier;

UNIQUE

UR1: identification,configuration,design_usage;

END_ENTITY;



ENTITY procedure

SUBTYPE OF (document);

END_ENTITY;



ENTITY product

SUPERTYPE OF (part ANDOR action_method ANDOR action ANDOR configuration_item
ANDOR product_concept ANDOR document ANDOR person ANDOR organization ANDOR
data_template ANDOR system);

description : text;
```

```
frame_of_reference : label;
name : label;
id : identifier;
UNIQUE
UR1: id;
END_ENTITY;

ENTITY product_anomaly
SUPERTYPE OF (product_issue ANDOR product_concern ANDOR product_flaw);
product_anomaly_identifier : identifier;
product_anomaly_description : text;
detection_method : text;
anomaly_type : text;
anomaly_cause : text;
INVERSE
products : SET[1:?] OF product_requiring_change FOR anomalized_products;
END_ENTITY;

ENTITY product_anomaly_disposition;
anomalized_product : product_anomaly;
disposition_actions : SET [1:?] OF action_execution;
END_ENTITY;

ENTITY product_change;
baseline_product : product_requiring_change;
baseline_product_disposition : product_anomaly_disposition;
resulting_product : product_version;
reasons : SET [1:?] OF text;
END_ENTITY;
```

```
ENTITY product_classification

SUBTYPE OF (security_classification_assignment);

items : SET [1:?] OF product;

END_ENTITY;


ENTITY product_concept

SUBTYPE OF (product);

product_concept_context : label;

UNIQUE

UR1: identification;

END_ENTITY;


ENTITY product_concern

SUBTYPE OF (product_anomaly);

END_ENTITY;


ENTITY product_definition;

version : product_version;

description : text;

frame_of_reference : label;

END_ENTITY;


ENTITY product_definition_relationship

SUPERTYPE OF (product_definition_usage);

related_product_definition : product_definition;

relating_product_definition : product_definition;

id : identifier;

name : label;
```

```
description : text;

END_ENTITY;



ENTITY product_definition_usage

SUPERTYPE OF (ONEOF(make_from_usage_option,assembly_component_usage))

SUBTYPE OF (product_definition_relationship);

UNIQUE

UR1:
SELF\product_definition_relationship.id,SELF\product_definition_relationship.rel

END_ENTITY;



ENTITY product_flaw

SUBTYPE OF (product_anomaly);

product_flaw_type : text;

END_ENTITY;



ENTITY product_issue

SUBTYPE OF (product_anomaly);

END_ENTITY;



ENTITY product_process_step

SUBTYPE OF (action_assignment);

products : SET [1:?] OF product;

END_ENTITY;



ENTITY product_requiring_change

SUPERTYPE OF (ONEOF(discrepant_product,enhancement_product) ANDOR
related_change)

SUBTYPE OF (product_version);

anomalized_products : SET [1:?] OF product_anomaly;
```

```
product_change_requirement_type : text;
requiring_change_product : action_execution;
END_ENTITY;

ENTITY product_responsibility;
project : organizational_project;
product : product;
END_ENTITY;

ENTITY product_state;
product : product_version;
action_transition : action_execution;
state_name : label;
END_ENTITY;

ENTITY product_version
SUPERTYPE OF (product_requiring_change ANDOR physical_unit);
of_product : product;
description : text;
version_id : identifier;
UNIQUE
UR1: version_id, of_product;
END_ENTITY;

ENTITY product_version_group;
version : product_version;
group : group;
END_ENTITY;
```

```
ENTITY program
SUBTYPE OF (organization);
END_ENTITY;

ENTITY promissory_usage_occurence
SUBTYPE OF (assembly_component_usage);
END_ENTITY;

ENTITY publication
SUBTYPE OF (document);
END_ENTITY;

ENTITY quantified_assembly_component_usage
SUBTYPE OF (assembly_component_usage);
quantity : measure_with_unit;
END_ENTITY;

ENTITY recommended_support_resource;
recommended_action : action_item;
supporting_resource : support_resource_select;
recommended_role : role;
END_ENTITY;

ENTITY related_change
SUBTYPE OF (product_requiring_change);
related_change_product : product_requiring_change;
anomalized_product : product_anomaly;
END_ENTITY;
```

```
ENTITY requested_action;
id : identifier;
version : label;
purpose : text;
description : text;
END_ENTITY;

ENTITY reuse_part
SUBTYPE OF (part);
END_ENTITY;

ENTITY role;
role_name : label;
END_ENTITY;

ENTITY security_classification;
security_level : security_classification_level;
name : label;
purpose : text;
END_ENTITY;

ENTITY security_classification_assignment
ABSTRACT SUPERTYPE OF (classified_item ANDOR product_classification);
assigned_security_classification : security_classification;
END_ENTITY;

ENTITY security_classification_level;
name : label;
END_ENTITY;
```

```
ENTITY serial_numbered_effectivity

SUBTYPE OF (planned_effectivity);

effectivity_start_id : physical_unit;

effectivity_end_id : OPTIONAL physical_unit;

END_ENTITY;


ENTITY signal_processor_design

SUBTYPE OF (configuration_item);

END_ENTITY;


ENTITY software_application

SUBTYPE OF (part);

software_language : text;

END_ENTITY;


ENTITY specification

SUBTYPE OF (document);

END_ENTITY;


ENTITY specified_higher_usage_occurence

SUBTYPE OF (assembly_component_usage);

next_usage : next_assembly_usage_occurence;

upper_usage : assembly_component_usage;

UNIQUE

UR1: upper_usage,next_usage;

END_ENTITY;


ENTITY specified_item
```

```
SUBTYPE OF (document_reference);

items : SET [1:?] OF product_version;

END_ENTITY;


ENTITY supplier

SUBTYPE OF (organization);

source_date : date;

source_product : product;

END_ENTITY;


ENTITY support_equipment;

name : label;

END_ENTITY;


ENTITY system

SUPERTYPE OF (hardware_software)

SUBTYPE OF (product);

END_ENTITY;


ENTITY system_user

SUBTYPE OF (person);

system : hardware_software;

access_level : level;

user_id : text;

password : text;

END_ENTITY;


ENTITY week_of_year_and_day_date

SUBTYPE OF (date);
```

```
week_component : week_in_year_number;

day_component : OPTIONAL day_in_week_number;

END_ENTITY;


END_SCHEMA;
```

A.2 RASSP Application Interpreted Model EXPRESS-G

The EXPRESS-G diagrams for the RASSP Application Interpreted Model are shown in the following pages. Table A.1 shows the position of each page in order to assemble the AIM as a single diagram.

83	85	87	89
91	93	95	97
99	101	103	105

Table A.1 - RASSP Application Interpreted Model Page Positions

Annex B ARM to AIM Mapping Table

This clause contains the mapping table that shows how each application reference object maps to one or several application interpreted objects via enterprise objects.

The mapping table is organized in four columns. The contents of these four columns are:

Column 1) Application element: Name of an application element as it appears in the application reference object definition in section 3.2 of the Build 1 Application Reference Model Report. Application reference object names are written in uppercase. Attribute names and application assertion names are listed after the application object to which the belong and are written in lowercase.

Column 2) AIM element: Name of an AIM element as it appears in the Application Interpreted Information Requirements (3). AIM entities are written in lowercase. Attribute name of AIM entities are referred to as (entity name).(attribute name). The mapping of an application element may result in several related AIM elements. Each of these AIM elements will require a line of its own in the table.

Column 3) Source: For those AIM elements that are interpreted from the RASSP Enterprise Data Model (REDM), this is the callout of the REDM. For those AIM elements that are created for the purpose of file configuration management, this is the callout of the ARM.

Column 4) Reference path: To fully describe the mapping of an ARM object, it may be necessary to specify a reference path through several related AIM objects. A single AIM object is documented on a single row within the reference path column with possibly a symbol which defines its relationship to the AIM object on the succeeding or preceding row in the column. The reference path column, therefore, documents the role of an AIM object relative to the AIM object in the row succeeding or preceding it. Two or more such related AIM objects defined the interpretation of the RASSP Enterprise Data Model which satisfy the requirement specified by the ARM object if a reference path is provided. For each AIM object that has been created for use within file configuration management, a reference path up to its REDM object is specified. For each ARM object that has been identified as a category entity of another ARM object, a reference path up to its identified parent entity is specified. Portions of reference paths may be enclosed by a set of parentheses. This is to signify either a bi-directional reference from an AIM object with two attributes, each of which spawns a reference path, or to signify alternatives for the reference path.

For the expression of reference paths, the following notational conventions apply:

a.) -> attribute references ENTITY or SELECT type given in the following row;

b.) <- ENTITY or SELECT type is referenced by the attribute in the following row;

c.) => entity is a SUPERTYPE of the entity given in the following row;

d.) <= entity is a SUBTYPE of the entity given in the following row;

e.) = a SELECT or ENUMERATION type identifies one choice within its list of alternatives which has possibly been constrained to particular choices or values.

Application element (ARM)	AIM element	Sourc
approval	approval	REDM
authorized_approval	ordered_action.name	REDM
level	level	REDM
status	status	REDM
approval_date_time	approval_date_time	REDM
date_time	date_and_time	REDM

dated_approval	dated_approval	REDM
approval_status	approval_status	REDM
name	name	REDM
approval_user_organization	approval_person_organization	REDM
authorized_approval	authorized_approval	REDM
user_organization	person_organization	REDM
assembly_component_usage	assembly_component_usage	REDM
context_product	relating_product_definition	REDM
component_product	related_product_definition	REDM
assembly_component_usage_substitute	assembly_component_usage_substitute	REDM
base	base	REDM
substitute	substitute	REDM
authorization	ordered_action	REDM
issuing_organization	organization.cage_code	REDM
CHANGE_REQUIREMENT_TYPE	product_requiring_change.product_change_requirement_type	REDM
CLMS_COMPONENT	reuse_part	REDM
configuration_design	configuration_design	REDM
design	design	REDM
configuration	configuration	REDM
configuration_item	configuration_item	REDM
purpose	purpose	REDM
date	date	REDM
year_component	year_component	REDM

month_component	calendar_date.month_component	REDM
day_component	calendar_date.day_component OR ordinal_date.day_component OR week_of_year_and_day_date.day_component	REDM
date_and_time	date_and_time	REDM
date_component	date_component	REDM
time_component	time_component	REDM
dated_effectivity	dated_effectivity	REDM
effectivity_start_date	effectivity_start_date	REDM
effectivity_end_date	effectivity_end_date	REDM
derived_unit	derived_unit	REDM
elements	elements	REDM
derived_unit_element	derived_unit_element	REDM
unit	unit	REDM
exponent	exponent	REDM
DOCUMENT	document	REDM
role	kind	REDM
size	size	REDM
file_version	file_folder	REDM
associated_product	representative_product	REDM

creation_date_time	approval_date_time.date_time	REDM
description	product_version.description	REDM
id	product_version.of_product	REDM
name	product.name	REDM

person	person.last_name person.first_name	REDM
item_anomaly	product_anomaly	REDM
anomaly_cause	anomaly_cause	REDM
anomaly_type	anomaly_type	REDM
detection_method	detection_method	REDM
description	product_anomaly_description	REDM
id	product_anomaly_identifier	REDM
item_anomaly_disposition	product_anomaly_disposition	REDM
anomalized_item	anomalized_product	REDM
disposition_process_step	disposition_actions	REDM
ITEM_ANOMALY_TYPE	product_anomaly.anomaly_type	REDM
iteM_change	product_change	REDM

baseline_item	baseline_product	REDM
baseline_item_disposition	baseline_product_disposition	REDM
reason	reasons	REDM
resultant_item	resulting_product	REDM
item_requiring_change	product_requiring_change	REDM
change_requirement_type	product_change_requirement_type	REDM
item_need	anomalized_products	REDM
requiring_change_process_step	requiring_change_product	REDM
requiring_item	product_version.of_product	REDM
location	address	REDM
mail_stop	mail_stop	REDM
postal_box	postal_box	REDM
street	street	REDM
street_number	street_number	REDM
town	town	REDM
region	region	REDM
postal_code	postal_code	REDM
country	country	REDM
facsimile_number	facsimile_number	REDM
telephone_number	telephone_number	REDM
electronic_mail_address	electronic_mail_address	REDM
lot_effectivity	lot_effectivity	REDM
effectivity_lot_id	effectivity_lot_id	REDM
effectivity_lot_size	effectivity_lot_size	REDM
make_from_usage_option	make_from_usage_option	REDM
context_product	relating_product_definition	REDM
component_product	related_product_definition	REDM
quantity	quantity	REDM
measure_with_unit	measure_with_unit	REDM
value_component	value_component	REDM
unit_component	unit_component	REDM
named_unit	named_unit	REDM

next_assembly_usage_occurence	next_assembly_usage_occurence	REDM
context_product	relating_product_definition	REDM
component_product	related_product_definition	REDM
organization	organization	REDM
name	product.name	REDM
cage_code	cage_code	REDM
organization_location	organizational_address	REDM
organizations	organizations	REDM
PART	part	REDM
part_document_association	specified_item	REDM
assigned_document	document_reference.assigned_document	REDM
assigned_part	items	REDM
planned_effectivity	planned_effectivity	REDM
id	identification	REDM
configuration	configuration	REDM
design_usage	design_usage	REDM
process_execution_resource	action_execution_support_resource	REDM
executed_process_step	executed_action	REDM
resource_tool	support_equipment.name	REDM

process_executor	person.last_name person.first_name	REDM
process_steP	action	REDM
consequence	action_method.consequence	REDM
purpose	action_method.purpose	REDM
process_step_execution	action_execution	REDM
authorized_step	order	REDM
requiring_product_process_step	action.method	REDM
run_identifier	requested_action.id	REDM
process_steP_execution_status	action_status	REDM

executed_process_step	assigned_action	REDM
status_date_time		
status_value	status	REDM
PRODUCT	product	REDM
description	description	REDM
frame_of_reference	frame_of_reference	REDM
id	id	REDM
name	name	REDM
product_approval	approved_item	REDM
approved_file	file_folder.represntative_product	REDM
assigned_approval	approval_assignment. assigned_approval	REDM
PRODUCT_CLASSIFICATION	product_classification	REDM
assigned_security_classification	security_classification_assignment. assigned_security_classification	REDM
classified_product	items	REDM
PRODUCT_DEFINITION	product_definition	REDM
functional_definition_id	frame_of_reference	REDM
product_id	version	REDM
product_process_step	product_process_step	REDM

assigned_process_step	action_assignment.assigned_action	REDM
product	products	REDM
resource_role	role.role_name	REDM
product_usage	product_definition_usage	REDM
id	id	REDM
context_product	relating_product_definition	REDM
component_product	related_product_definition	REDM
project	organizational_project	REDM
name	name	REDM
description	description	REDM
responsible_organizations	responsible_organizations	REDM
projecT_assignment	product_responsibility	REDM
assigned_product	product	REDM
assigned_project	project	REDM
promissory_usage_occurence	promissory_usage_occurence	REDM
context_product	relating_product_definition	REDM
component_product	related_product_definition	REDM
quantified_assembly_component_usage	quantified_assembly_component_usage	REDM
context_product	relating_product_definition	REDM
component_product	related_product_definition	REDM
quantity	quantity	REDM
related_change	related_change	REDM
anomalized_item	anomalized_product	REDM
REUSE_PART	reuse_part	REDM

library_component	product.id	REDM
role	role	REDM
role_name	role_name	REDM
SECURITY_CLASSIFICATION	security_classification	REDM
security_level_name	security_classification_level.name	REDM
purpose	purpose	REDM
serial_numbered_effectivity	serial_numbered_effectivity	REDM
effectivity_start_id	effectivity_start_id	REDM
effectivity_end_id	effectivity_end_id	REDM
SIGNAL_PROCESSOR_DESIGN	signal_processor_design	REDM
designed_part	configuration_design.design	REDM
signal_processor_document_association	specified_item	REDM
assigned_signal_processor	items	REDM
assigned_document	document_reference.assigned_document	REDM
SOFTWARE_PART	software_application	REDM
state	product_state	REDM
state_name	state_name	REDM
specified_higher_usage_occurence	specified_higher_usage_occurence	REDM
upper_usage	upper_usage	REDM
next_usage	next_usage	REDM

context_product	relating_product_definition	REDM
component_product	related_product_definition	REDM
time	local_time	REDM
hour_component	hour_component	REDM
minute_component	minute_component	REDM
second_component	second_component	REDM
tool	software_application	REDM
id	product.id	REDM
name	product.name	REDM
user	person	REDM
user_id	product_id	REDM
user_role	approval_person_organization.role	REDM
password		
name	first_name last_name	REDM
user_and_organization	person_and_organization	REDM
the_user	the_person	REDM
the_organization	the_organization	REDM
USER_location	personal_address	REDM

users	people	REDM
workflow	action	REDM
id	product.id	REDM
name	product.name	REDM
steps	method	REDM

Annex C Bibliography

ISO 10303-11 *Industrial automation - Product data representation and exchange*

- Part 11: The EXPRESS Language Reference Manual.

ISO/DIS 10303-41 *Industrial automation systems and integration - Product data representation and exchange: Integrated generic resources: Fundamentals of product description and support.*

ISO/DIS 10303-44 *Industrial automation systems and integration - Product data representation and exchange: Integrated generic resources: Product structure configuration.*

ISO/WD 10303-49 *Industrial automation systems and integration - Product data representation and exchange: Integrated generic resources: Process plan structure.*

MMC-RASSP-2.01.00 *STEP Configuration Management Suitability Report.*

NA-94-1387 *Rapid Prototyping of Application Specific Signal Processors (RASSP)*

Build 0 Information Model Report.

```
usage_occurence))
```

```
ating_product_definition,SELF\product_definition_relationship.related_product_def
```

e	Reference path
I	
I	approval <- assigned_approval.approval_assignment approval_assignment => approved_item approved_item.items -> product_version product_version <- action_itemitems action_item <= action_assignment action_assignment.assigned_action -> action action => action_execution action_execution.order -> ordered_action order_action.name
I	
I	
I	
I	

ordered_action <-

action_execution.order

action_execution <=

action <=

product =>

organization

organization.cage_code

I	date => calendar_date calendar_date.month_component
I	date => (calendar_date calendar_date.day_component) (ordinal_date ordinal_date.day_component) (week_of_year_and_day_date week_of_year_and_day_date. day_component)
I	
I	
I	
I	
I	
I	
I	
I	
I	
I	
I	
I	
I	
I	
I	
I	

I	file_folder <= physical_unit <= product_version <- approved_item.items approved_item <= approval_assignment approval_assignment.assigned_approval -> approval <- approval_date_time approval_date_time.date_time
I	file_folder <= physical_unit <= product_version product_version.description
I	file_folder <= physical_unit <= product_version product_version.of_product
I	file_folder <= physical_unit <= product_version product_version.of_product -> product product.name

I	file_folder <= physical_unit <= product_version product_version <- action_item.items action_item <= action_assignment action_assignment. assigned_action -> action action => action_execution <- action_execution_support_resource.executed_action action_execution_support_resource.supporting_resources -> support_resource_select support_resource_select = person (person.last_name person_first_name)
I	
I	
I	
I	
I	
I	
I	
I	
I	
I	
I	

product_requiring_change <=

product_version

product_version.of_product

[	
[	
[	
[	
[	organization <= product product.name
[	
[	
[	
[	
[	where the attribute items must be a product of subtype part
[	specified_item <= document_reference document_reference. assigned_document
[	
[	
[	
[	
[	
[	
[	
[	action_execution_support_resource action_execution_support_resource.supporting_resources -> support_resource_select support_resource_select = support_equipment support_equipment.name

I	action_execution_support_resource action_execution_support_resource.supporting_resources -> support_resource_select support_resource_select = person (person.last_name person.first_name)
I	
I	action action.method -> action_method action_method.consequence
I	action action.method -> action_method action_method.purpose
I	
I	
I	action_execution <= action action.method
I	action_execution action_execution.order -> ordered_action ordered_action.requests -> requested_action requested_action.id
I	

[	
[	
[	
[	
[	
[	
[	
[	where the attribute items must be a product of subtype physical_unit and sub-subtype file_folder
[	approved_item approved_item.items -> product_version => physical_unit => file_folder file_folder.represntative_product
[	approved_item <= approval_assignment approval_assignment. assigned_approval
[	
[	product_classification <= security_classification_assignment security_classification_assignment. assigned_security_classification
[	
[	
[	
[	
[	

[	product_process_step <= action_assignmenmt action_assignment.assigned_action
[	
[	product_process_step <= action_assignmenmt => action_item <- recommended_support_resource. recommended_action recommended_support_resource => role role.role_name
[	
[	
[	
[	
[	
[	
[	
[	
[	
[	
[	
[	
[	
[	
[	
[	
[	
[	
[	
[	
[	

[	reuse_part <= part <= product product.id
[	
[	
[	
[	security_classification -> security_classification_level security_classification_level.name
[	
[	
[	
[	
[	
[	signal_processor_design <= configuration_item <- configuration_design.configuration configuration_design.design
[	
[	
[	specified_item <= document_reference document_reference. assigned_document
[	
[	
[	
[	
[	
[	

[	
[	
[	
[	
[	
[	
[	
[	software_application <= part <= product product.id
[	software_application <= part <= product product.name
[	
[	person <= product product.id
[	person = person_organization_select <- approval_person_organization. person_organization approval_person_organization.role

[	
[	
[	
[	
[	

I	
I	
I	action <= product product.id
I	action <= product product.name
I	

```
inition;
```