Recommended Practices for AP 209 ed2
10303-209:2014

March 30, 2016
Version 2.0

Table of Contents

Figures

Preface

This document describes the recommended structure and the attribute qualifications in building data models based on the EXPRESS entities, types, and rules in the STEP AP 209 ed2 (multidisciplinary_analysis_and_design) schema. Sections 2.1 through 2.8 of this document dealing with configuration control and shape representations are based on the corresponding sections in the "Recommended Practices for AP 203" document published by PDES, Inc. Additional documents are available to clarify aspects of AP 209 ed2 not covered in this document such as Geometric Founding in ISO 10303-209 RevC and CAx-IF Recommended Practices for Composite Materials V 3.0 dated 2016.

Please direct questions relating to this document to:

Keith A. Hunten (Email: KAHunten@gmail.com)

1. Introduction

ISO 10303-209 ed2, Application Protocol: Multidisciplinary Analysis and Design, provides the data structures for the exchange of part and configuration identification with configuration control data with or without associated 3D part model information. This international standard was developed under the auspices of the International Organization for Standardization (ISO) and is one of a series of parts comprising the full STandard for the Exchange of Product model data (STEP) standard known as ISO 10303.

The Recommended Practices document has been prepared as a usage guide for industry. This document assumes that the reader has at least a rudimentary knowledge of both AP 209 ed2 and its associated application domain. The figures in this document are intended to provide a navigational view of portions of the AP with boxes representing entities, lines being relationships, and arrow heads indicating the pointer direction.

As previously stated, AP 209 ed2 is but one part of the entire ISO 10303 product data standard. It was developed to represent one domain which is described in its scope section. The second edition expands this domain to encompass a broader set of engineering analyses and analysis methods. The analysis types in AP 209 ed2 include a generic engineering analysis core covering mesh-based and meshless analyses, a generic mesh-based numerical analysis and computational fluid dynamics (CFD) capability. However, the 2.0 version of this Recommended Practices document covers only the classical Finite Element capabilities initially covered in ed1 and is centered on the design and analysis phase of composite and metallic structural parts.

AP 209 ed2 does not present configuration management of a product throughout its entire life cycle. As STEP evolves, other APs (currently under development or proposed) will carry the data in AP 209 ed2 forward through the product life cycle.

This document will provide pre- and post-processor recommendations where attributes from the conceptual STEP data models may not actually have values in the AP 209 ed2 application domain. The terms pre-processor and post-processor refer to the applications that write and read the AP 209 ed2 data respectively. In these recommendations, the term ‘no standard mapping’ means there is no mapping defined in the AP's ARM-to-AIM mapping table for the data.

AP 209 ed2 has been harmonized completely with AP 242 (Managed Model-based 3D Engineering) that is an intersection of the requirements of AP 203 (Configuration Controlled 3D Design of Mechanical Parts and Assemblies), and AP 214 (Core Data for Automotive Mechanical Design Processes). The modular STEP architecture enables application modules to be reused. In this fashion, AP 209 ed2 includes all the AP 242 ed2 modules and is a superset of AP 242 ed2.

2. Using AP 209 ed2

This section describes how AP 209 ed2 is intended to be used. The sole purpose of this section is to present the flow of the everyday engineering actions put forth in the AP 209 ed2 activity model using the application terms of the reference model tied to the constructs in the interpreted model. This section will establish limits on some of the data constructs that are not constrained in the Application Interpreted Model (AIM) of the Application Protocol (AP).

2.1. General

In using AP 209 ed2, there are some constructs that have global applicability across all data in the exchange. These constructs relate to the file header for physical file exchanges, data definitions within the file related to the AP itself, and fundamental constructs that contain the information for people, organizations, dates, times, approvals, security classifications, and units of measure.

2.1.1. The STEP Physical File

At present, the standard way to externalize AP 209 ed2 data is via a physical file that is an ASCII encoding of the data based on the EXPRESS constructs in AP 209 ed2 as mapped through ISO 10303-21. The **file_schema.schema_identifiers** attribute in the header section of an AP 209 ed2 physical file shall have the string **`'AP209_MULTIDISCIPLINARY_ANALYSIS_AND_DESIGN_MIM_LF'`** as the only element in the list. The user is referred to ISO parts 10303-11 for definitions of EXPRESS constructs and 10303-21 for information on how to map the EXPRESS constructs in AP 209 ed2 to the physical file.

Post-processor Recommendations: Post-processors should note all errors found during the reading of the file. It is recommended that post-processors provide options to the user on whether to continue when an error is encountered. There are no recommendations on what a post-processor should do with erroneous data. This is left to the discretion of the implementor. If the implementor elects to correct erroneous data, the post-processor should inform the user (as above) of the bad data and what correction was made.

2.1.2. AP Identification and Contexts

STEP is an intelligent data standard and as such the representation of the data for an AP identifies the AP data structure through computer-sensible data. This is done through the entities **application_context** and **application_protocol_definition**. These entities and related context elements are shown in Figure 1.

The **application_context** entity identifies the application that defines the data. The **application** attribute, based on its definition in ISO 10303-41, should have the value **`'AP209_MULTIDISCIPLINARY_ANALYSIS_AND_DESIGN_MIM_LF'`** as this is the application domain AP 209 ed2 is meant to cover.

The **application_protocol_definition** entity further identifies the AP. For AP 209 ed2, the **status** attribute, based on its definition in ISO 10303-41, should have the value **`'international standard'`**. The **application_interpreted_model_schema_name** attribute should have the value **`'AP209_MULTIDISCIPLINARY_ANALYSIS_AND_DESIGN_MIM_LF'`** and the **application_protocol_year** attribute should have the value **`'2014'`**, based on their definitions in ISO 10303-41.

The application identified by the **application_context** entity is broken down into elements in STEP. In AP 209 ed2, these elements are context entities. For AP 209 ed2, the valid context entities are **product_context**, **product_definition_context**, and **product_concept_context**.

The **product_context** entity identifies from what engineering discipline's point of view the data is being presented. This entity will establish the viewing perspective and therefore the requirements source for **product** entities that are defined in Sections 2.8.1.1 and 2.9.1.1.

The **product_definition_context** entity identifies the life cycle stage or maturity of the data being presented. For AP 209 ed2, the value of the **life_cycle_stage** attribute is restricted to be **`'DESIGN'`** or

'ANALYSIS'. This restriction is enforced through the **restrict_product_definition_context** rule in AP 209 ed2. The **product_definition_context** entities will establish the viewing perspective and therefore the requirements source for **product_definition** entities that are defined in Sections 2.8.1.3 and 2.9.1.4.

The **product_concept_context** entity identifies what market segment or customers provided requirements for the data. This entity will establish the source of the requirements for **product_concept** entities that are defined in Section 2.8.14.

Pre-processor Recommendations: There is no standard mapping for the **name** attribute for **product_context**, **product_definition_context**, and **product_concept_context**. Therefore, it is recommended that this attribute contain a null string as minimal content or any appropriate or mutually-agreed-upon string.

Post-processor Recommendations: Since there is no standard mapping for the **name** attribute for **product_context**, **product_definition_context**, and **product_concept_context**, it is recommended that post-processors not assign any processing significance to this value.

2.2. People and Organizations

AP 209 ed2 represents people and organizations as they perform functions related to other data and data relationships. In general, a **person** will exist in the context of some **organization**. A **person** in an **organization** is then associated to the data or data relationship in some role indicating the function being performed. AP 209 ed2 does allow for a person to exist independent of an organization.

Both people and organizations have addresses associated with them. This is done through the **address** entity being related to the **person** (through **personal_address**) or **organization** (through **organizational_address**).

2.2.1. People

AP 209 ed2 specifies information about people through the **person** entity. A **person** is identified by an **id** with other data representing their name and, optionally, titles that may apply to them. In populating the data, the **id** must be unique. This is typically not a problem when the person is taken in the context of some specific group such as a company or even country. In these instances, there are typically identifying numbers assigned to people. If the data being assembled is for worldwide consumption, the **id** must be unique in that domain.

Pre-processor Recommendations: All pre-processors should provide values for at least the **last_name** and **first_name** attributes for the **person** entity in order to provide a sense of meaning to the **id** attribute.

In cases where uniqueness of the **id** attribute may be a problem, pre-processors should prefix the **id** attribute with the **organization id** (as described in the following section) followed by a comma. For example, if the **organization id** value were 'USA,93699' and the **person id** were '111111', the actual value of the **person id** would be 'USA,93699,111111'.

2.2.2. Organizations

AP 209 ed2 represents groups of people (e.g., companies, countries, etc.) through the **organization** entity. The identification or **id** data is optional. This information can be highly important in providing unique identification to the organization or company. It is recommended that this field always be populated with unique data. The **name** attribute must contain a short identifier or acronym for the **organization**. The **description** attribute may contain the full name of the organization or a textual explanation its reason for existence.

Pre-processor Recommendations: All pre-processors should provide a unique **organization id** to eliminate ambiguities where organizations may have the same names. If the intended domain for the data is large, the reader is referred to ISO/IEC 8824-1, which can provide some guidance on creating unique identifiers. A unique string obtained under ISO/IEC 8824-1 can be used as or prefixed to the organization identifier. For example, if the organization typically used an identifier of '93699' and the unique string were 'USA', the actual value of the **organization id** would be 'USA,93699'.

Post-processor Recommendations: All post-processors should make use of any provided information in the **id** attribute to eliminate ambiguities where **organization**s may have the same name.

2.2.3. Roles

The connection of people to organizations is accomplished through the **person_and_organization** entity. It is used to identify approvers for different aspects of the product data. It is also related to certain constructs to identify the people and organizations responsible for them and how they are responsible. This is done through the **applied_person_and_organization_assignment** entity that relates a person and organization in some role to an entity. The role is established in the **person_and_organization_role** entity **name** attribute. The sections that describe the use of the entity that the **person_and_organization is_assignedto** will identify the allowed values for the **name** attribute of the **person_and_organization_role** entity. The **applied_organization_assignment** and **organization_role** entities shall be used to relate the organization in some role to an entity if person and organization exist independently.

2.3. Dates and Times

AP 209 ed2 represents dates and times to record when something occurred. In industry today, this is normally done with a date. AP 209 ed2 allows the use of either a date, or both a date and a time for all events.

2.3.1. Dates

AP 209 ed2 provides three different ways to represent a **date**. All of these are documented in ISO 10303-41. This may require multiple conversions of the data depending on the date type received and the date type used by the organization.

Pre-processor Recommendations: It is recommended that pre-processors use **calendar_date** for date data.

Post-processor Recommendations: Post-processors must be able to process all forms of **date** in AP 209 ed2.

2.3.2. Time

AP 209 ed2 represents time through the entity **local_time**. The **local_time** entity references a time zone identification through the **zone** attribute. The referred to **coordinated_universal_time_offset** entity identifies the delta from the current time zone to coordinated universal time. For AP 209 ed2's application domain, this should be considered the delta in hours and minutes between Greenwich Mean Time (GMT) and the local time zone.

> NOTE - Coordinated Universal Time is NOT exactly Greenwich Mean Time (GMT). The hour and minute offset is the same, but the second offset varies due to seasonal variations in the earth's axis orientation. The difference between GMT and coordinated universal time is on the order of .05 seconds, which has essentially no effect in a configuration management (AP 209 ed2) application.

Pre-processor Recommendations: All pre-processors should use noon in the originating time zone as a default for **local_time** when this data is unavailable. All pre-processors should view Greenwich Mean Time and coordinated universal time as equal.

2.3.3. Roles

The connection of dates to times is accomplished through the **date_and_time** entity. It is used to identify when approval occurred for different aspects of the product data. It is also related to certain constructs to identify the date and time something started, stopped or occurred and what started, stopped or occurred. This is done through the **applied_date_and_time_assignment** entity that relates a date and time in some role to a construct. The role is established in the **date_time_role** entity **name** attribute. The sections that describe the use of the entity to which the **date_and_time** is assigned will identify the allowed values for the **name** attribute of the **date_time_role** entity. If only date is provided, the **applied_date_assignment** and **date_role** entities are used to relate the date in some role to a construct. The **restrict_date_time_role** rule constrains the role established in the **date_time_role** entity **name** attribute.

2.4. Approvals

Approvals in AP 209 ed2 are optional, and are accomplished by establishing an **approval** entity and relating it to some construct through an **applied_approval_assignment** entity. There are rules related to the use of the **approval** entity that require it to have an associated **approval_person_organization** and **approval_date_time**. Help on the creation of these entities is given in Sections 2.2 and 2.3.

Every construct that has an optional **approval** is allowed only one **approval**. This might lead to the misconception that only one person on one date or time can approve something. This is not the case. The approval constructs in AP 209 ed2 actually designate that an approval cycle is required. The cycle may need one or more signatures. This explains the need for the **approval_status** entity and the fact that it allows for a status of **`'not yet approved'`**.

The **approval_date_time** records the date or time the status was changed. It does <u>not necessarily</u> record when the approval was given by the **approval_person_organization,** as there can be multiple **approval_person_organization**s related to an **approval** entity. If there is only one **approval_person_organization** and the **approval_status** is **`'approved'`**, the **approval_date_time** indicates that this person or organization approved it on this date or time. When an approval event is a cycle which requires multiple people to concur on possibly differing dates or times, the dates or times are recorded through the relation of an **applied_date_and_time_assignment** or **applied_date_assignment** entity with the

date_time_role or **date_role** being `'sign off date'`. This relation is explained in Section 2.3, but is not required in the AP. In the cycle case, the **approval_date_time** only indicates when the status of the **approval** was last changed.

The **approval_status name** attribute in AP 209 ed2 has a restriction on its possible values. The values shall only be `'approved'`, `'not yet approved'`, `'disapproved'` or `'withdrawn'`.

Pre-processor Recommendations: There is no standard mapping for the **approval level** attribute. Since there is no standard mapping in the AP 209 ed2 application domain for this attribute, it is recommended that this attribute contain a null string as minimal content or any appropriate or mutually-agreed-upon string. It is recommended that the **approval_role** attribute contain the value `'approver'` if no appropriate data (such as why this person or organization is approving) is available. It is recommended that all **approval_person_organization** instances have associated **applied_date_and_time_assignment** or **applied_date_assignment** entities to provide complete clarity.

Post-processor Recommendations: Since there is no standard mapping for the **approval level** attribute, post-processors should not assign any processing significance to this value.

2.5. Security

AP 209 ed2 requires that certain constructs indicate their sensitivity to the owning organization. This is accomplished by establishing the **security_classification** entity and relating it to the construct via the **applied_security_classification_assignment** entity. The classification is given in **security_classification_level name** attribute. AP 209 ed2 allows the values of this attribute to be '`unclassified`', `'classified'`, `'proprietary'`, `'confidential'`, `'secret'`, and `'top secret'`. It should be noted that the value of `'classified'` only indicates that the data is not unclassified. This value is used when an organization has a security classification that does not exactly match any of the other values.

A **security_classification** in AP 209 ed2 may have an approval (see Section 2.4), a related person and organization (see Section 2.2) in the role of `'classification officer'`, and an associated date and time (see Section 2.3) in the role of `'classification date'`. AP 209 ed2 also provides for indication of an expiration date for the classification by relating a date or time in the role of `'declassification date'` (see Section 2.3), but this is not required.

Pre-processor Recommendations: There is no standard mapping for the **security_classification purpose** attribute. It is recommended that this attribute contain a null string as minimal content or any appropriate or mutually-agreed-upon string. There is no standard mapping for the **security_classification name** attribute. It is recommended that this attribute contain a null string as minimal content or any appropriate or mutually-agreed-upon string.

If the **security_classification_level name** attribute is the value `'classified'`, it is recommended that the organization's classification designation be placed in the **security_classification name** attribute. For example, if an organization had a security classification of `'secret restricted'`, the **security_classification_level name** attribute value would have the value `'classified'`, and the **security_classification name** attribute would have the value `'secret restricted'`.

Post-processor Recommendations: There is no standard mapping for the **security_classification purpose** attribute. It is recommended that this attribute contain a null string as minimal content or any appropriate or mutually-agreed-upon string. If the **security_classification_level name** attribute is the value `'classified'`, it is recommended that post-processors regard the **security_classification name** data as the identification of a special or non-standard classification. If the **security_classification_level name** attribute has a value of other than `'classified'`, it is recommended that post-processors not assign any processing significance to the **name** attribute value.

2.6. Units of Measure

AP 209 ed2 provides for a number of units of measure that can be used for quantities or determining the dimensionality of a shape. The units of measure can be subdivided into explicit and context sensitive units of measure. The explicit units supported are the following: **length_unit**, **mass_unit**, **time_unit**, **thermodynamic_temperature_unit**, **amount_of_substance_unit**, **plane_angle_unit**, **solid_angle_unit**, **area_unit**, **volume_unit**, and **ratio_unit**. The context sensitive units are defined through the use of the **context_dependent_unit name** attribute. Each of the units has a related measure. For the explicit units, these are: **length_measure**, **positive_length_measure**, **mass_measure**, **time_measure**, **thermodynamic_ temperature_measure**, **amount_of_substance_measure**, **plane_angle_measure**, **positive_plane_ angle_measure**, **solid_angle_measure**, **area_measure**, **volume_measure**, and **ratio_measure**. For the context sensitive units, these are: **context_dependent_measure**, **count_measure**, **descriptive_measure,** and **numeric_measure**.

2.7. Shape

AP 209 ed2 provides eight types of **shape_representation** that are interoperable with those of AP 242 for shapes of parts. These classes are: geometrically bounded shape models that are represented by **geometrically_bounded_wireframe_shape_representation** and **geometrically_bounded_surface_ shape_representation** entities; wireframe with topology shape models that are represented by **edge_ based_wireframe_shape_representation** and **shell_based_wireframe_shape_representation** entities; manifold surface with topology shape models that are represented by **manifold_surface_shape_ representation** entities; faceted boundary representation shape models that are represented by **faceted_ brep_shape_representation** entities; tessellated shape models are represented by **tessellated_shape_represetation** entities; and advanced boundary representation models that are represented by **advanced_brep_shape_representation** entities.

AP 209 ed2 also provides two other specialized **shape_representation**s to represent the shape of composite constituents: **composite_sheet_representation** for plies and **beveled_sheet_representation** for sandwich cores. A **shape_representation** of points only (**point_representation**) is also included to allow a finite element model to be exchanged without any associated part shape information.

A **shape_representation** must be related to a **representation_context**.

It should be noted that the **name** attribute of **representation_item** was intended to contain an identification or tag for the geometrical and topological entities. It should be noted that tag values from one CAD system are often not compatible with tag values of another system.

This document will not go into detail on **shape_representation**. It will only present clarifications and practices for the different types of shape as appropriate.

Pre-processor Recommendations: Since CAD system internal entity tag values are typically not compatible from one system to another, it is recommended that pre-processors either use the physical file entity number for the **name** attribute value of **representation_item** entities or a null string for minimal content.

2.7.1. Units for Shape

Units for a type of **shape_representation** are defined through the use of a complex instance of **global_unit_assigned_context** and **geometric_representation_context**. When global units are used, units must be defined for **length_unit**, **plane_angle_unit**, and **solid_angle_unit**. The base units for STEP are Standard International (SI) units that are represented through the **named_unit** subtype **si_unit**. All other units (such as English units) are represented as **conversion_based_unit** entities that reference **si_unit**s. Physical file examples for SI and English units can be found in Appendix B.

In addition to the global measurement units described in the prior paragraph, AP 209 ed2 provides for the definition of a global gap tolerance for a shape model through the addition of **global_uncertainty_assigned_context** to the complex **representation_context** instance. This entity defines a set of **uncertainty_measure_with_unit** entities to represent various gap type measurements. Physical file examples for SI and English units of uncertainty can be found in Appendix B.

As a clarification to 10303-42, units on parametric representations are taken from the **global_unit_assigned_context** entity. They are not always degrees as might be extrapolated from reading the text of 10303-42. This is a consideration on choosing the global units for plane angles, as radian units are irrational and potentially unstable.

Pre-processor Recommendations: If a pre-processor uses **global_uncertainty_assigned_context**, it should point to one **uncertainty_measure_with_unit** that should identify a **length_measure**. The value of the **name** attribute shall be **`'closure'`**. The **length_measure** shall contain the value of the largest gap anticipated between elements that should be deemed coincident. Pre-processors should use degree as the unit for **plane_angle_unit** as it is more stable than using a radian unit.

Post-processor Recommendations: Post-processors shall use the **uncertainty_measure_with_unit** value for error checking of the file where an error is a gap in the shape that is larger than the **length_measure** value.

2.7.2. Shape Aspects

Portions of a shape model can be designated as **shape_aspects**. This can be done just for internal model subdivisions or to attach specifications to portions of the shape. The **shape_aspect name** and **description** attributes have no standard mapping. The **product_definitional** flag must be **`'.T.'`** if the portion of the shape identified is on the outer boundary of the shape model. The use of this construct will be dealt with in Sections 2.8.5, 2.8.10, 2.8.3 and 2.8.11.

2.7.3. Boundary Representation Models

This sub-section will not provide detailed information on boundary representation models.

Pre-processor Recommendations: Pre-processors should not use **face_outer_bound** designations on closed periodic surfaces (cylinder, sphere, torus) because this designation is ambiguous.

Post-processor Recommendations: Post-processors should ignore the **face_outer_bound** designations on closed periodic surfaces (cylinder, sphere, torus) because this designation is ambiguous.

2.8. Parts in AP 209 ed2

In order to define a part in AP 209 ed2, three basic AIM entities are used. The **product** entity establishes the part's identification (or part number), name (or nomenclature), and description. The **product_ definition_formation** or **product_definition_formation_with_specified_source** entity identifies its version (or change level). The **product_definition** or **product_definition_with_associated_documents** entity identifies the engineering discipline view that all the data related to it represents (e.g., design engineering, manufacturing engineering, logistics, etc.).

> NOTE - An analysis is defined in AP 209 ed2 by the same three entities. See Section 2.9 for analysis representations in AP 209 ed2.

Using the three entities (and EXPRESS subtypes), the part is identified, revision controlled, and life cycle stage insulated. Figure 1 describes the relationships among the entities needed to define a part in AP 209 ed2 at a high level. These entities and relationships are necessary in order for AP 209 ed2 to support the various configuration control methodologies that affect parts. The reader must remember that configuration control under AP 209 ed2 is a standard for all industry. As such, AP 209 ed2 represents the data (your data) as an abstraction from the way in which your organization actually does business.

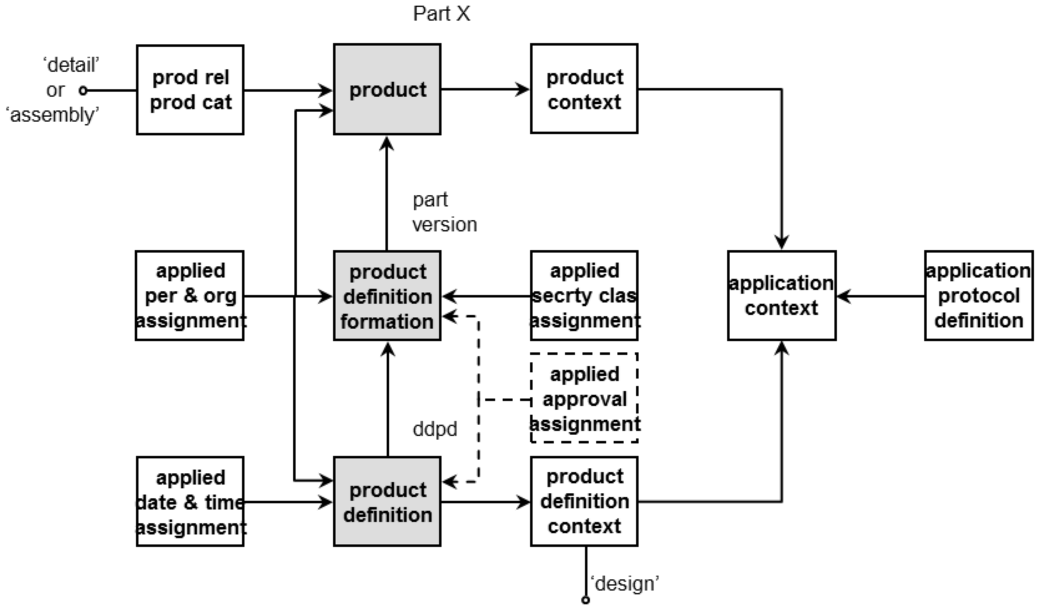


Figure 1 - High Level AP 209 ed2 Requirements for Parts

2.8.1. Identifying Parts

2.8.1.1. The Product Entity

AP 209 ed2 deals with all parts as **product**s. The part number for a part is stored in the **id** attribute. The nomenclature or name of the part is stored in the **name** attribute. If there is an expanded name or description of the part this is stored in the **description** attribute. All STEP **product**s must be founded in some **product_ context** that identifies the engineering discipline from which the data is viewed. See Section 2.1.2 for guidance on creating this entity.

In populating the data, the **id** or part number must be unique. This is typically not a problem when the part is used only within a single company. If the data being assembled is for worldwide consumption, the **id** must be unique in that domain.

AP 209 ed2 requires that all **product**s exist in at least one **product_related_product_category**. This will be addressed in Section 2.8.2.

Part **product**s in AP 209 ed2 require a **person_and_organization** or **organization** in the role of **`'design owner'`**. This designation is applied to the person and organization or authority who originally designed the part. More simply, this is the person and organization typically identified at the top of the title block on the drawing that defines the part. See Section 2.2 for guidance on creating the person and organization entities.

Pre-processor Recommendations: All pre-processors should use non-defaulted data or user input for the values assigned to the design or analysis owner of a **product** as defaulting this data has a high probability of causing the data to be incorrect.

If the data is intended for external usage, the part number should be prefixed with the **organization id** value followed by a comma to ensure uniqueness. For example, if the **organization id** value were 'USA,93699' and the **product id** were '999999', the actual value of the **product id** would be 'USA,93699,999999'.

2.8.1.2. The Product Definition Formation Entity

AP 209 ed2 requires that all part **product**s be associated with a **product_definition_formation** entity. This relation is required as AP 209 ed2 is required to support the versioning of parts and analyses. This rule ensures that all information which typically varies from version to version is always related to the part.

When the **make_or_buy_code** information is specified for a part version, the **product_definition_ formation_with_specified_source** entity should be used.

There are many organizations that claim quite firmly that they do not version parts. Requiring a version establishes a connection which may or may not have valuable data. There are certain things to consider before any group claims that they do or do not version parts.

In AP 209 ed2, the connection being established is actually a connection to the data that comprises the body of a parts list for the part (if it is an assembly). If an organization versions parts, the **product_definition_ formation id** attribute should contain the value that represents this version. The **description** attribute should contain the reason for the creation of the version. The **make_or_buy** attribute in the case of a

product_definition_formation_with_specified_source must contain a value of `'.MADE.'`, `'.BOUGHT.'` or `'.NOT_KNOWN.'`. The value should be `'.MADE.'` if the part is built within the company. The value should be `'.BOUGHT.'` for vendor parts. Discretion on the value is left to the producer of the data with the above guidance given as this distinction can be unclear particularly when the data is exchanged to another party.

AP 209 ed2 requires that all **product_definition_formation** and therefore all **product_definition_formation_with_specified_source** entities be associated with a **person_and_organization** or **organization** in the role of `'creator'`. This person and organization is the one that initiated the release or created the change. The data for this person can be found by looking at the release or change paper work data and finding the initiator. For guidance on creating the entities associated with this data, see Section 2.2.

AP 209 ed2 also requires that all **product_definition_formation** and therefore all **product_definition_formation_with_specified_source** entities be associated with at least one person and organization in the role of either `'design supplier'`, `'analysis supplier'` or `'part supplier'`. For guidance on creating the entities associated with this data, see Section 2.2. The person and organization in the role of `'design supplier'` or `'analysis supplier'` is the one that was the custodian of the master data or analysis when the version was created. The person and organization in the role of `'part supplier'` is the one that had manufacturing cognizance (if the part is made internally to the organization) or the vendor who supplies the part if it is a vendor part.

In AP 209 ed2, part **product_definition_formation** and all **product_definition_formation_with_specified_source** entities may be associated with an **approval**. This is the person and organization that approved the part version. The data for this person can be found by looking at the release or change paper work data and finding who approved the release or change. For guidance on creating the entities associated with an **approval**, see Section 2.4.

AP 209 ed2 requires that all **product_definition_formation** and therefore all **product_definition_formation_with_specified_source** entities be associated with a **security_classification**. For guidance on creating the entities associated with a **security_classification**, see Section 2.5.

Pre-processor Recommendations: If your organization does not version parts, the **id** attribute should contain a null string as minimal data content or any mutually-agreed-upon string. If the **id** attribute was a null string, the **description** value would also be a null string. All pre-processors should use non-defaulted data or user input for the values assigned to the creator, design, analysis and part suppliers, approvers, and approval date for **product_definition_formation** and **product_definition_formation_with_specified_source** entities as defaulting this data has a high probability of causing this data to be incorrect.

The **security_classification** classification officer, classification date, approvers, and approval dates can be extrapolated from the version creator and approval data if no appropriate data is available.

It is recommended that pre-processors use an **id** of `'ANY'` where they wish to indicate a generic revision of a part. This type of instancing would be used when the part with the revision of `'ANY'` is a component in an assembly to indicate that any existing revision of the component is valid for the assembly. This type of instancing reduces the amount of data to be sent in change packages. When this is used, it reduces the ability to track the actual contents of parts lists at a particular change level when the organization versions parts.

Post-processor Recommendations: When the value of the **id** and **description** attributes for **product_ definition_formation_with_specified_source** is a null string, post-processors should use this as an indication that there is no version of the part.

It is recommended that post-processors recognize an **id** of **`'ANY'`** as indicating a generic revision of a part. This type of instancing would be used when the part with the revision of **`'ANY'`** is a component in an assembly to indicate that any existing revision of the component is valid for the assembly.

2.8.1.3. The Product Definition Entity

AP 209 ed2 and STEP use the **product_definition** entity to establish specific life cycle stage views of the product data. The use of **product_definition** entities is not required in AP 209 ed2, but this entity establishes many important relationships such as part-to-part and part-to-shape. If the **product_definition** entity is not used, all that can be done with AP 209 ed2 is identify individual parts with respect to their part number, name and version identification.

It is possible to have many **product_definition**s for a part/version combination. The **id** attribute should identify whose view of the **product** a particular instance represents. There are no standard mappings in the AP for this attribute or the **description** attribute (see pre-processor recommendations).

AP 209 ed2 requires that all **product_definition**s have a **person_and_organization** or **organization** assigned in the role of **`'creator'`**. This person and organization is the one that defined the view. If the **product_definition** is being used as solely a connection to shape, this would be the person who filed the CAD model of the shape. For guidance on creating these constructs, see Section 2.2.

AP 209 ed2 requires that all **product_definition**s have a **date_and_time** or **date** assigned in the role of **`'creation date'`**. This date and time is when the view was defined. If the **product_definition** is being used as solely a connection to shape, this would be the file date and time for the CAD model of the shape. If this is not the case, see the pre-processor recommendations. For guidance on creating the date and time constructs, see Section 2.3.

In AP 209 ed2, **product_definition**s for parts may have an **approval**. This data is often difficult to obtain as those who approved the filing of the CAD or FEA model or creation of the **product_definition** are difficult to identify. If the information is available, see Section 2.4 for guidance on creating the approval.

AP 209 ed2 has an optional feature where a **product_definition** may be related to **document** entities through the subtype **product_definition_with_associated_documents**. In AP 209 ed2 this usage is intended for documents that identify associated Computer Aided Design (CAD) files (where the **document_type** attribute **product_data_type** has the value **`'cad file'`**), analysis model or Computer Aided Engineering (CAE) files (where the **document_type** attribute **product_data_type** has the value **`'cae file'`**), analysis results reports (where the **document_type** attribute **product_data_type** has the value **`'graphical report file'`** or **`'tabular report file'`**), and drawings (where the **document_type** attribute **product_data_type** has the value **`'drawing'`**).

There is no rule in AP 209 ed2 restricting the **product_data_type** of **document**s. AP 209 ed2 does allow for relating specification type **document**s using the **product_definition_with_associated_documents** subtype. This type of relationship indicates that the document is available through the design supplier's organization since it does not provide the **source** data for the specification. See Section 2.8.3 for more detail on creating and referencing specifications.

When using **product_definition_with_associated_documents** to reference CAD files, the **document id** attribute should contain the file name of the file with enough detail so that it is uniquely identified in the exchange. This means that the **id** attribute should identify the source and system (or standard, e.g., IGES, STEP, etc.) together with the file name. A good method for this is to suffix the file name with the CAD file producer's Internet domain name with the unique system name whether or not the system is physically attached to the Internet. For example, if the Internet domain name for the company was ‘widget.com’ and the system name was ‘sparky’ and the file name was ‘mymodel.fle’, the **document id** attribute value would be ‘sparky.widget.com/mymodel.fle’. If the producer does not use the Internet, use the unique organization identification (see Section 2.2 for guidance). The **name** attribute should contain the simple file name. The **description** attribute should contain the description of the CAD file contents complete with that CAD (or other) package used to create the data.

Pre-processor Recommendations: There is no standard mapping for the **id** attribute of **product_definition**. It is recommended that this attribute contain possible values of **`'design'`**, **`'analysis'`**, **`'digital pre-assembly'`**, **`'manufacturing'`**, **`'as built'`**, **`'as maintained'`**. These values should be used to indicate which group owns the view for concurrent engineering purposes within a life cycle stage. There is no standard mapping for the **description** attribute. Since there is no standard mapping in the AP 209 ed2 application domain for this attribute, it is recommended that this attribute contain a null string as minimal content or any appropriate or mutually-agreed-upon string. Where values for the creator and creation date are not readily available, this information can be extrapolated from the creator and approval related to the **product_definition_formation** as defined in Section 2.8.1.2. Pre-processors shall not use **product_definition_with_associated_documents** to relate specification type documents to the **product_definition**.

Pre-processors may use the / character as a delimiter to separate the sending system identification from the actual file name for the **document id** attribute if the receiving system does not have a uniqueness requirement on this value.

Post-processor Recommendations: All post-processors should utilize the values given above for pre-processors as computer sensible segregations of the **product_definition** data based on the **id** attribute. If a value other than those above is received, it should be regarded as **`'design'`**. Since there is no standard mapping for the **description** attribute for the **product_definition** entity (and subtype), it is recommended that post-processors not assign any processing significance to this value.

2.8.2. Categorizing Parts

AP 209 ed2 provides for assigning parts to categories and for creating hierarchical networks of categories. Categories can be extremely useful in adding intelligence to the data. Parts are assigned directly to categories through the **product_related_product_category** entity, which is a subtype of the **product_category** entity. Networks of categories can be created by relating super and subcategories through **product_category_relationship** entity. When a **product_related_product_category** participates in a **product_category_relationship** in the AP 209 ed2 domain, it should always be a subcategory.

Pre-processor Recommendations: There are no standard mappings for the **product_category_relationship name** or **description** attributes. Since there are no standard mappings in the AP 209 ed2 application domain for these attributes, it is recommended that these attributes contain a null string as minimal content or any appropriate or mutually-agreed-upon string. Pre-processors should use lower case for **product_category name** values. Leading and trailing blanks in the **product_category name** value should be removed.

Post-processor Recommendations: Since there are no standard mappings for the **product_category_ relationship name** or **description** attributes, it is recommended that post-processors not assign any processing significance to these values. Post-processors should attempt to store all categories and subcategories and category relationships received in an AP 209 ed2 exchange as this information adds meaning to the received data. If it is impossible to store the data, the user should be informed of all categories and relationships not processed. This would be best done by presenting the user with a report on the category structure in the file with subcategories indented. Post-processors should use non-case sensitive checking when determining matches on processed category data. Leading and trailing blanks in the **product_category name** value should be removed.

2.8.2.1. AP 209 ed2 Standard Categories

In AP 209 ed2 **product**s can be assigned directly. In AP 209 ed2 the **name** attribute of a **product_ related_product_category** may be one of following values: **‘assembly’**, **‘detail’**, **‘customer furnished equipment’**, **‘inseparable assembly’**, **‘standard part’**, **‘linear static analysis’**, **‘linear modes and frequencies analysis’**, **‘anisotropic material’**, **‘cast’**, **‘coined’**, **‘drawn'**, **‘extruded’**, **‘forged’**, **‘formed’**, **‘machined’**, **‘molded’**, **‘rolled’**, **‘sheared’**, **‘composite assembly’**, **‘discontinuous fiber assembly’**, **‘filament assembly’**, **‘filament laminate’**, **‘isotropic material’**, **‘ply’**, **‘ply laminate’**, **‘ply piece’**, **‘processed core’**, or **‘stock core’**.

There are no restrictions in AP 209 ed2 on the value of the **product_category name** attribute when the instance created is at the supertype level. This means that when hierarchical networks of categories are created there is no restriction on the names of the categories which are not directly related to a product.

2.8.2.1.1. Standard Parts

AP 209 ed2 defines via its mapping table a special mapping for standard parts. The defined mapping is that a **product_category** with a **name** attribute value of **‘standard part’** be created and that this category be related through a **product_category_relationship** to a **product_related_product_category** related to a part. The latter should be the subcategory in the **product_category_relationship**.

2.8.2.2. Recommended Categories

It is strongly recommended that all implementations of AP 209 ed2 establish an instance of **product_category** with a **name** attribute value of **`'part'`**. This recommendation is made to facilitate interoperability and allow implementations to defend against other value assignments made in other APs that will result from the reuse of the resource part **product_category** entities.

It is further recommended that all implementations of AP 209 ed2 support the following high level categories that are not standardized in the AP, but will undoubtedly have common usage:

`'commercial'` - This category indicates that the **product** referenced is a general-purpose commercially-available part.

`'customer furnished customer installed'` - This category indicates that the **product** referenced is part of the system or unit for requirements definition, but is actually placed in the system or unit after some portion of delivery.

`'government'` - This category indicates that the **product** referenced is a part that has been developed or purchased to meet specialized government specifications.

`'hazardous material'` - This category indicates that the **product** referenced is (or contains) hazardous material.

`'interchangeable'` - This category indicates that the **product** referenced is a part that requires no trimming or modification when replaced.

`'material'` - This category indicates that the **product** referenced is a material or bulk material.

`'replaceable'` - This category indicates that the **product** referenced is a part that requires trimming or some modification (usually for fit) when replaced.

`'serialized'` - This category indicates that the **product** referenced is (or contains) a serialized part.

2.8.3. Relating Specifications to Parts

AP 209 ed2 relates specifications to entire parts by relating an **applied_document_reference** entity to the **product_definition** of the part. If the specification only relates to a portion of the part, the **applied_document_reference** entity is related to a **shape_aspect,** which is in turn related to the **product_definition_shape** of the part. This relationship is shown in Figure 2. It should be noted that a specification related (through either method) to the **product_definition** of the part must be applicable in every usage.

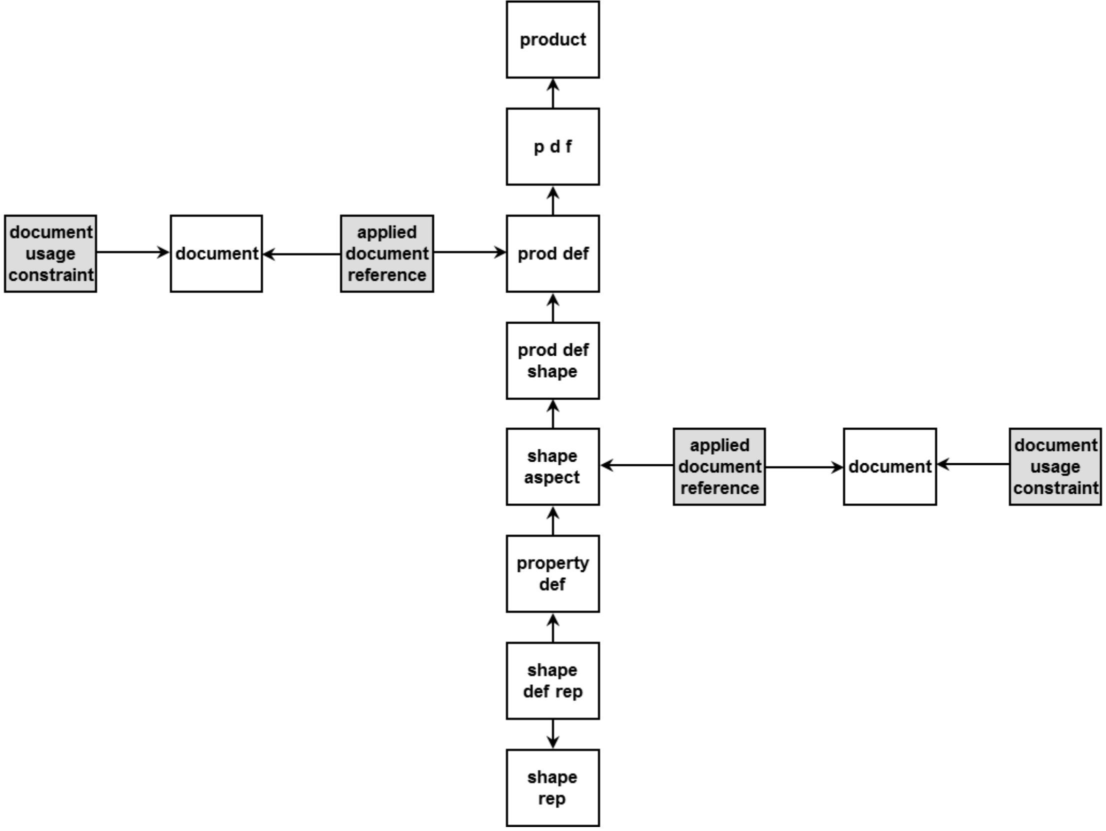

product
p d f
document usage constraint
document
applied document reference
prod def
prod def shape
shape aspect
applied document reference
document
document usage constraint
property def
shape def rep
shape rep

Figure 2 - Parts and Specifications/Shape Aspects

The **applied_document_reference** entity identifies the owner of the specification through the **source** attribute. This attribute should contain an unambiguous identification of where the receiver of the data could obtain a copy of the document. The **document** related to the **applied_document_reference** must be uniquely identified in the exchange by the **id** attribute. This means that the **id** should contain any revision information needed to identify the document completely. The **name** attribute should contain the title of the document. The **description** attribute should contain an expanded explanation of the **document**'s contents.

Since many specifications cover a variety of subtopics and options on a given topic, it may be necessary to identify a particular subtopic of the specification and assign option values. In AP 209 ed2 this is accomplished by relating a **document_usage_constraint** to the document. The **subject_element** attribute identifies the particular section or topic being referenced in the specification. The **subject_element_value** identifies any option choices or restrictions placed on the section or subtopic.

The **document_usage_constraint** should not be used to reference classes defined in specifications such as process specifications. This should be done by using the **document** entity subtype **document_with_class**. If classed documents require further restriction of the class, a **document_usage_constraint** may be related to the **document_with_class** entity.

AP 209 ed2 provides for **document**s related to a **product_definition** to be related to other **document**s in a network type relationship. This is accomplished through the **document_relationship** entity. There are no standard mappings for the **name** and **description** attributes in this entity.

Pre-processor Recommendations: There are no standard mappings for the **name** and **description** attributes in a **document_relationship**. Since there are no standard mappings in the AP 209 ed2 application domain for these attributes, it is recommended that these attributes contain a null string as minimal content or any appropriate or mutually-agreed-upon string.

Post-processor Recommendations: Post-processors should store all data found in specification **document**s attached to **product_definition**s or **shape_aspect**s. If it is not possible to store all the data, the user must be informed of the data being omitted and its relationship to the **product_definition** or **shape_aspect**. Since there are no standard mappings for the **name** and **description** attributes in a **document_relationship**, it is recommended that post-processors not assign any processing significance to these values.

2.8.4. Relating Parts to Contracts

AP 209 ed2 provides an optional relationship of **product**s to contracts through the **applied_contract_ assignment** entity that relates a **contract** to a **product_definition_formation**. In AP 209 ed2, a **contract** can be used to represent either an explicit contract that provides the requirements (and typically the funds) for the creation of designs or analyses for the **product** or some other agreement (such as a purchase order) that fulfills the same function. The **contract name** attribute should contain the contract or agreement identifying number or name if no number exists. The **purpose** attribute should contain the reason for the existence of the contract or agreement. It is recommended that the **contract_type description** contain possible values of **`'fixed price'`** or **`'cost plus'`**.

In AP 209 ed2, a **contract** may have an associated **approval**. For guidance in creating the **approval** constructs, see Section 2.4. A **contract** may have an associated **person_and_organization** or **organization** in the role of **`'contractor'`**. For guidance in creating the person and organization constructs, see Section 2.2. A **contract** may have an associated **date_and_time** in the role of **`'contract date'`**. For guidance in creating the date and time constructs, see Section 2.3.

Pre-processor Recommendations: It may be difficult to obtain the **approval** and **contractor** information. If this information is not available, it should be provided either through user input or from default data based on the **contract** name value.

2.8.5. Relating Shapes to Parts

AP 209 ed2 uses two entities to form the link between the configuration management data for a part and the shape for a part. These two entities are **product_definition_shape** and **shape_definition_representation**. There are no standard mappings for the **product_definition_shape name** and **description** attributes. It should be noted that no link to shape is required. It is possible to use the **product_definition_shape** entity to indicate that a part has (or will have) shape without relating a **shape_definition_representation**.

There must be only one **product_definition_shape** for each **product_definition** in an AP 209 ed2 exchange file. There may be more than one **shape_definition_representation** entity related to the **product_definition_shape**, so as to describe alternate representations of the part shape (Figure 3).

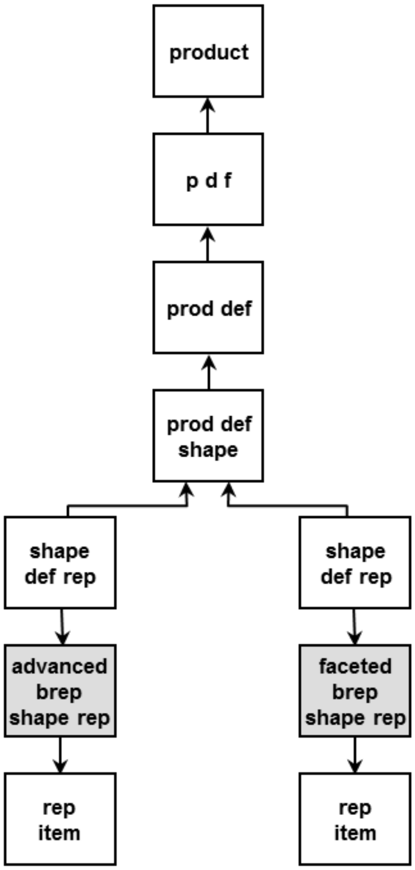


Figure 3 - Part with Alternate Shape Representations

If the shape of the part is composed of shape constructs from multiple types of **shape_representation** to form the entire shape model, the main **shape_representation** shall be related to a **shape_definition_representation** that relates to the **product_definition_shape**. The other **shape_representations** are related to the main **shape_representation** through a **shape_representation_relationship**. This is depicted in Figure 4.

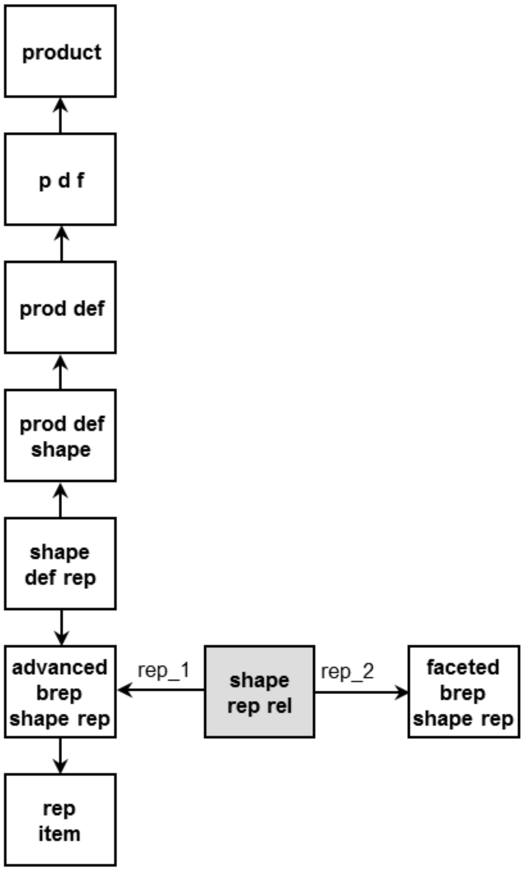


Figure 4 - Single Part Represented with Multiple Shape Representations

In some cases, the shape of a part is based on the shape of another part. This commonly occurs when the one part is the mirror image of the other. When this occurs, it is through a **representation_relationship_ with_transformation**. This structure is shown in Figure 5. The transformation is constructed based on a **functionally_defined_transformation**. It is presumed that the transformation would be applied to the coordinate system of the source part prior to it being mapped to that of the mirrored part.

Pre-processor Recommendations: There are no standard mappings for the **name** and **description** attributes for **product_definition_shape**. Since there are no standard mappings in the AP 209 ed2 application domain for these attributes, it is recommended that these attributes contain a null string as minimal content or any appropriate or mutually-agreed-upon string.

Post-processor Recommendations: Since there are no standard mappings for the **name** and **description** attributes for **product_definition_shape**, it is recommended that post-processors not assign any processing significance to these values.

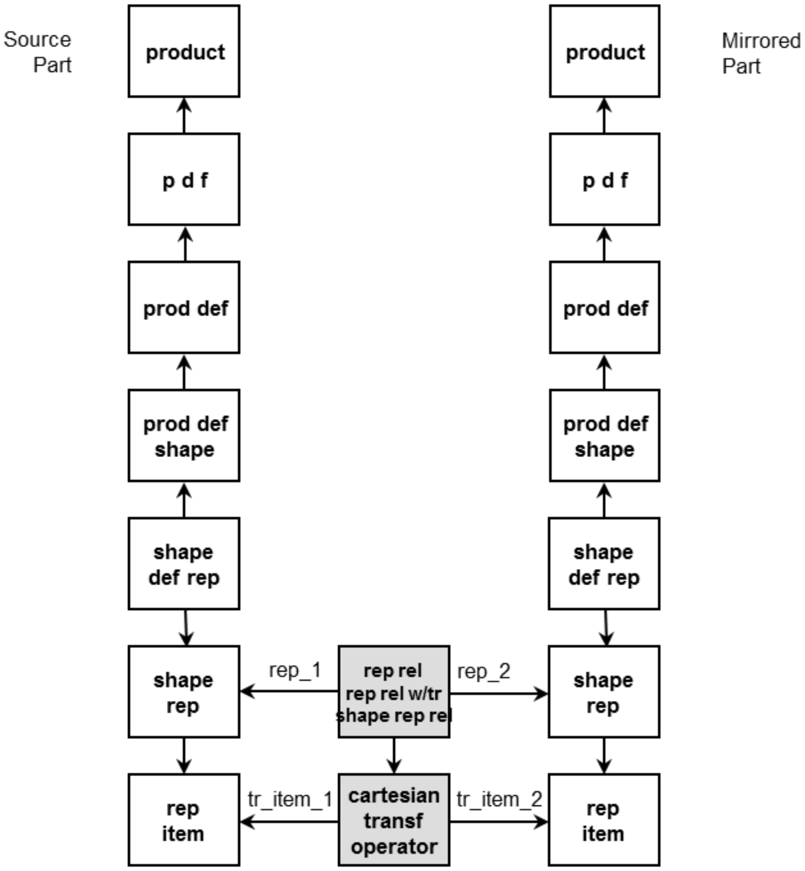


Figure 5 - Part Shape Based on Mirroring

2.8.6. Renumbering Vendor Parts

In all realms of design and manufacturing business, it is common to buy parts from a vendor and renumber them under an internal numbering scheme. In today's practice, this is done through envelope, specification and source control drawings. An envelope drawing is used for a simple renumber of a part where the part is referenced on the envelope drawing and assigned a new part number via the associated parts list. A specification control drawing renumbers a part to show that it meets or exceeds the specifications defined on the drawing and to recommend sources for the part. A source control drawing renumbers a part and creates a restricted list of suppliers that are qualified to produce the part based on the specifications.

In AP 209 ed2, all of the above relationships are supported through the **supplied_part_relationship**. This relationship is used for the identification of **`'part supplier'`**, **`'design supplier'`** and **`'analysis supplier'`**. The identification of **`'design supplier'`** and **`'analysis supplier'`** is actually redundant as this information can be obtained from the **person_and_organization** related to the **product_definition_formation** in the role of **`'design supplier'`** or **`'analysis supplier'`**. This document will only address the use of **supplied_part_relationship**s for renumbering of parts. The structure of a **supplied_part_relationship** is shown in Figure 6.

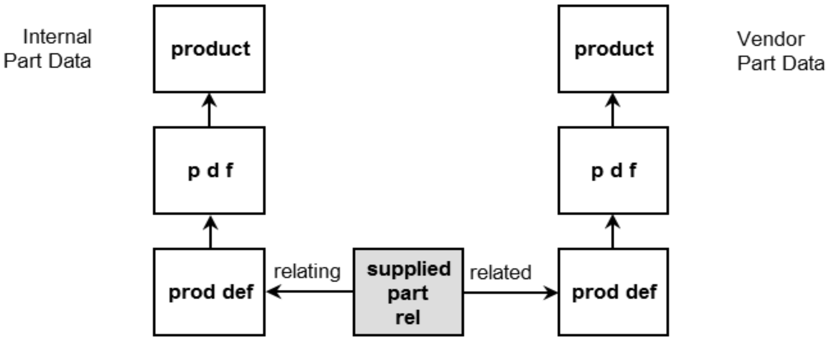


Figure 6 - Supplied Part Relationship

To renumber parts through a **supplied_part_relationship**, both parts must be defined (see Section 2.8.1.1 for guidance on how to create the constructs to do this). The **supplied_part_relationship** relates the 'new' part number's **product_definition** in the **relating_product_definition** attribute to the 'old' part number's **product_definition** in the **related_product_definition** attribute. There are no standard data or mappings for the **name** and **description** attributes. The **id** attribute must be unique, but again there is no standard mapping.

Certification of suppliers can be indicated through a **supplied_part_relationship**. This is accomplished by relating an **applied_certification_assignment** to the **supplied_part_relationship,** which relates a **certification** to the relationship. There are no standard mappings for the values of the **name** and **purpose** attributes for the **certification** entity. It is recommended that the **certification_type description** contain possible values of **`'part supplier'`**, **`'design supplier'`**, or **`'analysis supplier'`**.

If a **certification** is used, the **certification** may be related to an **approval**. See Section 2.4 for guidance on creating the **approval** and related entities. The **certification may** be associated with a **date_and_time** in the role of **`'certification date'`**. See Section 2.3 for guidance on creating dates and time in AP 209 ed2.

> NOTE - This **supplied_part_relationship** is a type of **product_definition_relationship** and as such may have specifications related to it. The relationship of specifications to **product_definition_relationship**s is explained in Section 2.8.11 for the specific case of a **next_assembly_usage_occurrence**.

Pre-processor Recommendations: It may be difficult to obtain the data for the **certification**'s approval and **`'certification date'`**. Where this data is not immediately available, it can be extrapolated from the **approval** related to the **product_definition_formation** found on the path referenced by the **relating_ product_definition** attribute.

There are no standard mappings for the **name** and **description** attributes in a **supplied_part_relationship**. Since there are no standard mappings in the AP 209 ed2 application domain for these attributes, it is recommended that these attributes contain a null string as minimal content or any appropriate or mutually-agreed-upon string. The **id** attribute must be constructed so as not to duplicate any assignments made to other entities that are subtypes of **product_definition_relationship.**

There are no standard mappings for the **name** and **purpose** attributes in a **certification**. Since there are no standard mappings in the AP 209 ed2 application domain for these attributes, it is recommended that these attributes contain a null string as minimal content or any appropriate or mutually-agreed-upon string.

Post-processor Recommendations: Since there are no standard mappings for the **name** and **description** attributes for a **supplied_part_relationship**, it is recommended that post-processors not assign any processing significance to these values.

Since there are no standard mappings for the **name** and **purpose** attributes for a **certification**, it is recommended that post-processors not assign any processing significance to these values.

2.8.7. Alternate Parts

AP 209 ed2 designates alternate and substitute parts differently. Alternate parts are interchangeable in all occurrences whereas substitutes (see Section 2.8.9.3) are interchangeable only in a particular usage. Alternate parts in AP 209 ed2 are defined through the **alternate_product_relationship** entity. This relationship is used in the definition of parts list data for alternate item designations. There are no standard mappings to the **name** and **description** attributes of this entity. The **basis** attribute should contain a rationale for the interchange (e.g., any use, first available, etc.).

Pre-processor Recommendations: There are no standard mappings for the **name** and **description** attributes of **alternate_product_relationship**. Since there are no standard mappings in the AP 209 ed2 application domain for these attributes, it is recommended that these attributes contain a null string as minimal content or any appropriate or mutually-agreed-upon string.

Post-processor Recommendations: Since there are no standard mappings for the **name** and **description** attributes of **alternate_product_relationship**, it is recommended that post-processors not assign any processing significance to these values.

2.8.8. Make From Relationships

In AP 209 ed2, the fact that a part or the design for a part is made from another part or the design for another part is indicated by the **design_make_from_relationship**. To indicate either of the above, both parts must be defined (see Section 2.8.1.1 for guidance on how to create the constructs to do this). The **design_make_from_relationship** relates the source part number's **product_definition** in the **relating_product_definition** attribute to the resultant part number's **product_definition** in the **related_product_definition** attribute. The **id** attribute must be unique, but there is no standard mapping. There is no standard mapping for the **name** attribute. The **description** attribute should be set to whatever data is shown on the parts list of the resultant part as its material specification, if any. The **design_make_from_relationship** is shown in Figure 7.

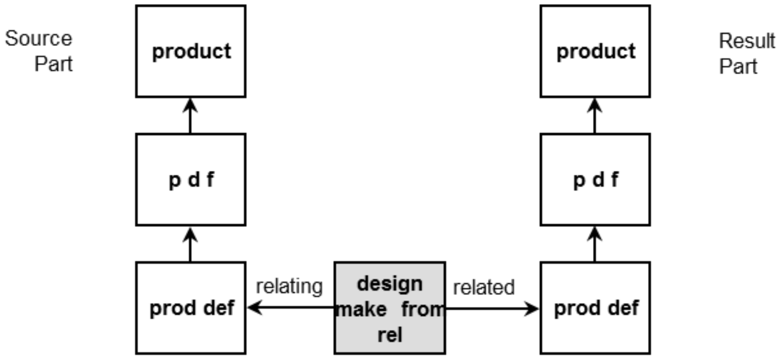


Figure 7 - Design Make From Relationship

NOTE - This relationship is a type of **product_definition_relationship** and as such may have specifications related to it. The relationship of specifications to **product_definition_relationship**s is explained in Section 2.8.11 for the specific case of a **next_assembly_usage_occurrence**.

Pre-processor Recommendations: There is no standard value for the **name** attribute in a **design_make_from_relationship**. Since there is no standard mapping in the AP 209 ed2 application domain for this attribute, it is recommended that this attribute contain a null string as minimal content or any appropriate or mutually-agreed-upon string. The **id** attribute must be constructed so as not to duplicate any assignments made to other entities that are subtypes of **product_definition_relationship.**

Post-processor Recommendations: Since there is no standard value for the **name** attribute for a **design_make_from_relationship**, it is recommended that post-processors not assign any processing significance to this value.

2.8.9. Assembling Parts

In AP 209 ed2, assemblies are defined in the same way an individual part is defined. The major difference is that an assembly has other related parts. These are the detail parts and sub-assemblies that comprise the assembly. This relationship of an assembly part to its components is defined through a **next_assembly_usage_occurrence** in AP 209 ed2. The structure of this relationship is shown in Figure 8.

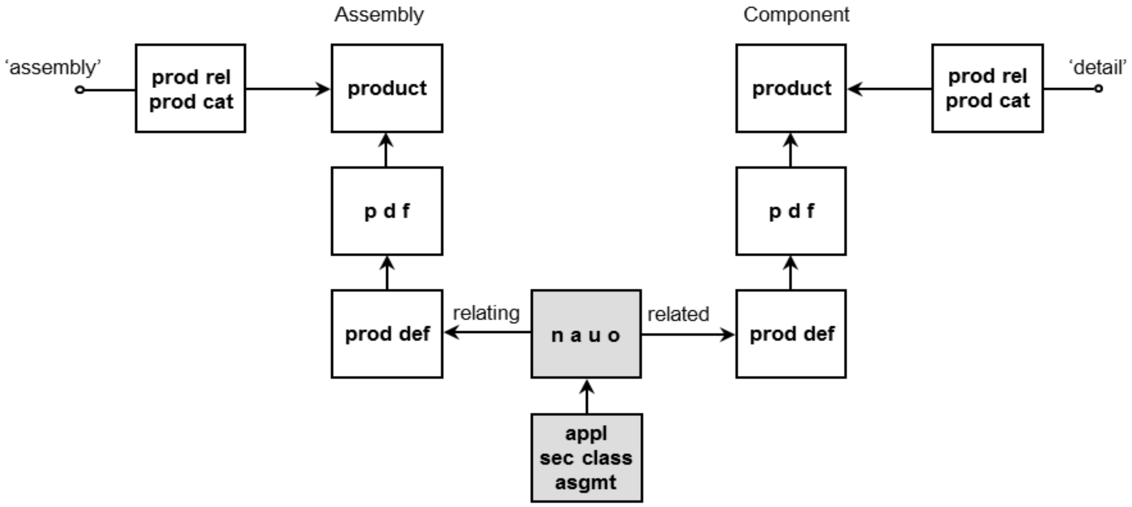


Figure 8 - Assembly/Component Relationship

> NOTE - Processors may use a version **id** of **`'ANY'`** where they wish to indicate a generic revision of a part when the part is a component in an assembly. This indicates that any existing revision of the component is valid for the assembly. This type of instancing reduces the amount of data to be sent in change packages. When this is used, it reduces the ability to track the actual contents of parts lists at a particular change level when the organization versions parts.

The **next_assembly_usage_occurrence id** attribute has no standard mapping, but must be unique for all instances of the entity. The **name** attribute should contain the item/find number from the parts list that identifies the usage. It is recommended that the **description** attribute contain an instance identifier for the usage, if one exists or contain a null string as minimal content. The **reference_designator** attribute is optional and when present designates a unique positional location.

It should be noted that since the usage is described by a **product_definition_relationship**, many different views of the usage can be established by varying the **relating_product_definition**. AP 209 ed2 can maintain one usage based on the **`'design'`** **product_definition** and another based on the **`'manufacturing'`** **product_definition**. The various **product_definition**s can move into other life cycle stages for the product as well. In this way, usages or parts lists can be defined for any of a number of views and life cycle stages of a design. See Section 2.8.1.3 for recommended values of the **product_definition id** attribute to support concurrent engineering during a particular life cycle stage.

AP 209 ed2 requires that all **product_definition_usage** entities and therefore all **assembly_component_usage** and **next_assembly_usage_occurrence** entities be associated with a **security_classification**. For guidance on creating the entities associated with a **security_classification**, see Section 2.5.

Pre-processor Recommendations: The **id** attribute of the **next_assembly_usage_occurrence** must be constructed so as not to duplicate any assignments made to other entities which are subtypes of **product_definition_relationship.**

The **security_classification** classification officer, classification date, approvers and approval dates can be extrapolated from the version creator and approval data for the assembly part if no appropriate data is available.

2.8.9.1. Instances in Multi-Level Assemblies

AP 209 ed2 provides ability to identify individual occurrences of component in an multi-level assembly. This provides the ability to assign to each occurrence an identifier, a position in the assembly and, possibly, a geometrical representation that may be different from the one assigned to the product_definition of the component (for example, if a component is a flexible pipe, the geometrical representation assigned to the **product_definition** of the component may be a I shaped tube while the geometrical representation assigned to the occurrence of pipe positioned in the assembly may be an S shape constrained by the environment).

In order to distinguish the occurrences of component in an assembly of more than two hierarchical levels, the **specified_higher_usage_occurrence** entity is used. For example, in the case of the table, an instance of **specified_higher_usage_occurrence** with **description** attribute set to 'Cap1' will correspond to the occurrence 1 of the cap. The **upper_usage** attribute of the **specified_higher_usage_occurrence** will identify that 'Cap1' is a component mounted on the leg occurrence #1 that is used on the table (see Figure 9 and Figure 10).

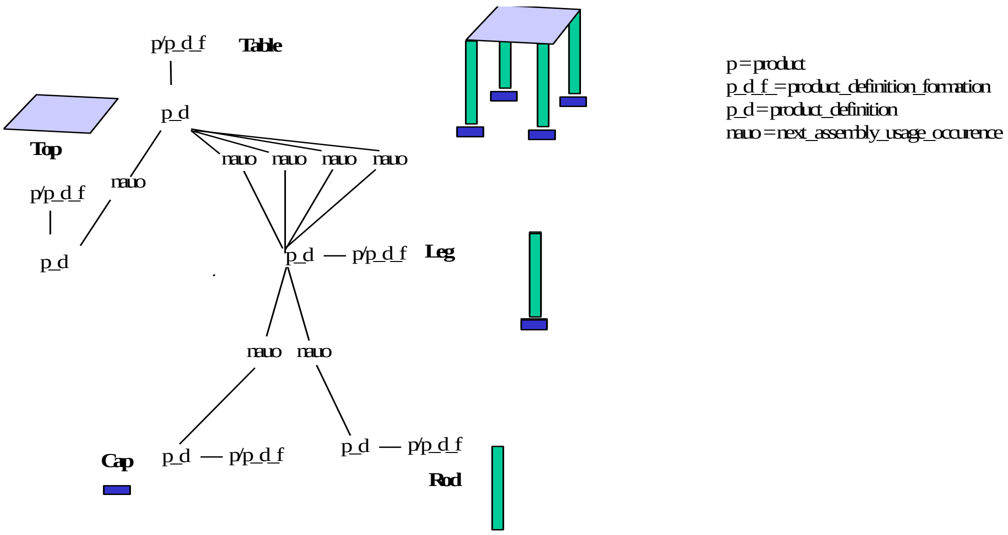


Figure 9 - Table Assembly Example

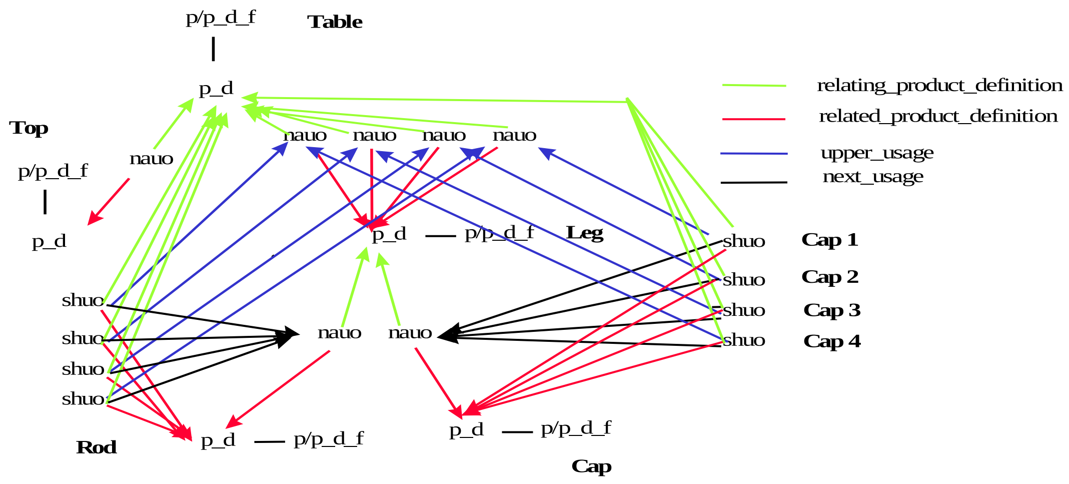


Figure 10 - Specified higher usage occurrence and instancing

NOTE - These figures and practice are courtesy of GOSET.

2.8.9.2. Quantities in Assemblies

AP 209 ed2 provides for designating quantities of components in next assemblies and higher assemblies. The most common types of quantities are next assembly quantity and quantity for an end item. A next assembly quantity is the amount (count or other measure) of a part in its immediate parent part. The quantity for an end item is the amount (count or other measure) of a part in a finished manufactured item. The end item itself is designated by the organization and may be a configuration item (see Section 2.8.14). These two types of quantity and their related data is typically what comprises the body of an application list.

2.8.9.2.1. Next Assembly Quantity

AP 209 ed2 provides two methods for specifying next assembly quantity. One method is to count the number of **next_assembly_usage_occurrence**s where the pair of the **relating_product_definition** and **related_product_definition** attributes are identical among multiple instances of the **next_assembly_usage_occurrence** entity. This type of quantity specification can only be used for items that are counted one piece at a time as there can be no unit of measure attached to this type of quantity. This method is extremely valuable where all instances of a component are specified geometrically as well as in the product structure.

The other method of specifying next assembly quantity in AP 209 ed2 is to create a complex instance of **next_assembly_usage_occurrence** and **quantified_assembly_component_usage**. The quantity is explicitly stated in the **measure_with_unit** related to the **quantified_assembly_component_usage**.

NOTE - Since these constructs are subtypes of **assembly_component_usage**, they will require a **security_classification**. See Section 2.5 for guidance on creating these constructs and Section 2.8.9 for processor recommendations.

Quantity designations are used on parts lists for **product**s. The AP 209 ed2 data structure is quite capable of providing the data for the body of a parts list. The information for each record in this list is generated for an assembly by obtaining the data for the **product**s related to it through **next_assembly_usage_ occurrence**s. For a make from part, the same rationale is applied to the **make_from_usage_occurrence** with the resultant part from the make from also being called out. For a material callout, the parts list is determined from the **material_specification**s related to its **product_definition** unless the bulk material is assigned a part number internally by the organization or a quantity unit of measure other than a simple count is needed. If a bulk material is assigned an internal part number by an organization or a unit of measure other than a simple count is needed, the usage of the material becomes a **next_assembly_usage_ occurrence** between the two.

2.8.9.2.2. End Item Quantity

End Item Quantity is the total quantity of a component in either the entire delivered unit or some major subsection of a delivered unit. This quantity is designated in AP 209 ed2 by establishing a complex instance of **promissory_usage_occurrence** and **quantified_assembly_component_usage**. The quantity in the **measure_with_unit** related to the **quantified_assembly_component_usage** is the quantity of the part in the final article. This relationship is described in Figure 11.

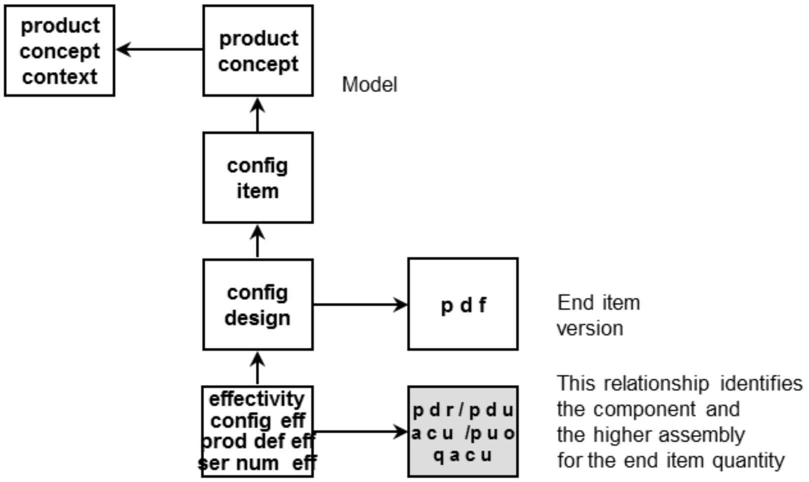


Figure 11 - End Item Quantity

It should be noted that this could be a simple direct relationship or a more complex relationship. In the simple instance, the **relating_product_definition** will point to the **product_definition** of the **product** that is designated as the end item. In this case, the quantity is the total for the component (specified by the **related_product_definition**) in the end item for the indicated effectivity. In the more complex instance, the **relating_product_definition** will point to a **product_definition** of a higher assembly that is not the end item. In this case, the quantity is for the component (specified by the **related_product_definition**) in the assembly (specified by the **relating_product_definition**) as the assembly is used in the end item for the indicated effectivity.

Since **promissory_usage_occurrence** is a subtype of **assembly_component_usage**, it will require a **security_classification**. See Section 2.5 for guidance on creating the **security_classification** constructs and Section 2.8.9 for processor recommendations for subtypes of **product_definition_relationship**. This relationship is a type of **product_definition_relationship** and as such may have specifications related to it. The relationship of specifications to **product_definition_relationship**s is explained in Section 2.8.11 for the specific case of a **next_assembly_usage_occurrence**.

2.8.9.3. Substituting Parts in Assemblies

AP 209 ed2 designates alternate and substitute parts differently. Alternate parts (described in Section 2.8.7) are interchangeable in all occurrences where as substitutes are interchangeable only in a particular usage. A substitute part is designated through the use of an **assembly_component_usage_substitute**. This relationship is used to define information that is represented on a parts list. There are no standard mappings for the **name** and **description** attributes in an **assembly_component_usage_substitute**. The **base** attribute points to the normal or preferred usage. The **substitute** attribute points to the surrogate usage.

Pre-processor Recommendations: There are no standard mappings for the **name** and **description** attributes in an **assembly_component_usage_substitute**. Since there are no standard mappings in the AP 209 ed2 application domain for these attributes, it is recommended that these attributes contain a null string as minimal content or any appropriate or mutually-agreed-upon string.

Post-processor Recommendations: Since there are no standard mappings for the **name** and **description** attributes in an **assembly_component_usage_substitute**, it is recommended that post-processors not assign any processing significance to these values.

2.8.10. Assemblies and Shape

The shape of an assembly is most often derived from the shape of its components. AP 209 ed2 provides many methods for dealing with the shape of an assembly, composite structures, and with the shapes of finite element models.

This information has been removed from this document and put into the document: Geometric Founding in ISO 10303-209 RevC.

2.8.11. Assemblies and Specifications

The reference of a specification to an assembly itself without respect to any particular component in the assembly is done in the same manner it is for parts (see Section 2.8.3). An assembly is peculiar since specifications may need to be related to the usage of a particular component in an assembly or the interface between the component and the assembly. AP 209 ed2 relates specifications to assembly-component relationships by relating a **applied_document_reference** entity to a **shape_aspect** that references a **product_definition_shape** that is related to the **next_assembly_usage_occurrence** of the part. The shape related to the **shape_aspect** describes the actual area of application for the specification. This relationship is shown in Figure 12.

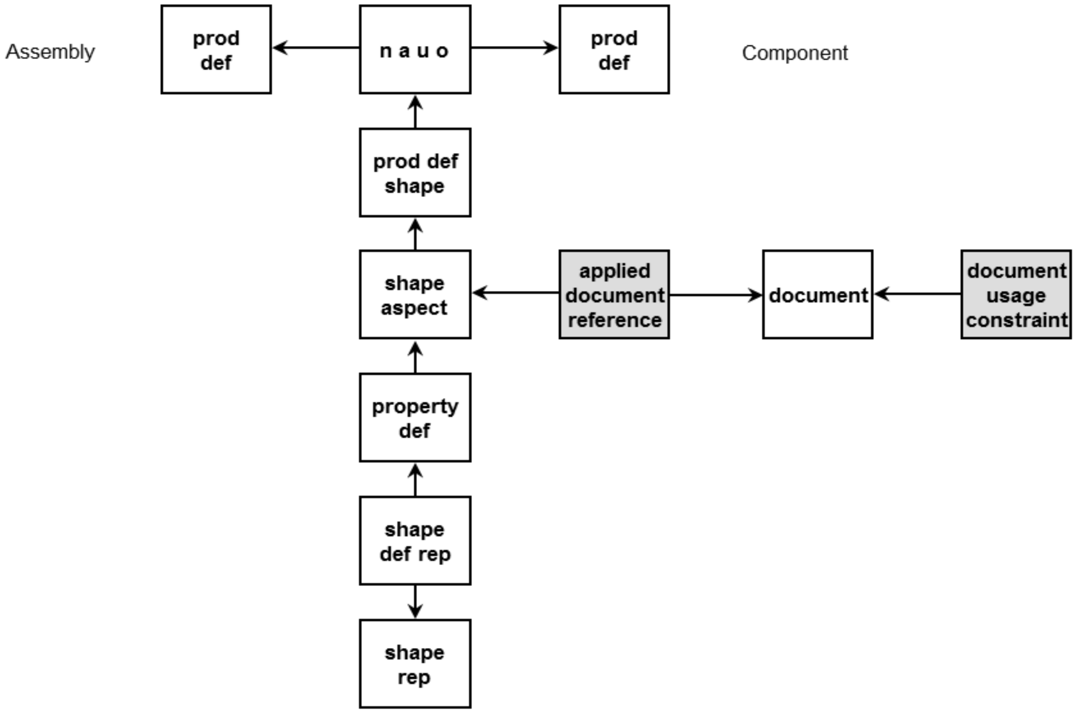


Figure 12 - Assemblies and Specifications/Shape Aspects

It should be noted that in an AP 209 ed2 implementation there is no shape data to specify the actual area to which the specification is applicable. In this type of implementation, a **document_usage_constraint** should be used to clarify the application of the specification. This entity is explained later in this section.

The **applied_document_reference** entity identifies the owner of the specification through the **source** attribute. This attribute should contain an unambiguous identification of where the receiver of the data could obtain a copy of the document. The **document** related to the **applied_document_reference** must be uniquely identified in the exchange by the **id** attribute. This means that the **id** should contain any revision information needed to identify the document completely. The **name** attribute should contain the title of the document. The **description** attribute should contain an expanded explanation of the **document**'s contents.

Since many specifications cover a variety of subtopics and options on a given topic, it may be necessary to identify a particular subtopic of the specification and assign option values. In AP 209 ed2 this is accomplished by relating a **document_usage_constraint** to the document. The **subject_element** attribute identifies the particular section or topic being referenced in the specification. The **subject_element_value** identifies any option choices or restrictions placed on the section or subtopic.

The above should not be used to reference classes defined in specifications such as process specifications. This should be done by using the **document** entity subtype **document_with_class**. If classed documents require further restriction of the class, a **document_usage_constraint** may be related to the **document_with_class** entity.

Post-processor Recommendations: Post-processors should store all data found in specification **document**s attached to **shape_aspect**s. If it is not possible to store all the data, the user must be informed of the data being omitted and its relationship to the **shape_aspect**.

2.8.12. Engineering Release/Change Data - Work Requests and Work Orders

AP 209 ed2 provides data structures for representation of the data used in the engineering release and change process. The structures are based on a request and action process where a request is established documenting the need for a potential release or change that may or may not ever be incorporated. If the request is incorporated, it is done through some action being taken on the request, which results in either a new release of a design or a change to a existing design.

It should be noted that these constructs have been designed to represent all request and incorporation structures in the AP 209 ed2 application domain. All release and change proposals and requests (Engineering Change Proposals, Requests for Engineering Action, etc.) are represented by the request portion of the structure. All release and change incorporations are represented by the action portion of the structure. Differentiation between types of requests and actions can be done structurally based on the guidance in this section, by its identification (**id** for requests, **name** for actions), or by the originator. Differentiation by identification or originator is very process dependent but can be necessary particularly for preliminary requests and proposals.

Some types of releases and changes in organizations may not involve a two step process. In this case, both data structures are implemented simultaneously and reference the same release or change documentation. Since these constructs in AP 209 ed2 are intended to support many different release and change processes/documentation, in some cases, some of the required data may not exist.

In AP 209 ed2, the release process is initiated through a **versioned_action_request** (as a start request) that is related to the design or analysis being released through an **applied_action_request_assignment**. The **versioned_action_request** has a related **action_method**. In this case, both the **versioned_action_request** and the **action_method** would indicate that the respective **purpose**s were to initially release the design or analysis, or create the design or analysis for the initial release. This request process is followed (in the data) by an **action_directive** (as a start order). The **action_directive** identifies the **versioned_action_request** as the request being satisfied or incorporated. A **directed_action** relates the **action_directive** to the new design or analysis to be released through an **applied_action_assignment**. The **directed_action** also relates the **action_method** to the **action_directive,** which may be moot in the case of initial release. The high-level structure of these relationships is shown in Figure 13.

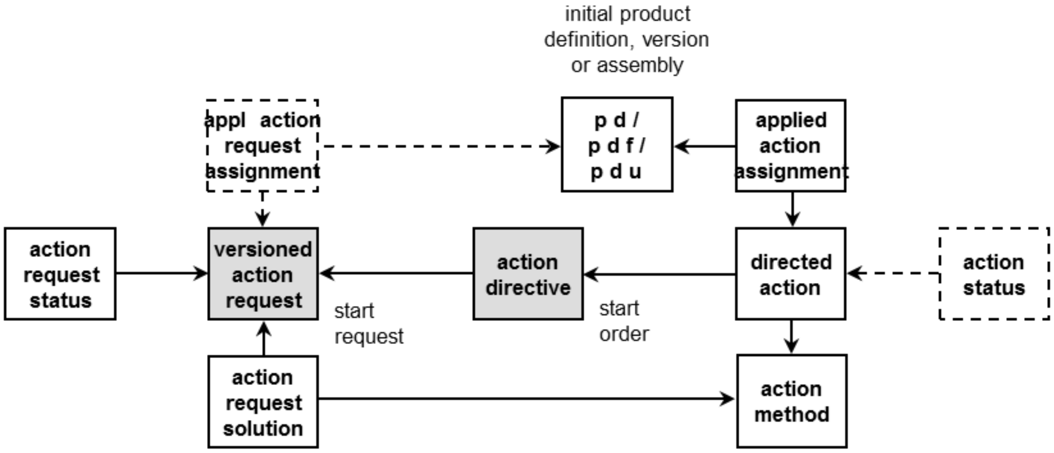


Figure 13 - AP 209 ed2 Engineering Release Process

The change process is initiated through a **versioned_action_request** (as a change request), which is related to the design or analysis product definition or version proposed to be changed through an **applied_action_request_assignment**. The **versioned_action_request** has a related **action_method**. In this case, there may be many **action_method**s or ways to solve the problem. This request process is followed (in the data) by an **action_directive** (as a change order) The **action_directive** identifies the **versioned_action_request**(s) as the request(s) being satisfied or incorporated. A **directed_action** relates the **action_directive** to the new design or analysis product definition or version to be released through an **applied_action_assignment**. The **directed_action** also relates the **action_method** to the **action_directive** indicating which of possibly many methods for the request or requests incorporated was chosen. The structure of these relationships (at a high level) is shown in Figure 14.

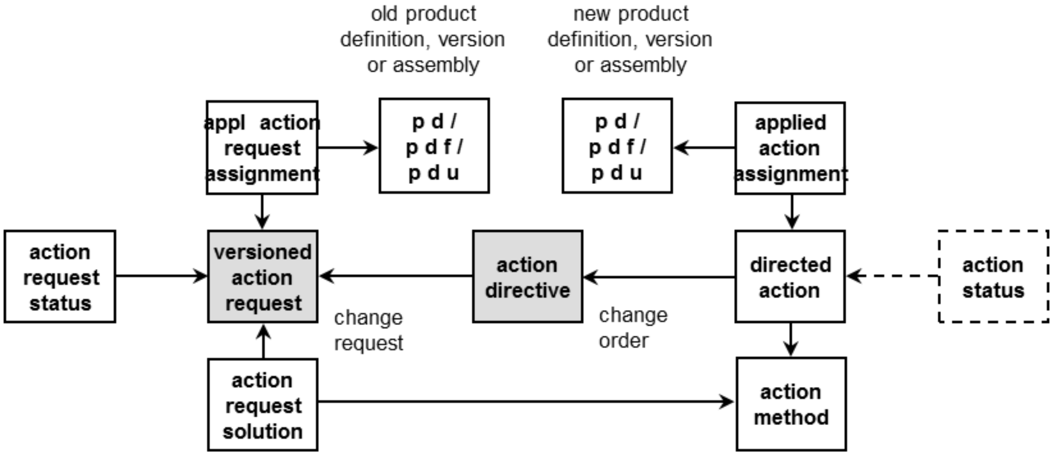


Figure 14 - AP 209 ed2 Engineering Change Process

2.8.12.1. Work Requests for Release/Change

Requests for release or change are created in AP 209 ed2 by relating a **versioned_action_request** (as a start or change request) to a **product_definition**, **product_definition_formation**, or **product_definition_relationship** through an **applied_action_assignment_request**. The **applied_action_assignment_request** identifies through the **items** attribute the **product_definition**, **product_definition_formation**, or **product_definition_relationship** to be released or changed. In the case of a start request, AP 209 ed2 does not require an identification of a product definition or version at request time. In fact, the product definition or version will result from the start request. The **version** attribute of a **versioned_action_request** that is a start request should be set to 'initial' in order to differentiate it from a **versioned_action_request** that is a change request.

The **versioned_action_request id** attribute contains the identification of the request. This information is the document or request number. The **version** attribute is the version of the request itself. This attribute is used to identify actual versioning of the request or reissues of the request. In the case of a start request, the **version** attribute should be set to **`'initial'`** as discussed above. The **purpose** attribute should contain text identifying the end result anticipated from this version of this request. The **description** attribute should contain a general description of the request. In AP 209 ed2, a **versioned_action_request** is required to have an associated **action_request_status**. The AP restricts the values for the **status** attribute to **`'proposed'`**, **`'in work'`**, **`'issued'`**, or **`'hold'`**.

A request for release or change may have many possible ways it can be resolved. This is more common for changes than releases, but the AP 209 ed2 data structure supports the documentation of the engineering thought process gone through in either case. This is accomplished through a combination of the **action_request_solution** and **action_method** entities. The **action_request_solution** entity relates an **action_method** to a **versioned_action_request**. The **action_method name** attribute should contain a reference to any formal documentation for a proposed solution to the release/ change request. The **description** attribute should contain a detailed description of the method through which the request is to be satisfied. The **consequence** attribute should contain any determined or perceived consequence to using this method to satisfy this request. The **purpose** attribute should contain the intention of the method as a single method may be used to satisfy many requests.

In AP 209 ed2, a **versioned_action_request** may have a related **approval**. As these requests normally have a number of signatories, there should be no problem obtaining this data if it is stored in electronic form. For guidance in creating the **approval** constructs, see Section 2.4. A **versioned_action_request** is required to be associated with a date and time in the role of **`'request date'`**, which indicates when the request was created. For guidance in creating the date and time constructs, see Section 2.3. Lastly, a **versioned_ action_request** is required to be associated with at least one person and organization in the role of **`'request recipient'`**. For guidance in creating person and organization constructs, see Section 2.2.

2.8.12.2. Incorporation of Work Orders for Release/Change

Release of a design or analysis, or change incorporation into a design or analysis is accomplished in AP 209 ed2 through the **applied_action_assignment** entity that relates an **action_directive** to the new design or version by pointing to the **product_definition_formation** that results from the release or change. A **directed_action** related to the **action_directive** identifies the **action_method** actually used to satisfy the requests related to the **action_directive**. In the case where many requests are being incorporated, there many be many **directed_action**s to indicate the appropriate methods.

The **action_directive name** attribute is the identification of the formal documentation to incorporate the change or release the design or analysis. In cases where there is no second set of paper work or documentation (i.e., there is a one to one correspondence between **versioned_action_request** and **action_directive**), the **action_directive name** value is the same as the **versioned_action_request id** value. The **description** attribute should contain a phrase or group of phrases indicating the final result of the release or change. The **analysis** attribute should identify any investigative results that support the release or change. Likewise, the **comment** attribute should contain any textual commentary that supports the release or change. An **action_directive** may be associated with an **action_status** that serves the same function as **action_request_status** in the previous request section. AP 209 ed2 does not require that the **action_directive** be related to an **action_status** as the two sets of data may represent one or two documents.

In order to differentiate an **action_directive** that is start order from an **action_directive** that is a change order, the related **directed_action name** attribute should be set to **`'design'`** or **`'analysis'`** for the former, and to **`'design change'`** or **`'analysis change'`** for the latter.

In AP 209 ed2, an **action_directive** may have an associated **approval**. For guidance in creating the **approval** constructs, see Section 2.4. An **action_directive** is required to have a date and time associated with it in the role of **`'start date'`** or **`'change date'`**, which is when the work to satisfy the request or requests began. Once completed, an **action_directive** may have a date and time associated with it in the role of **`'release date'`**. For guidance on creating these date constructs, see Section 2.3.

2.8.13. Release/Change Reissues

Engineering releases and changes may be reissued. This may be done to correct an error or omission in the change package. It may also be done to signify changes in effectivity that have no effect on the version of the part.

AP 209 ed2 supports the reissue of releases and changes. To reissue a release or a change, a **versioned_action_request** is created with an **id** attribute value equal to the **action_directive name** being reissued. The **versioned_action_request version** attribute contains the reissue identifier. This new **versioned_action_request** is added to the set of **requests** in the original **action_directive** that was issued.

2.8.14. Configuration Identification

Configuration identification in AP 209 ed2 is done through the **configuration_item** entity. This entity identifies **product**s as end items or items that are sold or delivered. As in industry, this designation can be applied to full systems or spares (which are also referred to as the lowest level replaceable units).

The **configuration_item id** attribute is a unique identification of the item that may be a part number but more probably a moniker. The **name** attribute is a short description of the item. The **description** attribute is optional and would be the expanded name or description of the item. The **purpose** attribute is also optional and would contain a description of the item's intended use.

A **configuration_item** is related to a **product_concept**. The **product_concept id** attribute is more commonly known as the model designation. The **product_concept** taken together with the **configuration_item** describe a model series or configured production run. The **name** attribute is a short description of the model. The **description** attribute is the expanded name or description of the model. The **product_concept** is related to a **product_concept_context** where the **market_segment_type** attribute identifies what customer or group of customers provided the requirements for the model.

In AP 209 ed2, a **configuration_item** may have associated an **approval**. For guidance on creating the **approval** constructs, see Section 2.4. A **configuration_item** must be associated with a **person_and_organization** in the role of **`'configuration manager'`**. For guidance on creating these constructs, see Section 2.2.

Pre-processor Recommendations: In some cases, it may be difficult to determine the **approval** and **`'configuration manager'`** for a **configuration_item**. If the item has effectivity (see next section), this information may be extrapolated from the **approval** and **`'creator'`** information for the **product_definition_formation** for that **product**. If not, this information should be obtained from user input or a default based on the **configuration_item id** attribute.

2.8.15. Effectivity

Effectivity is the designation that something or a relationship between two things is used or planned to be used in some **configuration_item**. In AP 209 ed2, effectivity is designated on relationships between **product_definition**s by either ranges of serial numbers, ranges of dates or a lot. This is accomplished through a complex instance of the entities **effectivity**, **configuration_effectivity**, **product_definition_effectivity** and one of either **serial_numbered_effectivity**, **dated_effectivity** or **lot_effectivity**.

A **serial_numbered_effectivity** specifies an **effectivity_start_id** with an optional **effectivity_end_id**. If the **effectivity_end_id** does not exist, the effectivity is good for the starting serial number and all following serial numbers. A **dated_effectivity** follows the same pattern using dates rather than serial numbers. A **lot_effectivity** indicates an **effectivity_lot_id** and an **effectivity_lot_size**.

The above entities specify the effectivity identifiers. These entities are related to a **product_definition_relationship** through the **usage** attribute in the **product_definition_effectivity** entity. The **effectivity** entity **id** attribute has no standard mapping. The **configuration_effectivity** entity relates these relationships to a **configuration_design** that relates a **configuration_item** to a **product_definition_formation**. Figure 15 shows this relationship for a **serial_numbered_effectivity**. The whole relationship here can be simply stated as a range of serial numbers, dates or a lot number related to a **product_definition_formation** which is designated as a **configuration_item.** This does mean that all **configuration_items** must be associated to a design or analysis version in order to have **effectivity**.

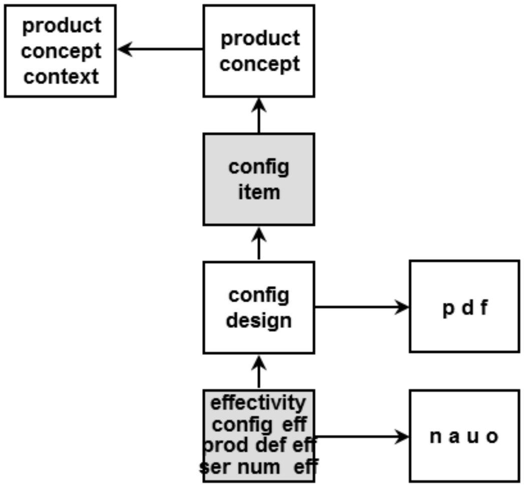


Figure 15 - Configuration Item and Serial Numbered Effectivity

In AP 209 ed2, an **effectivity** may have an associated **approval**. For guidance in creating the **approval** constructs, see Section 2.4.

It should be noted that since the effectivity is related to a **product_definition_relationship**, many different views of the effectivity can be established by varying the **relating_product_definition**. AP 209 ed2 can maintain one effectivity based on the **`'design'`** **product_definition** and another based on the **`'manufacturing'`** **product_definition**. The various **product_definition**s can move into other life cycle stages for the design as well. In this way, effectivities can be defined for any of a number of views and life cycle stages of the design. See Section 2.8.1.3 for recommended values of the **product_definition id** attribute to support concurrent engineering during a particular life cycle stage.

AP 209 ed2 does not require that effectivity relationships be instantiated. The reason for this is that there are occasions where data needs to be exchanged or shared prior to an effectivity being defined. This tends to occur early in a new design.

All effectivities in AP 209 ed2 are explicit effectivities and there are no assumed effectivities. Some systems in existence today assume a part is effective for all planned or actual instances of a product model if the effectivity is not explicitly defined. This is not the intent in AP 209 ed2. If a part has no effectivity in the AP 209 ed2 data structures, it has no effectivity. If a part is effective for all instances of a product model, the data should explicitly state all the effective instances. The effectivities in AP 209 ed2 contain open ranges for serial numbers and dates to allow for open or full effectivities. Using these constructs, all that is required is a start point. If there is a desire for full effectivity and the start point is not defined, the value `‘1’` should be used for the **serial_numbered_effectivity.effectivity_start_id** or the equivalent date of January 1st year 1 should used for **dated_effectivity.effectivity_start_date**.

> NOTE - Open effectivity does not make sense for a lot effectivity as it is inherently closed (other than lot size). Lot effectivity is typically an effectivity designated in the manufacturing view of a product or part.

The exchange or sharing of effectivity information creates the need for optional processing capability in at least pre-processors to allow for perspective. It is typically desirable for the lead contractor in a partnership or team to provide effectivity definitions to sub-contractors. It is usually undesirable for the lead contractor to utilize effectivities echoed back by sub-contractors as they reflect what was originally sent but not necessarily the most current data (in some cases).

The above is a simple case. Most cases involve even more variables such as who in the exchange or sharing arrangement is the defining body for the effectivity of a particular part or usage. One way to deal with this situation is for pre-processors to provide options for ignoring effectivity entirely, loading it or either ignoring or loading it based on externally defined criteria such as the part's design owner, design supplier or part number and for post-processors to provide a switch for a user choice on whether or not defined effectivity information in the system should be used in the interchange.

Pre-processor Recommendations: There is no standard mapping for the **id** attribute of the **effectivity** entity. Since there is no standard mapping in the AP 209 ed2 application domain for this attribute, it is recommended that this attribute contain a null string as minimal content or any appropriate or mutually-agreed-upon string. If the effectivity approval information is not readily available, it can be extrapolated from the engineering change that designated the effectivity. Pre-processors should interpret the value `‘1’` for the **serial_numbered_effectivity.effectivity_start_id** or the equivalent date of January 1st year 1 for **dated_effectivity.effectivity_start_date** as full or open effectivity when the values are specified with no ending range value. It is recommended that pre-processors provide options for ignoring effectivity entirely, loading it, or either ignoring or loading it based on externally defined criteria such as the part's design owner or part number to allow for a user choice as to whether the data is utilized or not depending on the source.

Post-processor Recommendations: There is no standard mapping for the **id** attribute of the **effectivity** entity. Since there is no standard mapping in the AP 209 ed2 application domain for this attribute, it is recommended that post-processors assign no processing significance to this value. When there is a need for full effectivity and the start point is not defined, post-processors should use the value `‘1’` for the **serial_numbered_effectivity.effectivity_start_id** or the equivalent date of January 1st year 1 for **dated_effectivity.effectivity_start_date**. It is recommended that post-processors provide a switch for a user choice on whether or not defined effectivity information in the system should be used in the interchange.

2.8.16. Composite Part and Constituent Representations

The composite part and constituent representations recommendations has been removed from this document and is now available in a separate document devoted only to this subject. As of the publication of this revision of this document, the composite Recommended Practices can be found in the document:

CAx-IF Recommended Practices for Composite Materials V 3.0 dated 2016.

2.9. Materials and Properties

Stock material is treated as a **product** in AP 209 ed2. A stock material product shall be among the **products** of a **product_related_product_category** with a name of : **`‘isotropic material’`**, **`‘anisotropic material’`**, **`‘filament assembly’`**, **`‘discontinuous fiber assembly’`**, **`‘braided assembly’`**, **`‘woven assembly’`**, or **`‘stock core’`** (Figure 16). The stock_material **product_definition** may have an **approval** in AP 209 ed2. See Section 2.4 for guidance on creating the **approval** and related entities.

Material properties, including finite element analysis material properties, are represented by the **property_definition** subtype **material_property**. The **name** attribute inherited from the **property_definition** supertype is used to denote the particular property being qualified or quantified. The **material_property_representation** entity links a **material_property** to a **representation** that may contain a **measure_representation_item** in its set of items to provide a quantitative value the property.

For a finite element analysis (FEA), the **material_property_representation** subtype **fea_material_ property_representation** entity is used to link an FEA **material_property** to a property **representation**. There shall be a single FEA material property representation item for each material property. Therefore, the FEA material property **representation** shall contain only one **fea_material_property_ representation_item** subtype in its set of **items**. The subtypes of **fea_material_property_ representation_item** represent finite element analysis properties such as linear elasticity, mass density, shell shear stiffness, and coefficient of thermal expansion. The material id assigned to a material by an application is represented by the **name** attribute of the **representation**. The material id shall be unique within the **fea_model** (see Section 2.9.3.1).

Conditions such as temperature and moisture content that relate to the material properties are grouped in a **data_environment** that is referenced by the **material_property_representation** entities as their **dependent_environment**. The **representation** for each condition is associated with the stock material through a **property_definition**. The **representation** of a material reference direction is likewise associated with the stock material through a **property_definition**.

2.9.1.1. Material Specifications

Material specifications that are applicable to a material are related to the material **product_definition** through an **applied_document_reference** entity. The material **product_definition** is contained in the **items** of the **applied_document_reference**. The **assigned_document** attribute inherited from the **document_reference** supertype of **applied_document_reference** points to the specification **document** (Figure 16).

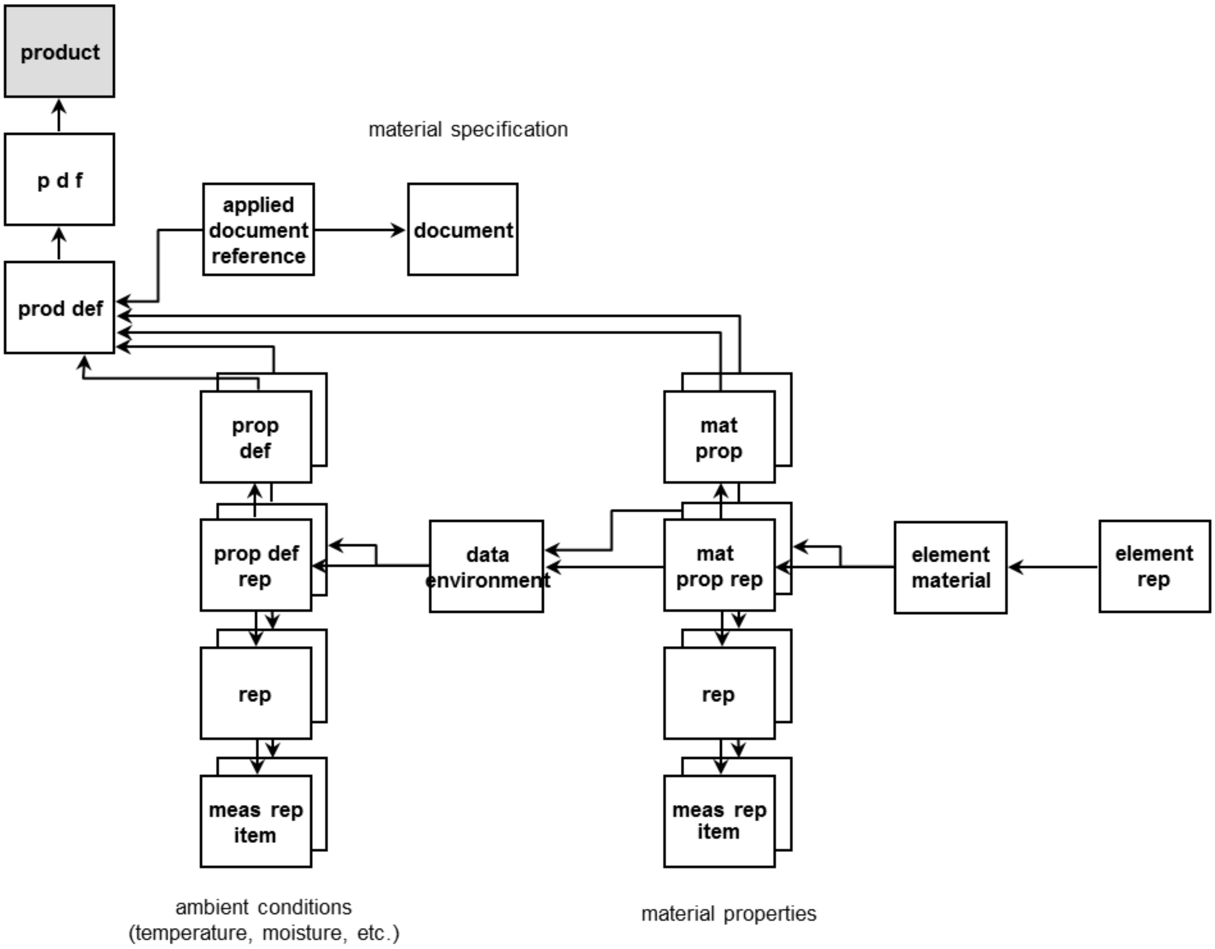


Figure 16 – FEA Generic Material Properties

2.9.1.2. Material Callout

The designation of the material for a part is accomplished through a **make_from_usage_option** entity. The **make_from_usage_option.relating_product_definition** shall be the **`'design discipline'`** **product_definition** for the part. If the component part is produced from a single material, then the **make_from_usage_option.related_product_definition** shall be the **product_definition** for the material (Figure 16).

A simplified methodology for specifying material properties as a general property is specified in the document:

Recommended Practices for Material Identification and Density Release 2.1 July 12, 2005 published by the CAx Implementor Forum of PDES, Inc and PROSTEP AG.

2.10. Retention

Retention defines a period of time for which product data is to be maintained due to organizational policy or legal requirements. In AP 209 ed2, the **action** subtype **retention** is used for this purpose. An instance of **retention** shall have an associated **applied_retention_assignment**, whose **items** contain the entities that are to be retained by **retention**. An instance of **retention** shall also have associated with it **applied_date_and_time_assignment**s or **applied_date_assignment**s that indicate the start, earliest end, and latest end dates for retention of data. The **description** attribute inherited from **action** is used to describe the purpose for **retention**. The method chosen for retention is given by an **action_method** (see Figure 17).

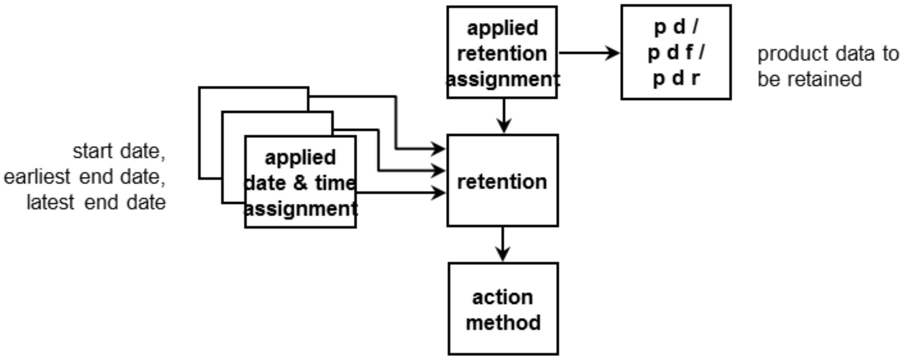

applied retention assignment
p d /
p d f /
p d r
product data to be retained
start date,
earliest end date,
latest end date
applied date & time assignment
retention
action method

Figure 17 - Retention of Product Data

The **retention** may have an **approval** in AP 209 ed2. See Section 2.4 for guidance on creating the **approval** and related entities. Retention may be applied to product data such as analysis and design discipline product definitions, parts, analyses, part and analysis versions, assemblies, material callouts, material properties and specifications, fea models, control and results, and analysis reports.

2.11. Analyses in AP 209 ed2

An analysis is defined in AP 209 ed2 in the same manner as parts, by using three entities. The **product** entity establishes the type and description of analysis. The **product_definition_formation** entity identifies its version (or change level). The **product_definition** or **product_definition_with_associated_documents** entity identifies the engineering discipline view that all the data related to it represents (e.g., design/analysis engineering). Through these three entities and their EXPRESS subtypes, the analysis is identified, revision controlled, and life cycle stage insulated. Figure 18 describes the relationships among the entities that are necessary to define an analysis in AP 209 ed2 at a high level.

The analysis types in AP 209 ed2 have been greatly expanded over the initial ed1 version, adding a generic engineering analysis core covering mesh-based and meshless analyses, and a generic mesh-based numerical analysis and CFD capability. This 2.0 version of the Recommended Practices document covers only the classical Finite Element capabilities initially covered in ed1.

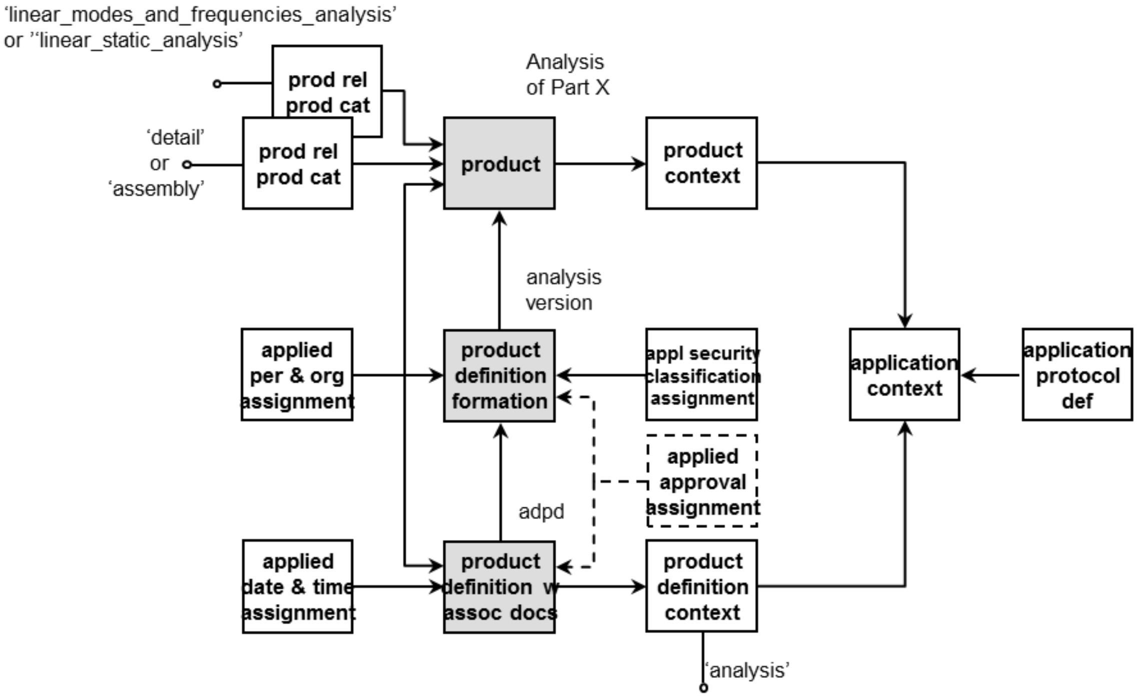


Figure 18 - High Level AP 209 ed2 Requirements for Analyses

2.11.1. Identifying Analyses

Figure 19 shows the overall representation for an analysis and its relationship to the corresponding part. The part is represented by the design discipline view as described in Section 2.8. Details of the analysis representation are discussed in the following paragraphs.

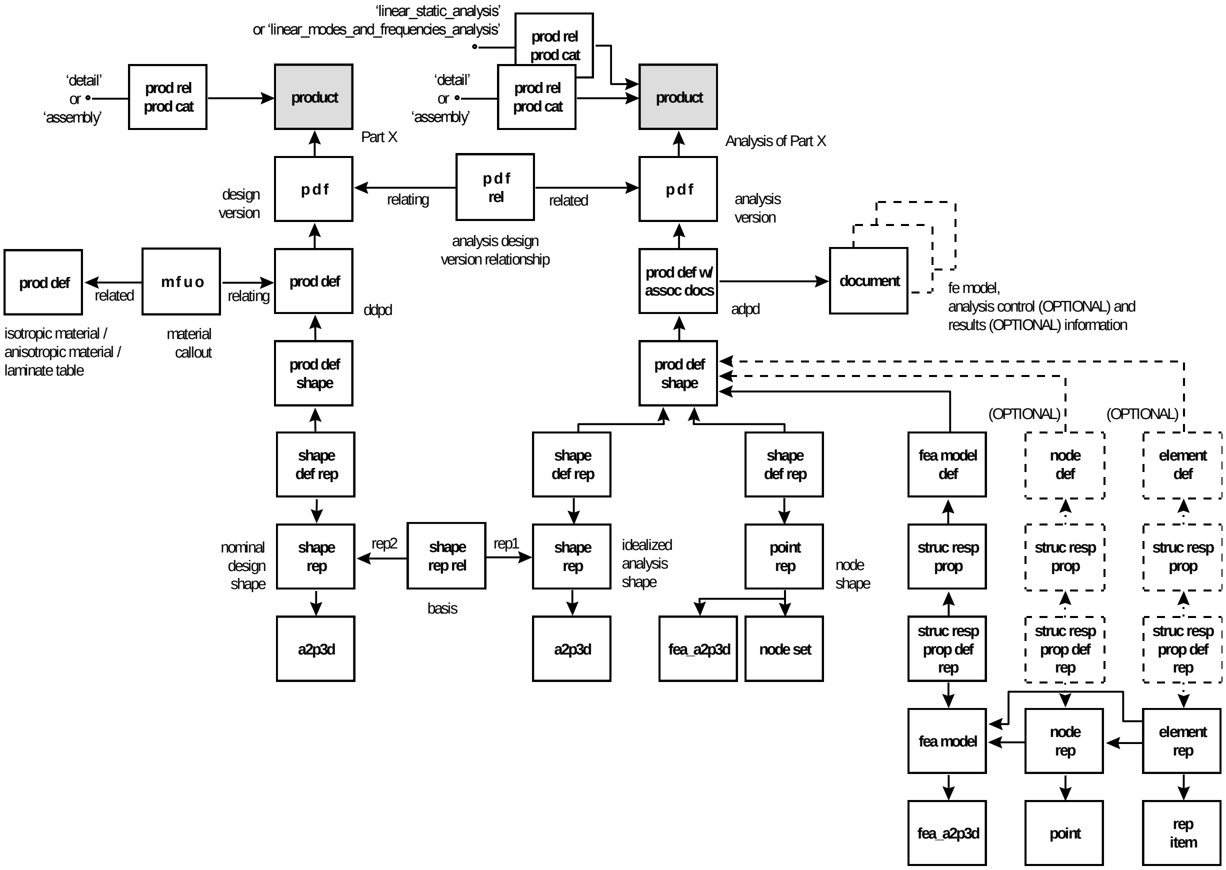


Figure 19 - Analysis vs. Design Discipline Product Definition

2.11.1.1. The Product Entity

AP 209 ed2 deals with analyses as **product**s. The **name** and **description** attributes can be used to indicate that the product represents an analysis for a particular part or assembly.

AP 209 ed2 requires that all **product**s exist in at least one **product_related_product_category**. This restriction (**product_requires_product_category**) forces all parts into one of the following categories: **`'detail'`**, **`'assembly'`**, **`'inseparable assembly'`**, or **`'customer furnished equipment'`**. An analysis **product** should be assigned to a **product_related_product_category** with the same name as that for the part that is being analyzed.

An analysis **product** should also be assigned to a **product_related_product_category** that indicates the type of analysis. The appropriate categories in AP 209 ed2 are: **`'linear static analysis'`** and **`'linear modes and frequencies analysis'`**. An analysis **product** should be assigned to both categories if it includes **control**s for both analysis types.

Analysis **product**s in AP 209 ed2 require a **person_and_organization** or **organization** in the role of **`'analysis owner'`**. This designation is applied to the person and organization or authority who originally performed the analysis. See Section 2.2 for guidance on how to create the person and organization entities.

Pre-processor Recommendations: All pre-processors should use non-defaulted data or user input for the values assigned to analysis owner of a **product** as defaulting this data has a high probability of causing the data to be incorrect.

2.11.1.2. The Product Definition Formation Entity

AP 209 ed2 requires that all part **product**s be associated with a **product_definition_formation** entity. This relation is required to support separate versioning of analyses in AP 209 ed2.

AP 209 ed2 requires that all **product_definition_formation** entities be associated with a **person_and_ organization** or **organization** in the role of **`'creator'`**. This person and organization is the one that created the change. The data for this person can be found by looking at the release or change paper work data and finding the initiator. For guidance on creating the entities associated with this data, see Section 2.2.

The **product_definition_formation** entity should also be associated with at least one person or organization in the role of **`'analysis supplier'`**. For guidance on creating the entities associated with this data, see Section 2.2. The person and organization in the role of **`'analysis supplier'`** is the one that was the custodian of the master data or analysis when the version was created.

In AP 209 ed2, the analysis **product_definition_formation** and entities may be associated with an **approval**. This is the person and organization that approved the analysis version. The data for this person can be found by looking at the release or change paper work data and finding who approved the release or change. For guidance on creating the entities associated with an **approval**, see Section 2.4.

AP 209 ed2 requires that all **product_definition_formation** entities be associated with a **security_ classification**. See Section 2.5 for guidance on creating entities associated with a **security_classification**.

Pre-processor Recommendations: All pre-processors should use non-defaulted data or user input for the values assigned to the creator, analysis supplier, approvers, and approval date for **product_definition_ formation** entities as defaulting this data has a high probability of causing this data to be incorrect.

The **security_classification** classification officer, classification date, approvers, and approval dates can be extrapolated from the version creator and approval data if no appropriate data is available.

2.11.1.3. Relating Analysis and Design Versions

AP 209 ed2 allows for separate versioning of analysis and part (design) versions. The corresponding **product_definition_formation**s are related through a **product_definition_formation_relationship**, where part version is the relating **product_definition_formation**, and analysis version is the related **product_definition_formation**.

2.11.1.4. The Product Definition Entity

AP 209 ed2 and STEP use the **product_definition** entity to establish specific analysis stage views of the analysis information. The use of **product_definition** entity establishes many important relationships such as analysis to analysis shape and finite element models. The **product_definition** entity can be related to **document** entities that describe the finite element model, analysis control and results, through the subtype **product_definition_with_associated_documents**.

It is possible to have many **product_definition**s for an analysis-version combination. The **id** attribute should identify whose view of the analysis **product** a particular instance represents. There are no standard mappings in the AP for this attribute or the **description** attribute (see pre-processor recommendations).

AP 209 ed2 requires that all **product_definition**s have a **person_and_organization** or **organization** assigned in the role of **`'creator'`**. This person and organization is the one that defined the view. For guidance on creating these constructs, see Section 2.2.

AP 209 ed2 requires that all **product_definition**s have a **date_and_time** or **date** assigned in the role of **`'creation date'`**. This date and time is when the view was defined. Typically, this would be the date and time for the CAD model of the shape or for the FEA Model If this is not the case, see the pre-processor recommendations. For guidance on creating the date and time constructs, see Section 2.3.

In AP 209 ed2, the analysis **product_definition**s may have an **approval**. This data is often difficult to obtain as those who approved the filing of the CAD or FEA model or creation of the **product_definition** are difficult to identify. If the information is available, see Section 2.4 for guidance on creating the approval.

AP 209 ed2 has an optional feature where a **product_definition** may be related to **document** entities through the subtype **product_definition_with_associated_documents**. In the context of the analysis product, this usage is intended for documents that identify analysis controls or results reports. See Section 2.9.3.3 for more information.

Pre-processor Recommendations: There is no standard mapping for the **id** attribute of **product_definition**. Therefore, it is recommended that this attribute contain possible values of **`'design'`**, **`'analysis'`**, **`'digital pre-assembly'`**, **`'manufacturing'`**, **`'as built'`**, **`'as maintained'`**. These values should be used to indicate which group owns the view for concurrent engineering purposes within a life cycle stage. There is no standard mapping for the **description** attribute. Therefore, it is recommended that this attribute contain a null string as minimal content or any appropriate or mutually-agreed-upon string. Where values for the creator and creation date are not readily available, this information can be extrapolated from the creator and approval related to the **product_definition_formation** as defined in Section 2.8.1.2.

Pre-processors may use the / character as a delimiter to separate the sending system identification from the actual file name for the **document id** attribute if the receiving system does not have a uniqueness requirement on this value.

Post-processor Recommendations: All post-processors should utilize the values given above for pre-processors as computer sensible segregations of the **product_definition** data based on the **id** attribute. If a value other than those above is received, it should be regarded as **`'design'`**. Since there is no standard mapping for the **description** attribute for the **product_definition** entity (and subtype), it is recommended that post-processors not assign any processing significance to this value.

2.11.2. Relating Analysis Shape to Analysis

AP 209 ed2 uses two entities to form the link between the configuration management data for an analysis and its shape. These two entities are **product_definition_shape** and **shape_definition_representation**. There are no standard mappings for the **product_definition_shape name** and **description** attributes.

There must be only one **product_definition_shape** for each **product_definition** in an AP 209 ed2 exchange file. If there are multiple **shape_definition_representation** entities related to the **product_definition_shape**, these relationships describe alternate representations. In AP 209, the analysis shape representation may be a point model, composed of points only (**point_representation**). The **name** attribute of this **representation** is set to **`'node shape'`**. A **point_representation** shall contain only **node_set**, **mapped_item** and **fea_axis2_placement_3d** entities in its set of **items**. Alternately, the analysis shape representation may an idealization of the design shape. The **name** attribute of the **representation** is set to **`'idealized analysis shape'`** in this case. The idealized analysis shape representation is related to the design shape representation **(`'nominal design shape'`)** through a **shape_ representation_relationship** (see Figure 19).

Pre-processor Recommendations: There are no standard mappings for the **name** and **description** attributes for **product_definition_shape**. Therefore, it is recommended that these attributes contain a null string as minimal content or any appropriate or mutually-agreed-upon string.

Post-processor Recommendations: Since there are no standard mappings for the **name** and **description** attributes for **product_definition_shape**, it is recommended that post-processors not assign any processing significance to these values.

2.11.3. Finite Element Analysis

In performing an analysis with finite element models, the continuum of the product is discretized into a finite element model that is composed of a mesh of points. The nodes are connected with elements, which represent finite subdivisions of the continuum and model its behavior.

2.11.3.1. Finite Element Model

The finite element model is represented by an **fea_model** entity, that will be either an **fea_model_2d** or **fea_model_3d**, depending on the analysis coordinate space dimension. The finite element id assigned to the model by an application is represented by the **name** attribute inherited from its **representation** supertype. The model id shall be unique to the **fea_model**. The name of the software used to create the model is given by the **creating_software** attribute. The type of analysis to be performed with this model is specified by the **analysis_type** attribute, and the analysis code(s) that the model was created for by the **intended_analysis_code** attribute. Each analysis code shall have the vendor, version, computer system, operating system and descriptions specified. The file that contains the information describing the **fea_model** is a **document** associated with the analysis **product_definition** (**product_definition_with_associated_documents**). The **document** points to a document type of **`'cae file'`**, and the **document id** attribute specifies the name of the file.

NOTE: TC1 for Part 104 removed a duplicate element ID for **fea_model** entity.

AP 209 ed2 uses two entities to form the link between the configuration management data for an analysis **product** and the finite element model or its individual element and node representations. These two entities are **structural_response_property** and **structural_response_property_definition_representation** (see Figure 19).

The **structural_response_property_definition_representation** points to the representation for the finite element model as a whole, or to the representation for an individual element or node of the model (**fea_model**, **element_representation**, or **node_representation**), while the **structural_response_property** points to the corresponding descriptions (**fea_model_definition**, **element_definition**, or **node_definition**). As subtypes of **shape_aspect**, these definitions are related to the configuration management data for an analysis and its shape representation via the analysis **product_definition_shape**.

In AP 209 ed2, **fea_model**s may have an **approval**. This is the person and organization that approved the **fea_model**. This data is often difficult to obtain as those who approved the filing of the FEA model are difficult to identify. If the information is available, see Section 2.4 for guidance on creating the approval.

Pre-processor Recommendations: There are no standard mappings for the **name** and **description** attributes for **structural_response_property**. Therefore, it is recommended that these attributes contain a null string as minimal content or any appropriate or mutually-agreed-upon string.

Post-processor Recommendations: Since there are no standard mappings for the **name** and **description** attributes for **structural_response_property**, it is recommended that post-processors not assign any processing significance to these values.

As a clarification to 10303-42, units on parametric representations are taken from the **global_unit_assigned_context** entity. They are not always degrees as might be extrapolated from reading the text of 10303-42. This is a consideration on choosing the global units for plane angles, as radian units are irrational and potentially unstable.

2.11.3.1.1. Node Representation

Nodes in a finite element model are represented by one of the subtypes of **node_representation** (**node**, **geometric_node**, and **dummy_node**). A **node_representation** is associated to the **fea_model** through its **model_ref** attribute (Figure 20). The node id assigned to a node by an application is represented by the **name** attribute inherited from its **representation** supertype. The node id shall be unique within the **fea_model**.

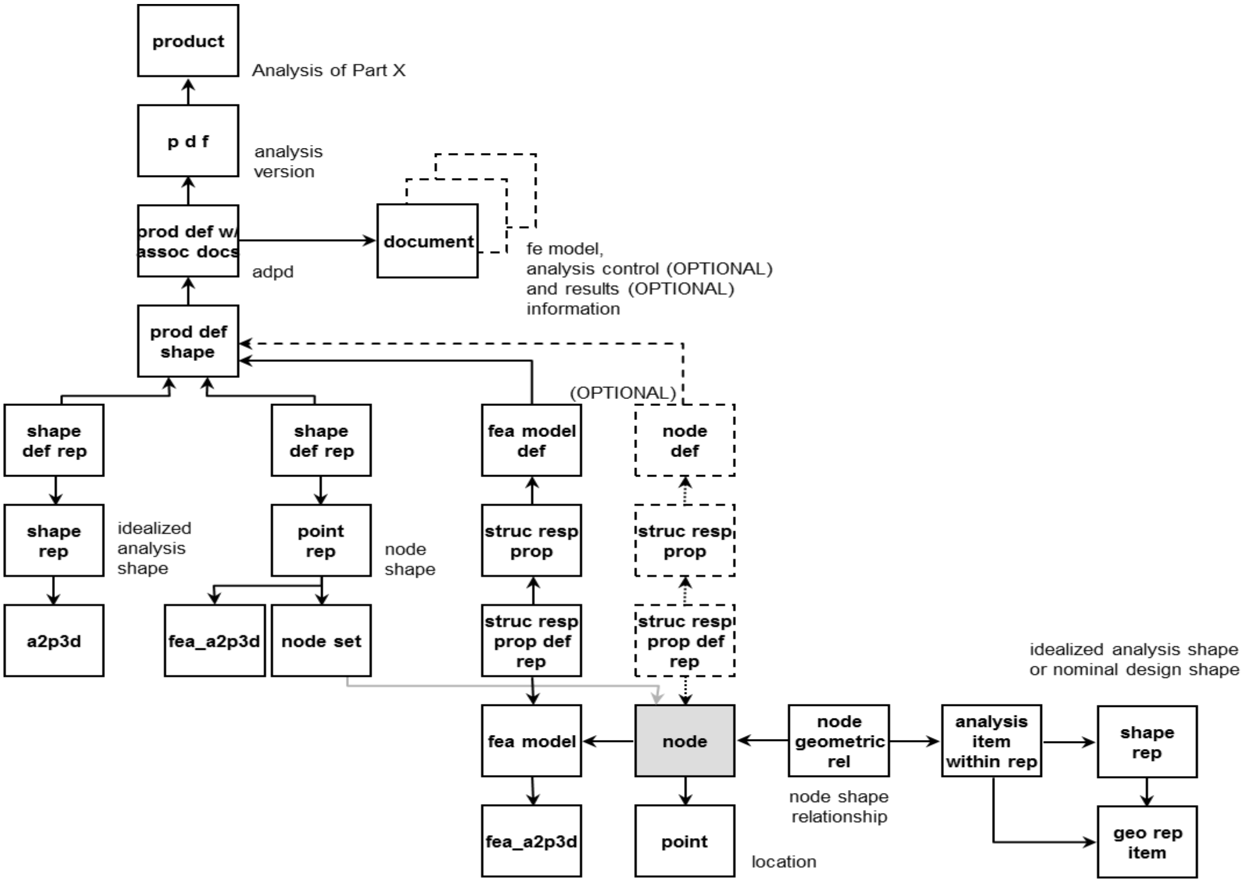


Figure 20 - Node

Nodes of the finite element model are aggregated in a **node_set** for geometric founding. The **node_set** is included in the set of items of the **point_representation**. The **point_representation** may also contain an **fea_axis2_placement_2d** or **fea_axis2_placement_3d** to relate to other coordinate systems.

A node may be associated to a geometry through the **node_geometric_relationship** entity. The **node_geometric relationship** shall point to an **analysis_item_within_representation** that references: a) the **geometric_representation_item** that the **node** is being associated to, and b) the **shape_representation** that the **geometric_representation_item** is used in. This **shape_representation** can be the nominal design or idealized analysis **shape_representation**, or be related to the nominal design or idealized analysis **shape_representation** through a **shape_representation_relationship**.

2.11.3.1.2. Element Representation

Elements used in a finite element model are represented by one of the **element_representation** subtypes. A **element_representation** is associated to the **fea_model** through its **model_ref** attribute (Figure 21). The element id assigned to an element by an application is represented by the **name** attribute inherited from its **representation** supertype. The element id shall be unique within the **fea_model**.

NOTE: TC1 for Part 104 removed a duplicate element ID for **point_element_representation**, **directionally_expicit_element_representation**, and **explicit_element_representation entities**.

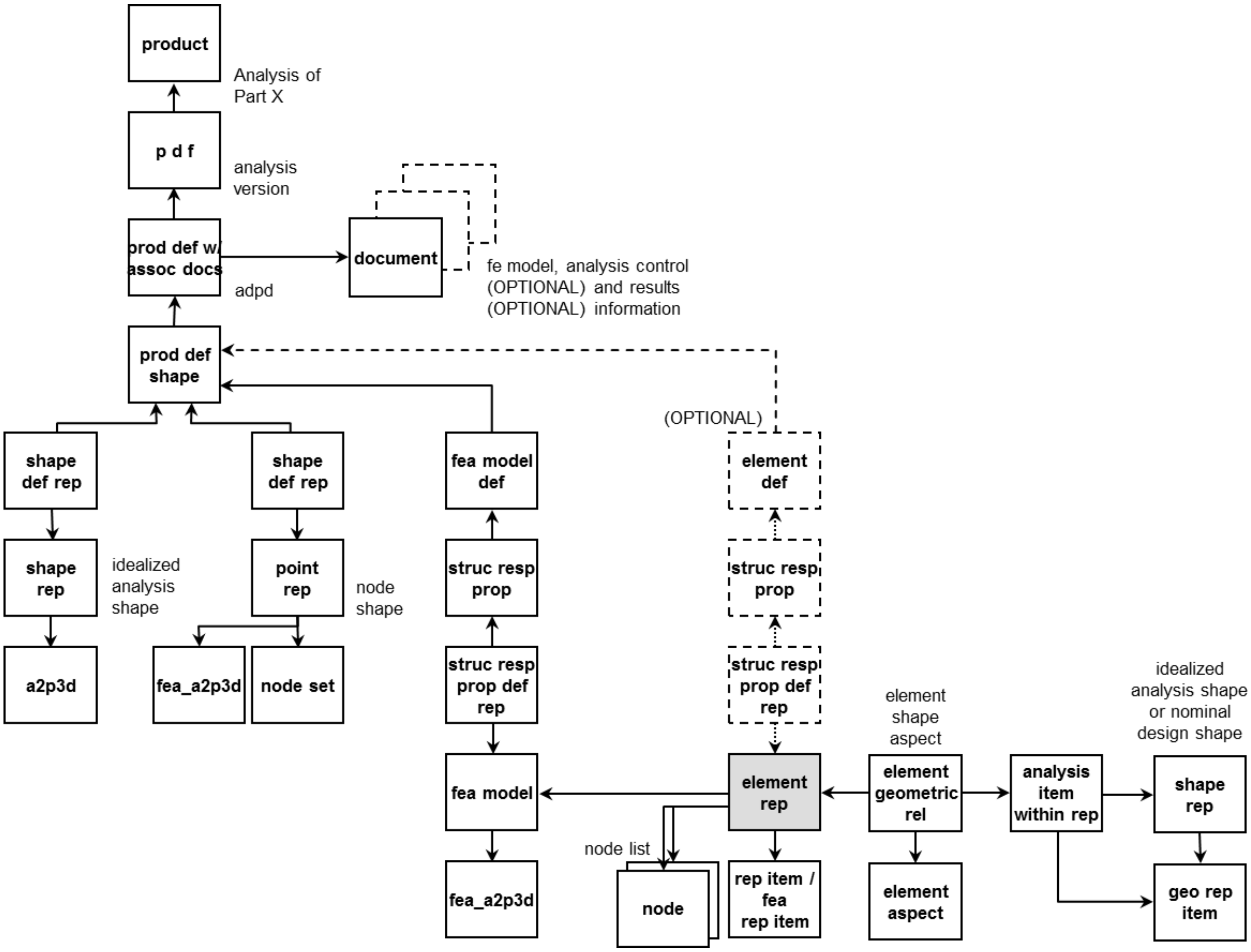


Figure 21 - Element

An **element_representation** has a list of **node_representation**s that are associated with the element. The number of nodes in the list shall be consistent with the order of interpolation for the element. The order of interpolation is specified by a reference to the appropriate **element_descriptor** subtype that is also used to define the shape and responses of like elements.

Aspects of an element such as a face, edge, or volume are associated to an appropriate geometry through the **element_geometric_relationship** entity. The **element_geometric relationship** shall point to an **analysis_item_within_representation** that references: a) the **geometric_representation_item** that the **element** is being associated to, and b) the **shape_representation** in which the **geometric_representation_ item** is used. This **shape_representation** can either be the nominal design or idealized analysis **shape_representation**, or be related to the nominal design or idealized analysis **shape_representation**.

Element_representations shall reference the proper element coordinate system to establish the material or section properties of the element. Volume, surface and curve **element_representation**s shall reference an **element_material** entity that is an aggregate of material properties. The material properties are represented by **material_property_representation** or **fea_material_property_representation** entities, as discussed in Section 2.8.17.

2.11.3.1.3. Element Locations

Element locations are needed to integrate finite element matrices over the volume or surface of an element, or to evaluate a field variable at specific points in an element. The locations of integration, basis, and field variable output points are all specified in the same manner. Note that there is no requirement that the output points for an element be identical to the integration points (if any). The specification of these locations within **element_representations** is detailed in clause 5.11 of ISO 10303-104, and a discussion of how these locations are used in element matrix integration is detailed in clause 5.10. In the implementation of the writing of this type of data, care should be taken to only output as many locations as is necessary for the various element types in order to minimize the volume of data. To this end, the data model provides the capability to share the element integration and output definitions amongst many elements.

2.11.3.1.4. Relating Element Locations to Lamina for Layered Elements

The location of integration, basis, and field variable specification points in curve, surface, and volume elements, though specified in the same manner, differ due to the different levels of geometric abstraction in the element types. The integration points are of interest when considering how the material properties are interpreted within an element along with other matrix integration topics (see clauses 5.10 and 5.11 of ISO 10303-104). Note that element parametric coordinate systems (see clause 5.9 of ISO 10303-104) have to be taken into consideration when calculating element material properties.

The relationship between a laminate table and an FEA material property may be established at one or more levels within a laminate table (all illustrated in the upper right portion of Figure 22):

- the overall laminate table may be linked to an FEA material property;
- a ply laminate sequence table may be linked to an FEA material property;
- a composite assembly sequence table may be linked to an FEA material property.

For curve elements, the relationship of the element properties to materials is strictly of informational value as far as the response of the element is concerned since the stiffness of the curve elements are explicitly defined by the curve properties such as moments of inertia and torsional constant. The relationship to composite structure in a laminate table may be used in pre- or post-processing of property and field information using direct geometric correspondence.

For surface elements, a location is specified by a combination of the location over the surface combined with a location through the thickness. For layered surface elements, the thickness is specified by the related laminate table and therefore is not explicitly defined in the surface property entity. There are two methods of defining location through the thickness of a surface element: the first is absolute, where the location through the thickness is a specific distance; and the second is a dimensionless distance from -1.0 to 1.0 that spans the thickness of the element. If a location falls on a ply boundary, a flag is available to indicate the side that it lies upon. In both the absolute and dimensionless cases there is no direct correlation between plies in a laminate table and a location: the correlation must be geometrically established in a post-processing step. Though computationally cumbersome, this provides an extremely general correlation capability that supports a wide range of composite laminates, from simple point zone laminate specifications to complex laminate assemblies.

When part laminate tables are used to specify the composite material, a piercing algorithm must be used to calculate the laminate thickness at each location within a surface element. Except for a simple laminate table where all ply boundaries are identical, the non-uniform coverage of the part by plies necessitates the piercing to calculate thickness and the plies involved. In the case of zone laminate tables (edge or point), no piercing is necessary since by definition a zone is of constant thickness and hence the list of plies and the corresponding thickness are directly known. In all cases the laminate is assumed to be congruent to the surface of the surface element, so that major material direction always runs tangent to the element surface.

For volume elements, the locations within an element are specified in parametric 3-D space (non-dimensional). Again the correlation of volume element locations to plies in a laminate table is strictly geometric. Note that since the material direction in a volume element is an arbitrary three-dimensional direction, there are no restrictions on the major material direction within the element.

For all element types, the calculation of material direction must take into account the method chosen to specify the ply orientation angle for the laminate table (see Figure 22). This will have the most (detrimental) effect on highly curved parts if the simplest option, i.e., a direction (equivalent to a single rosette), is used as the direction may change over the curvature of the part. If the point path or curve option is chosen to provide a more accurate direction specification, then the material direction must be calculated at each location in the element using the corresponding geometric location in the laminate part. This is done by first piercing the table to find the part location that corresponds to the element location, and then deriving the direction at that location within the laminate.

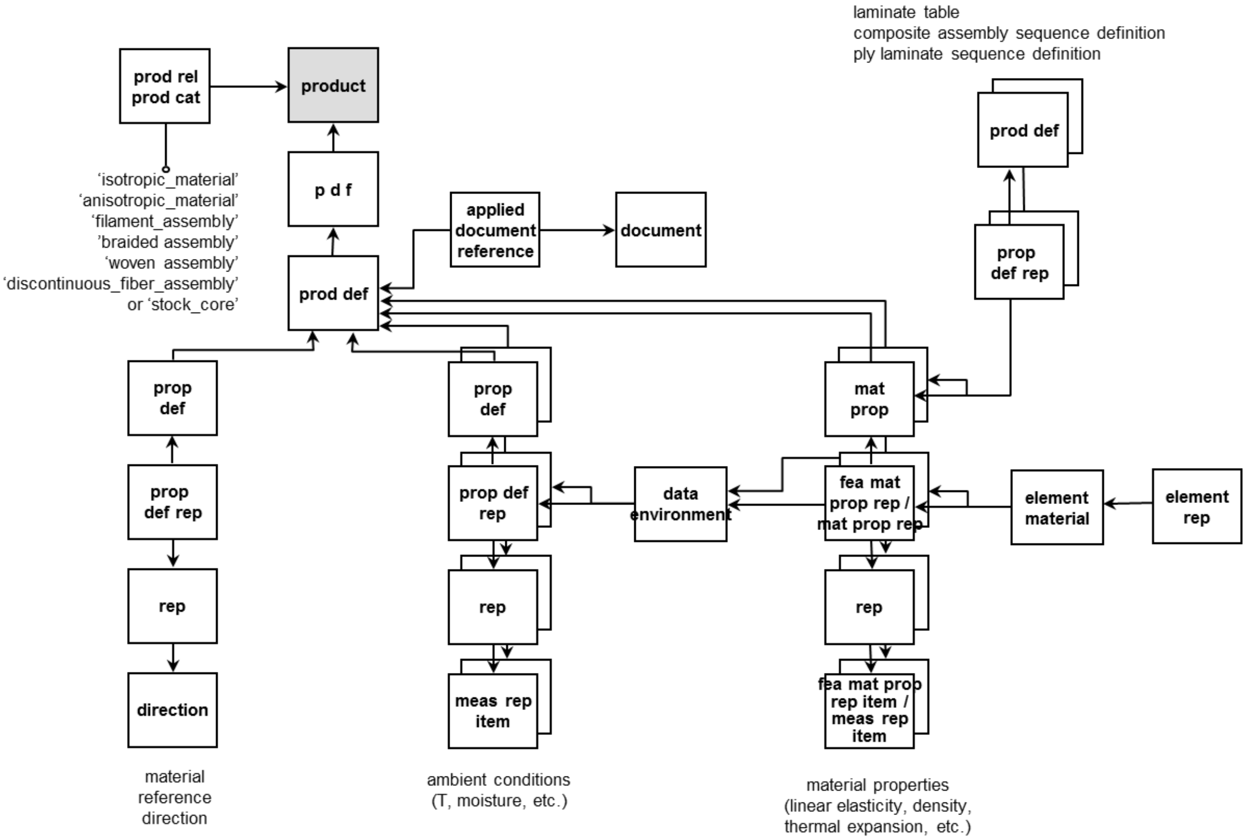

laminate table
composite assembly sequence definition
ply laminate sequence definition
prod rel prod cat
product
‘isotropic_material’
‘anisotropic_material’
‘filament_assembly’
’braided assembly’
‘woven assembly’
‘discontinuous_fiber_assembly’
or ‘stock_core’
p d f
applied document reference
document
prod def
prod def
prop def rep
prop def
prop def
mat prop
prop def rep
prop def rep
data environment
fea mat prop rep / mat prop rep
element material
element rep
rep
rep
rep
direction
meas rep item
fea mat prop rep item / meas rep item
material reference direction
ambient conditions (T, moisture, etc.)
material properties (linear elasticity, density, thermal expansion, etc.)

Figure 22 – FEA Composite Material Properties

2.11.3.1.5. Surface Element Section Properties

Section properties such as offsets, nonstructural mass, thickness, bending thickness and shear thickness of a 3D or 2D surface element are specified by the **surface_property** entity (Figure 23). The **surface_property** references a subtype of **surface_section_field** to denote the variation of section properties over the surface of the element.

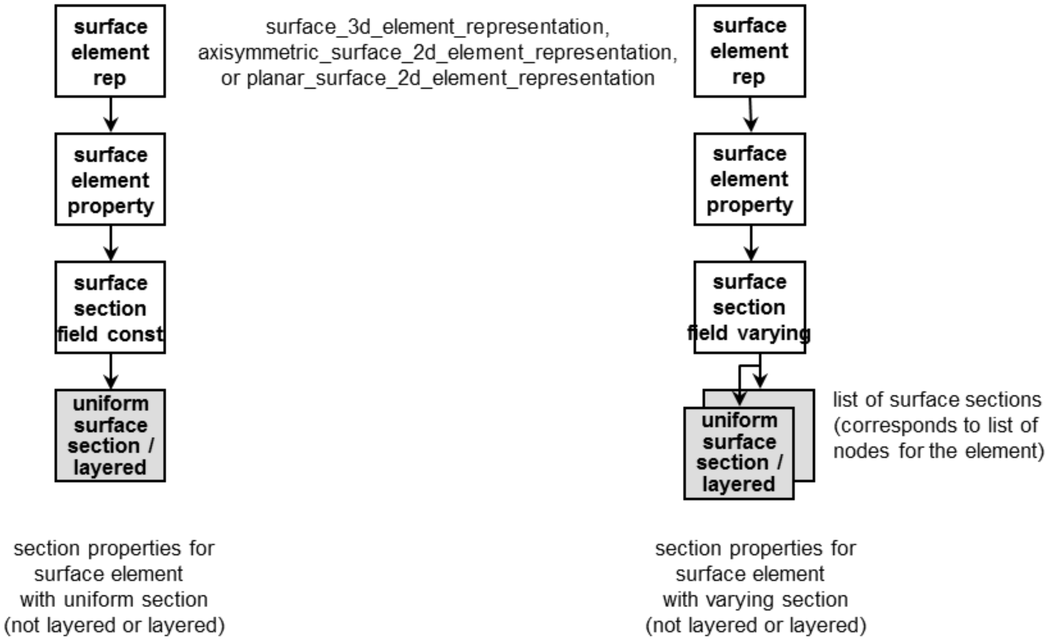

surface element rep
surface_3d_element_representation, axisymmetric_surface_2d_element_representation, or planar_surface_2d_element_representation
surface element rep
surface element property
surface element property
surface section field const
surface section field varying
uniform surface section / layered
uniform surface section / layered
list of surface sections (corresponds to list of nodes for the element)
section properties for surface element with uniform section (not layered or layered)
section properties for surface element with varying section (not layered or layered)

Figure 23 - Surface Property

If the section properties are uniform, then the **surface_section_field_constant** entity is used. The **surface_section_field_constant** references one of the **surface_section** subtypes to define the section properties for the element. For a nonlayered element, the subtype **uniform_surface_section** is used to specify the membrane thickness, and any bending thickness and shear thickness. If the surface element is layered across its thickness, the **surface_section_field_constant** references a **uniform_surface_section_layered** entity. In this case, the membrane thickness is specified or determined elsewhere, by summing up the layer thicknesses.

If the section properties vary over the surface of the element, then the **surface_section_field_varying** entity is used. The variation over the surface is represented by a list of **surface_section** entities referenced by the **surface_section_field_varying** entity. As discussed above, the **surface_section** subtype **uniform_surface_section** is used for nonlayered elements, and **uniform_surface_section_layered** for layered elements**.**

2.11.3.1.6. Curve Element Cross-section Properties

Cross_section information such as area, moment of area, torsional and warping constants, etc. is specified by referencing the **curve_2d_element_property** for 2D axisymmetric and planar curve elements, and the **curve_3d_element_property** for 3D curve elements.

2.11.3.1.7. Curve 2D Element Cross-section Properties

The **curve_2d_element_property** references a **curve_element_derived_section_definitions** entity that defines the cross-sectional information. The cross-sectional shape is represented by a **shape_representation** and related to the **element_representation** through a **representation_relationship**. The cross-sectional **shape_representation** may also be linked to the idealized analysis or nominal design **shape_representation** by a **shape_representation_relationship** (Figure 24).

2.11.3.1.8. Curve 3D Element Cross-section Properties

A 3D curve element can be divided into intervals, each of which may have a separate property specification. Each interval is represented by a subtype of the **curve_element_interval** entity. The subtype **curve_element_interval_constant** is used for an interval of constant section, and the subtype **curve_element_interval_linearly_varying** is used for an interval whose section varies linearly from one end of the interval to the other (Figure 24).

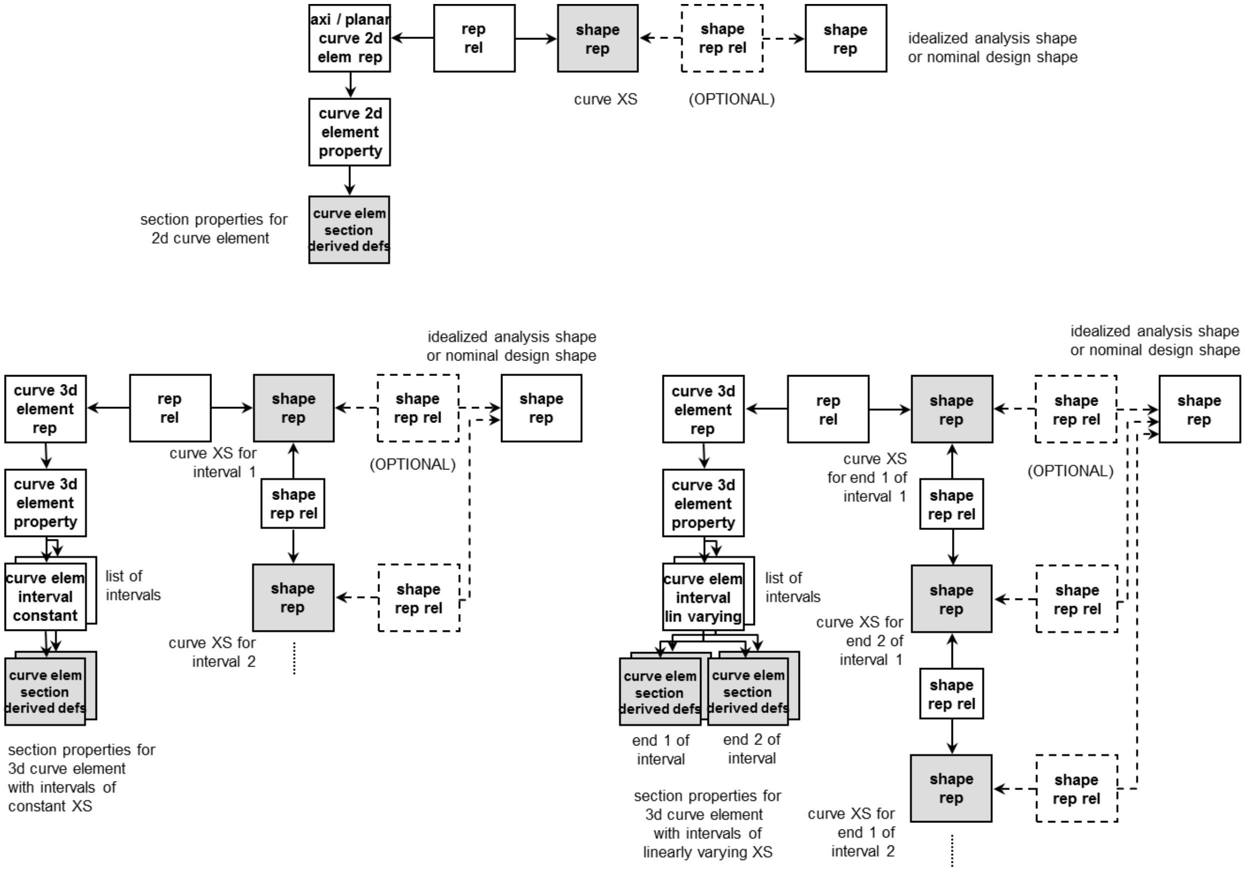

axi / planar curve 2d elem rep
rep rel
shape rep
shape rep rel
shape rep
idealized analysis shape or nominal design shape
curve XS
(OPTIONAL)
curve 2d element property
section properties for 2d curve element
curve elem section derived defs
idealized analysis shape or nominal design shape
curve 3d element rep
rep rel
shape rep
shape rep rel
shape rep
curve XS for interval 1
(OPTIONAL)
curve 3d element property
shape rep rel
curve elem interval constant
list of intervals
shape rep
shape rep rel
curve XS for interval 2
curve elem section derived defs
section properties for 3d curve element with intervals of constant XS
idealized analysis shape or nominal design shape
curve 3d element rep
rep rel
shape rep
shape rep rel
shape rep
curve XS for end 1 of interval 1
(OPTIONAL)
curve 3d element property
shape rep rel
curve elem interval lin varying
list of intervals
shape rep
shape rep rel
curve XS for end 2 of interval 1
curve elem section derived defs
curve elem section derived defs
end 1 of interval
end 2 of interval
shape rep rel
section properties for 3d curve element with intervals of linearly varying XS
curve XS for end 1 of interval 2
shape rep
shape rep rel

Figure 24 - Curve Property and Cross-section

The **curve_element_interval_constant** entity references a **curve_element_derived_section_definitions** entity that defines the cross-sectional information for the interval. The cross-sectional shape of each interval is given by a **shape_representation**. The shape_representations are chained together by **shape_representation_relationship** entities in the order of the intervals. The first **shape_representation** in the chain, i.e., the **shape_representation** for the first interval, is related to the **element_representation** through a **representation_relationship**.

The **curve_element_interval_linearly_varying** entity references two **curve_element_derived_section_ definitions** entities that define the cross-sectional information for the two ends of the interval. The cross-sectional shape at either end of the interval is given by a **shape_representation**. These **shape_representations** are chained together by **shape_representation_relationship** entities, from end 1 for the first interval to end 2 for the first interval, and then to end 1 for the second interval, and so on. The **shape_representation** for end 1 of the first interval is related to the **element_representation** through a **representation_relationship**.

The cross-sectional **shape_representation**s for the intervals may be linked to the idealized analysis or nominal design **shape_representation** by **shape_representation_relationship** entities.

2.11.3.1.9. Explicit Elements

Explicit element representations (**point_element_representation**, **explicit_element_representation**, and **directionaly_explicit_element_representation**) are used to state the stiffness, mass, and damping nodal response matrices explicitly.

2.11.3.1.10. Node and Element Groups

Groups of nodes and elements are represented by the **fea_group** subtypes **node_group** and **element_group**, respectively (Figure 25). **Element_group**s are subtyped further to contain 2D or 3D volume, surface, and curve elements. An **fea_group** is associated to the **fea_model** through its **model_ref** attribute The group id assigned to an element by an application is represented by the **name** attribute inherited from the **group** supertype of **fea_group**. The group id shall be unique within the **fea_model**. Relationships between groups of nodes or elements are represented by the **group_relationship** subtype **fea_group_relation**.

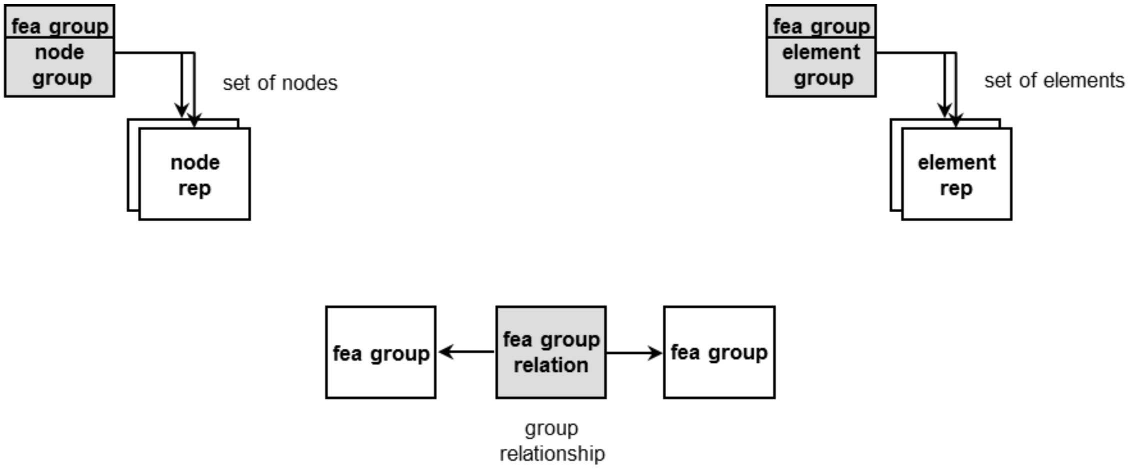


Figure 25 - Node and Element Groups

2.11.3.1.11. Units for Finite Element Model

Units for an **fea_model** are defined through the use of a complex instance of **global_unit_assigned_context** and **geometric_representation_context**. When global units are used, units must be defined for **length_unit**, **plane_angle_unit**, **solid_angle_unit, mass_unit, time_unit,** and **thermodynamic_temperature_unit**. The base units for STEP are Standard International (SI) units that are represented through the **named_unit** subtype **si_unit**. All other units (such as English units) are represented as **conversion_based_unit** entities that reference **si_unit**s. Physical file examples for SI and English units can be found in Appendix B.

2.11.3.2. FEA Specific Material Properties

Material properties idealized for use in FE Models are illustrated in Figure 26. This approach is an extension of the generic more CAD oriented approach illustrated in 2.9. The FEA specific properties are specialized for various types of idealizations of elastic response, along with properties such as density, and thermal and moisture coefficients.

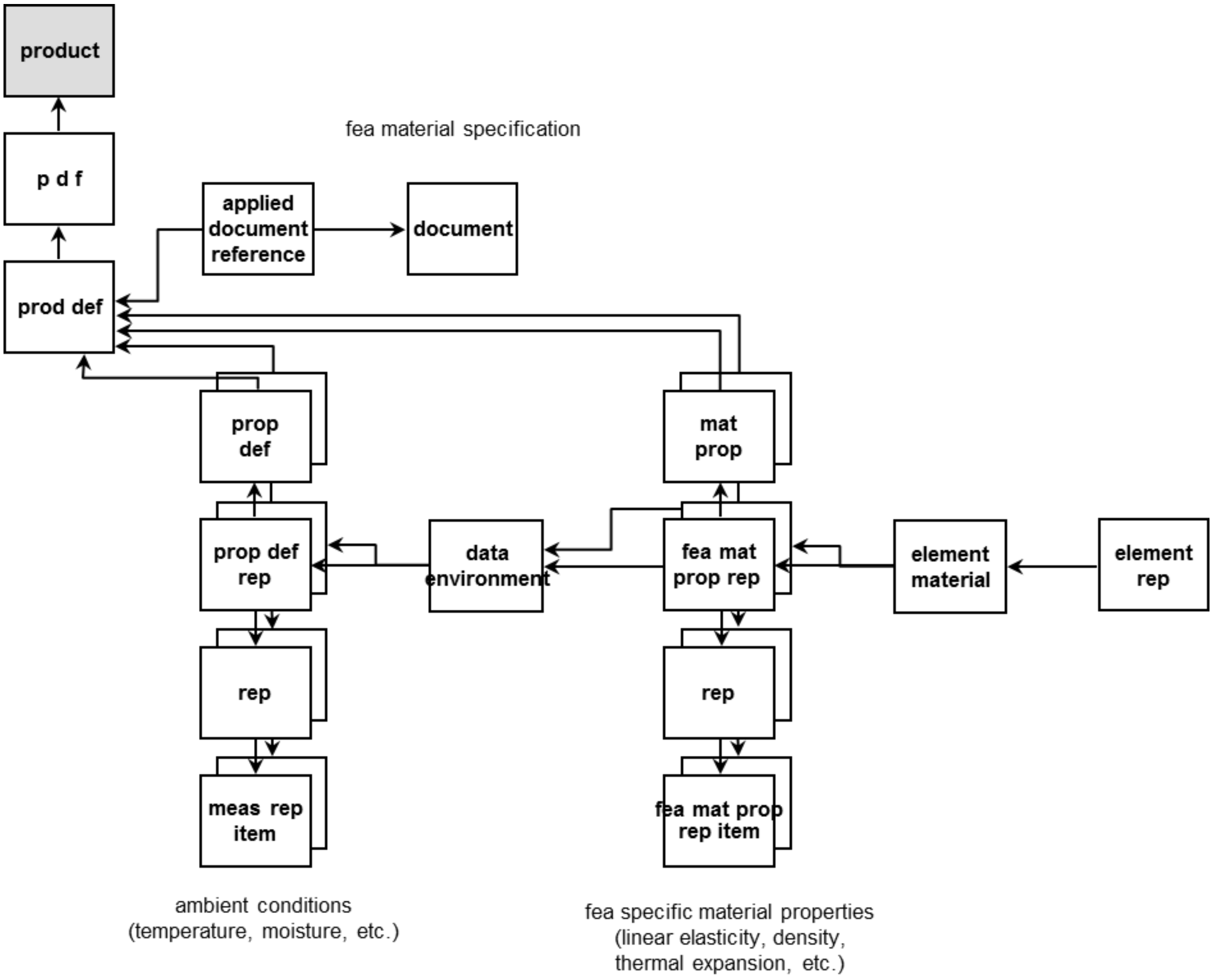

product
fea material specification
p d f
applied document reference
document
prod def
prop def
mat prop
prop def rep
data environment
fea mat prop rep
element material
element rep
rep
rep
meas rep item
fea mat prop rep item
ambient conditions (temperature, moisture, etc.)
fea specific material properties (linear elasticity, density, thermal expansion, etc.)

Figure 26 – FEA Specific Material Properties

2.11.3.3. Finite Element Analysis Control and Results

AP 209 ed2 supports two types of FE analysis: linear static analysis and linear dynamic (modes and frequencies) analysis. Each analysis has specific controls and associated results information.

2.11.3.3.1. Analysis Control

An FE analysis **control** describes the operations carried upon an fea_model. A **control** references the **fea_model** through its **model_ref** attribute. The control identifier represented by the **control_id** attribute shall be unique within the **fea_model**. In AP 209 ed2, an FE analysis **control** may have an **approval**. See Section 2.4 for guidance on creating the **approval** and related entities.

A single step in an analysis is represented by a **control_analysis_step**. A **control_analysis_step** is associated with a **control** through the attribute **analysis_control** inherited from its **analysis_step** supertype, and has an initial state defined by a **state** entity.

The **control_linear_static_analysis_step** subtype of the **control_analysis_step** is used in a linear static analysis. The final equilibrium state resulting from the application of static loads and constraints is represented by the **control_linear_static_load_increment_process** (Figure 27).

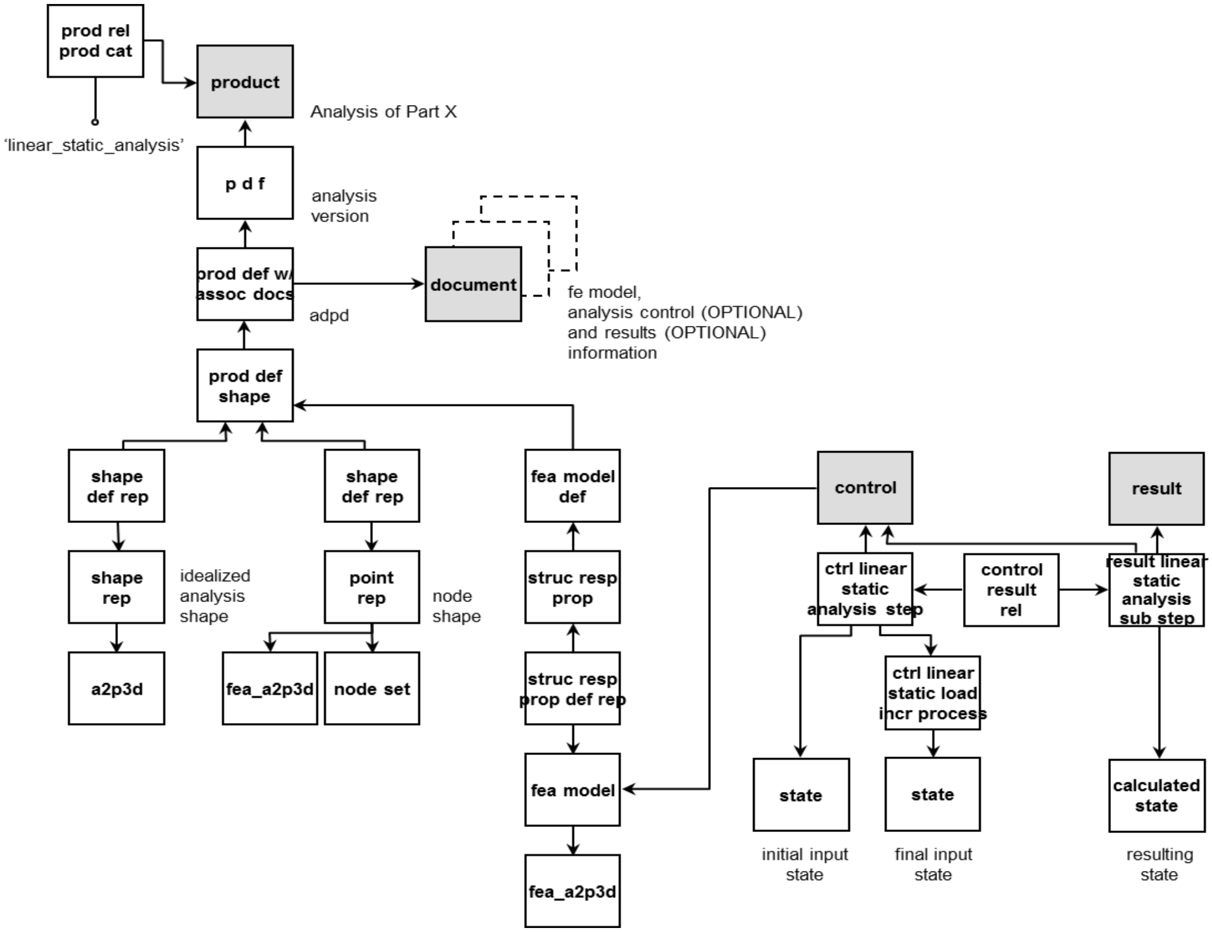


Figure 27 - Linear Static Analysis - Control and Results

The **control_linear_modes_and_frequencies_analysis_step** subtype of the **control_analysis_step** is used in a linear dynamic analysis. The final equilibrium state resulting from the application of dynamic loads and constraints is represented by the **control_linear_modes_and_frequencies_process** (Figure 28).

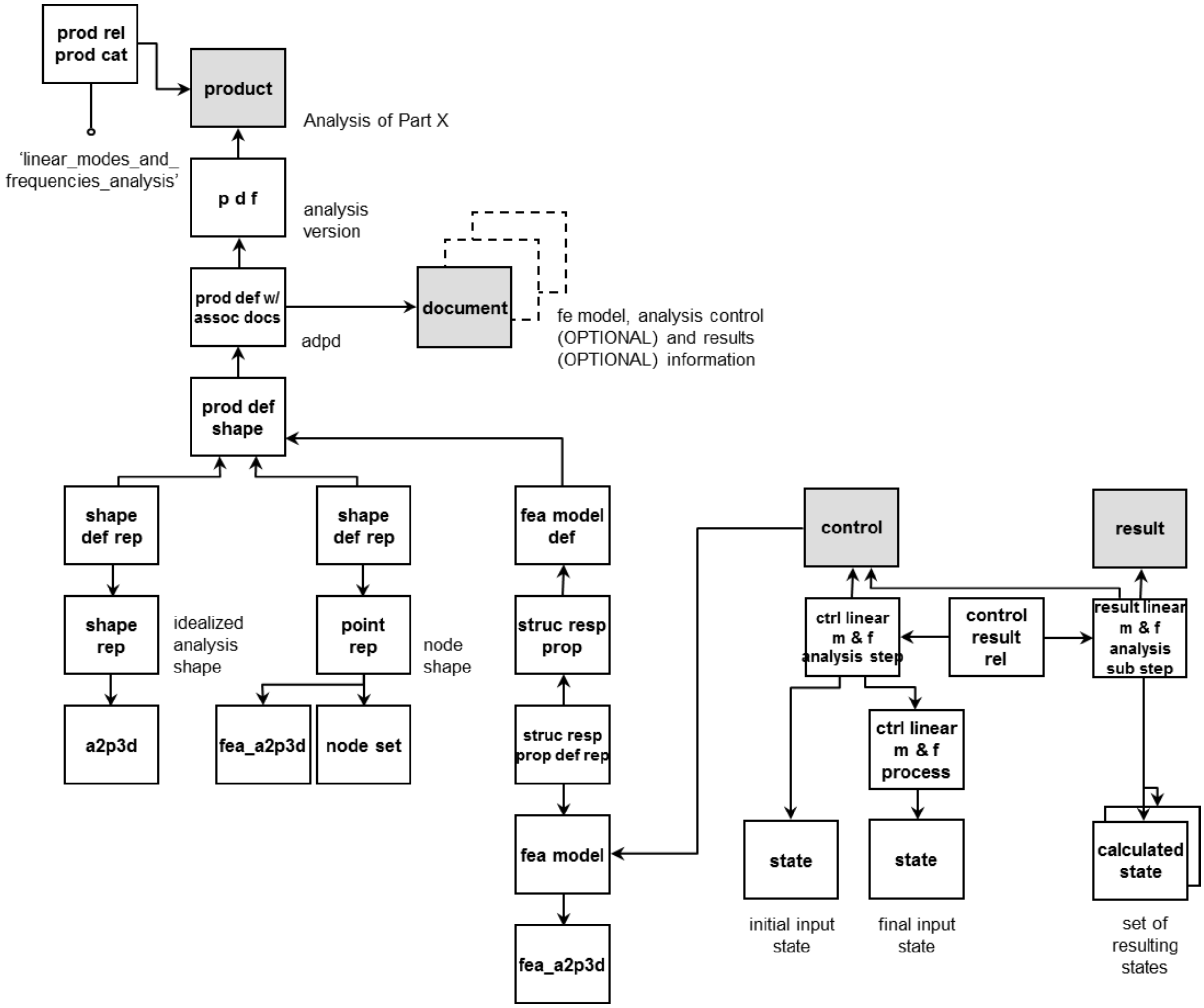


Figure 28 - Linear Modes and Frequencies Analysis - Control and Results

One or more constraints that direct the response of an **fea_model** may be associated with a **control_analysis_step**. Constraints are represented by subtypes of the **constraint_element** entity. The constraint identifier represented by the **element_id** attribute shall be unique within the **fea_model**. Constraints may be applied to a node, a group of nodes, or a geometry element. A constraint that sets values for one or more degrees of freedom at a single node or substructure node is given by a **single_point_ constraint_element**. A multi-point constraint that restricts the nodal freedoms at multiple nodes is given by a **linear_constraint_equation_element**. The reduction of the degrees of freedom for a node, group of nodes, or a geometry element is given by a **nodal_dof_reduction** entity. The reference to a geometry element is established through an **analysis_item_within_representation** entity (Figure 29).

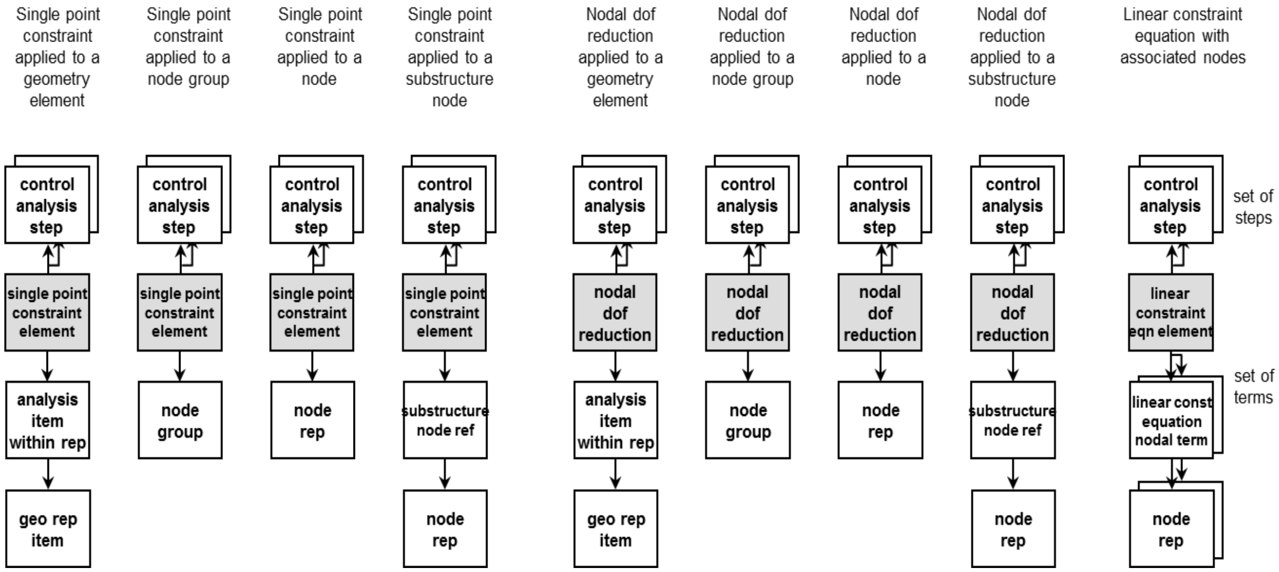

Single point constraint applied to a geometry element
Single point constraint applied to a node group
Single point constraint applied to a node
Single point constraint applied to a substructure node
Nodal dof reduction applied to a geometry element
Nodal dof reduction applied to a node group
Nodal dof reduction applied to a node
Nodal dof reduction applied to a substructure node
Linear constraint equation with associated nodes
control analysis step
set of steps
single point constraint element
nodal dof reduction
linear constraint eqn element
set of terms
analysis item within rep
node group
node rep
substructure node ref
linear const equation nodal term
geo rep item
node rep

Figure 29 - Constraint Definition Reference

2.11.3.3.2. Analysis Results

The analysis results specifying the response an fea_model to a control are represented by the **result** entity. A **result** follows from a single **result_analysis_step** in a linear analysis. A **result_analysis_step** references the **control** through its inherited attribute **analysis_control**. A **result_analysis_step** is related to a **control_ analysis_step** by a **control_result_relationship**.

The **result_linear_static_analysis_sub_step** subtype of the **result_analysis_step** is used in a linear static analysis. The final state resulting from the application of static loads and constraints is represented by a calculated state (Figure 27).

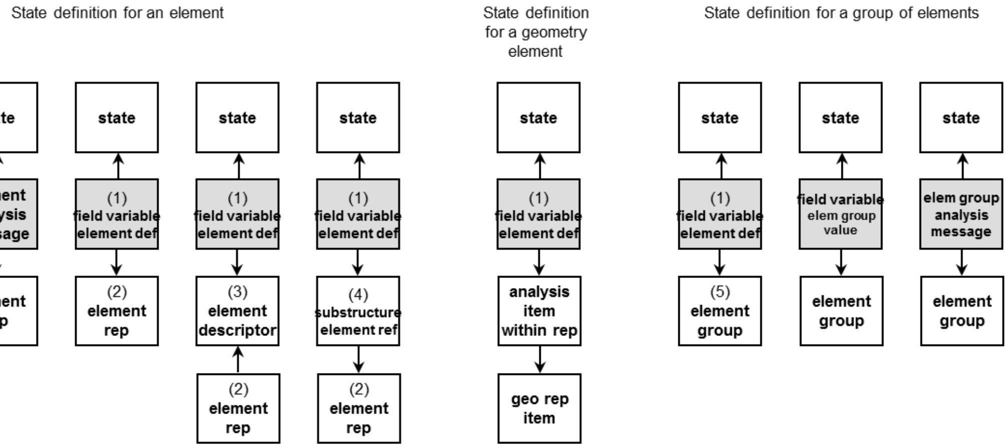
The **result_linear_modes_and_frequencies_analysis_sub_step** subtype of the **result_analysis_step** is used in a linear dynamic analysis. The final equilibrium state resulting from the application of static loads and constraints is represented by is represented by a set of calculated states (Figure 28).

In AP 209 ed2, an FE analysis **result** may have an **approval**. See Section 2.4 for guidance on creating the **approval** and related entities.

2.11.3.3.3. States and State Definitions

A **state** represents the body of information describing or requesting values of the analysis variables of an fea_model. A state can be one of: a) **specified_state** in which all of the information is known a priori; b) **calculated_state** in which information is calculated from a previous analysis step, c) **linearly_ superimposed_state** in which the information is obtained from a combination of previously calculated and/or specified states, or d) **output_request_state** in which information is requested about the variables of the model. An **output_request_state** applies to one or more **control_analysis_step**s.

A **state** is made up of the aggregate of **state_definition**s that reference it. Information about the field variables, prescribed or derived values are specified by specialized subtypes of **state_definition** as described below (see Figure 30 and Figure 31):


State definition for an element
State definition for a geometry element
State definition for a group of elements
state
element nodal freedom actions
element rep
element analysis message
(1) field variable element def
(2) element rep
(3) element descriptor
(4) substructure element ref
analysis item within rep
geo rep item
(5) element group
field variable elem group value
element group
elem group analysis message

(1)	(2)	(3)	(4)	(5)
volume_3d_element_field_value_defn	volume_3d_element_rep	volume_3d_element_descriptor	volume_3d_substructure_element_ref	volume_3d_element_grp
volume_2d_element_field_value_defn	planar_volume_2d_element_rep	planar_volume_2d_element_descriptor	volume_2d_substructure_element_ref	volume_2d_element_grp
volume_2d_element_field_value_defn	axisymmetric_volume_2d_element_rep	volume_2d_element_descriptor	volume_2d_substructure_element_ref	volume_2d_element_grp
surface_3d_element_field_value_defn	surface_3d_element_rep	surface_3d_element_descriptor	surface_3d_substructure_element_ref	surface_3d_element_grp
surface_2d_element_field_value_defn	planar_surface_2d_element_rep	planar_surface_2d_element_descriptor	surface_2d_substructure_element_ref	surface_2d_element_grp
surface_2d_element_field_value_defn	axisymmetric_surface_2d_element_rep	axisymmetric_surface_2d_element_descr	surface_2d_substructure_element_ref	surface_2d_element_grp
curve_3d_element_field_value_defn	curve_3d_element_rep	curve_3d_element_descriptor	curve_3d_substructure_element_ref	curve_3d_element_grp
curve_2d_element_field_value_defn	planar_curve_2d_element_rep	planar_curve_2d_element_descriptor	curve_2d_substructure_element_ref	curve_2d_element_grp
curve_2d_element_field_value_defn	axisymmetric_curve_2d_element_rep	axisymmetric_curve_2d_element_descr	curve_2d_substructure_element_ref	curve_2d_element_grp

Figure 30 - FE State Definition Reference - part 1 of 2

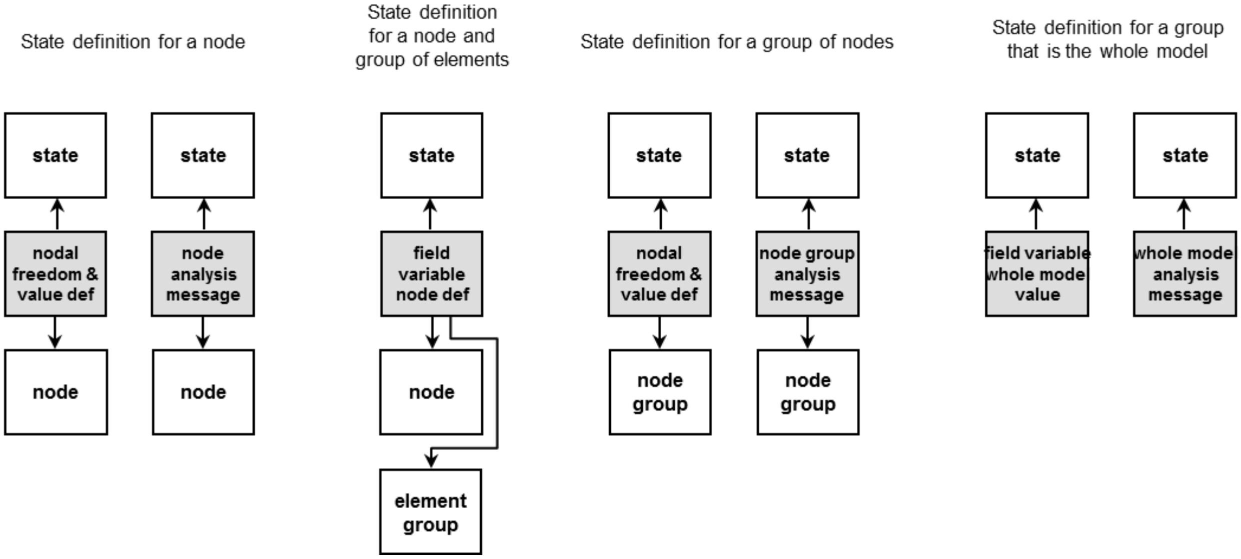

State definition for a node
State definition for a node and group of elements
State definition for a group of nodes
State definition for a group that is the whole model
state
nodal freedom & value def
node
node analysis message
field variable node def
element group
node group
node group analysis message
field variable whole mode value
whole mode analysis message

Figure 31 - FE State Definition Reference - part 2 of 2

Information pertaining to the field variables for an fea_model is given by the **field_variable_definition** subtype of **state_definition**. The **field_variable_definition** entity is further subtyped as it is applied to a node (**field_variable_node_definition**), an element (**field_variable_element_definition**), a group of elements (**field_variable_element_group_value**), or the entire model (**field_variable_whole_ model_value**). The **field_variable_node_definition**, which may also be applied to a node and a group of elements, has specialized subtypes depending on whether the type of element that the node belongs (2D or 3D; volume, surface or curve element). The **field_variable_node_definition** is similarly subtyped depending on the particular **element_representation** it is applied to (2D or 3D; volume, surface or curve element). These subtypes may be applied individually to the respective **element_representation**s, or indirectly through their respective **element_descriptor**. The **field_variable_element_definition** subtypes may also be applied to a group of like elements, or to a geometry element through an **analysis_item_ within_representation**.

Since element locations for field variables are specified in the same manner as those for element properties, refer to Section 2.11.3.1.3 for a discussion on element locations. Similarly, refer to Section 2.11.3.1.4 for a discussion of relating element locations to plies in a laminate table.

Information representing values of nodal action for an element is given by the **element_nodal_ freedom_actions** subtype of **state_definition**. The action is applied to a node by the element.

Information pertaining to the solution degrees of freedom at a node is given by the **nodal_freedom_ and_value_definition** subtype of **state_definition**. The **nodal_freedom_and_value_definition** may be applied to a single node or to a group of nodes.

> NOTE - In specifying nodal actions for elements or solution degrees of freedom for nodes, ISO 10303-104, and therefore AP 209 ed2, provides for the values to be specified separately from the corresponding degrees of freedom. In order to minimize the volume of data to be exchanged, an application should define one instance of **freedoms_list** entity for each of the combinations of the degrees of freedom that are needed. The degrees of freedom for any **element_nodal_feedom_actions, nodal_freedom_and_value_definition, point_freedom_and_value_definition**, **curve_freedom_and_value_definition**, **surface_freedom_and_ value_definition**, and **solid_freedom_and_value_definition** entity can then be specified by referencing the appropriate **freedoms_list** instance.

Finally, information such as notes, warnings, or errors is given by the **analysis_message** subtype of **state_definition**. The **analysis_message** entity is further subtyped as it is attached to a node (**node_ analysis_message**), an element (**element_analysis_message**), a group of elements (**element_group_ analysis_message**), or the entire model (**whole_model_analysis_message**).

2.11.3.3.4. State Relationships

In associating **state_definition**s to the corresponding **state**s, **state_relationship**s may be used to provide an unordered grouping of these **state**s. In this manner, the association of **state_definition**s can be extended to more than one **state**, as in the case of a boundary condition belonging to more than one load case. In such a situation, each boundary condition (that is, **state_definition**) would be associated with its own **specified_state**. Load cases, which are composed of some combination of these boundary conditions, are represented in turn by their own **state**s. **State_relationship**s are then used to relate the **specified_state** for each boundary condition in the load case to the **state** for the load case (see Figure 32).

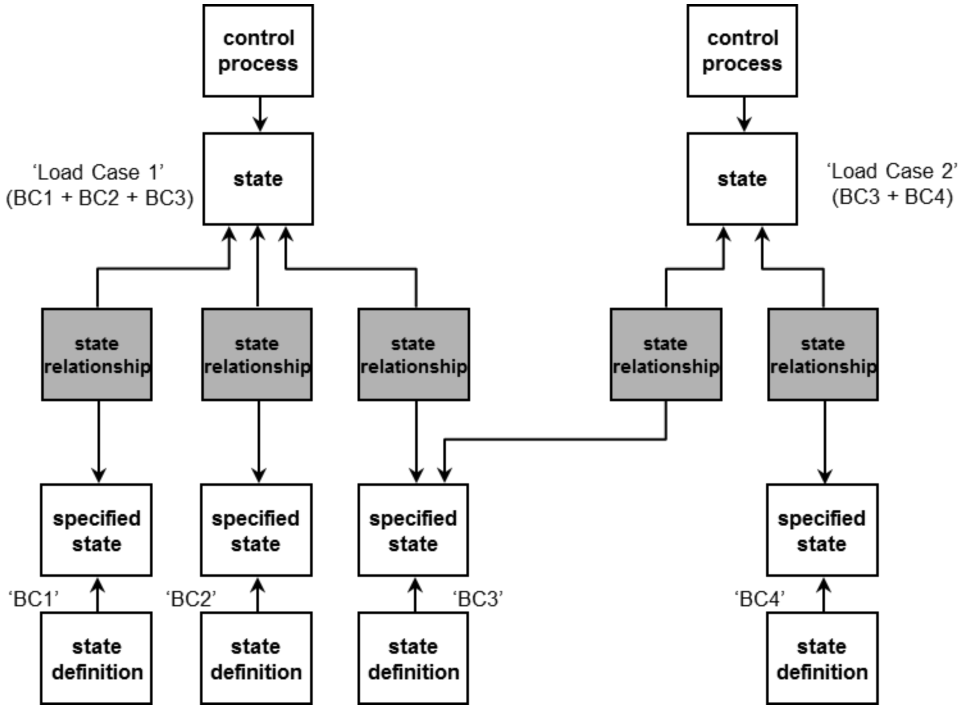


Figure 32 - State Relationships

2.11.3.4. Analysis Reports

Documentation of the fea model, controls and results information is accomplished through **document** entities are attached to the analysis **product_definition** (**product_definition_with_associated_documents**). Tabular or graphical information is indicated by setting the **product_data_type** attribute of the document_type entity referenced by an analysis **document** to **`'tabular report file'`** or **`'graphical report file'`**. The association of analysis reports to an **fea_model** is realized through the **fea_model_definition** shape aspect of the **product_definition_shape** of the analysis **product_definition**. The association of analysis reports to a **control** is in turn realized through referencing of the **fea_model** by the **control**. The association of analysis reports to a result is made through the **result_analysis_step,** which links the **result** to the **control**, as shown in Figure 27 and Figure 28.

Appendix A Physical File Example

This physical file shows the minimum number of entities instances required in an AP 209 ed2 file. This file uses entity long names and is formatted for readability. The actual number of characters needed to represent the data would be considerably less if short names were used and was not formatted for readability.

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION(('THIS IS A SAMPLE AP209 ed2 STEP MODEL'),'2;1');
FILE_NAME('CONCEPTUAL PART EXAMPLE', ‘2016-02-15 T12:00:00',
   ('KEITH HUNTEN'),('NONE'),'NO VERSION',
    'HAND POPULATED','APPROVED BY KEITH A. HUNTEN');
FILE_SCHEMA((AP209_MULTIDISCIPLINARY_ANALYSIS_AND_DESIGN_MIM_LF'));
ENDSEC;
DATA;
#1078=COORDINATED_UNIVERSAL_TIME_OFFSET(6,$,.BEHIND.);
#1079=LOCAL_TIME(12,0,$,#1078);
#1083=PERSON('1111111','HUNTEN','KEITH',(''),$,$);
#1084=ORGANIZATION('CAx IF','LOTAR','');
#1085=PERSON_AND_ORGANIZATION(#1083,#1084);
#1086=CALENDAR_DATE(1999,28,3);
#1087=DATE_AND_TIME(#1086,#1079);
#1102=APPLICATION_CONTEXT('AP209_MULTIDISCIPLINARY_ANALYSIS_AND_DESIGN_MIM_LF');
#1103=APPLICATION_PROTOCOL_DEFINITION('international standard',
      'AP209_MULTIDISCIPLINARY_ANALYSIS_AND_DESIGN_MIM_LF',2014,#1102);
#1104=PRODUCT_CONTEXT('',#1102,'design');
#1105=PRODUCT('2865000-1','REAR PANEL','REAR PANEL FOR BOX',(#1104));
#1106=PRODUCT_RELATED_PRODUCT_CATEGORY('detail','',(#1105));
#1107=PERSON_AND_ORGANIZATION_ROLE('design owner');
#1108=APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT(#1085,#1107,(#1105));
#1109=PRODUCT_CATEGORY('part','');
#1110=PRODUCT_CATEGORY_RELATIONSHIP('','',#1109,#1106);
#1119=PRODUCT_DEFINITION_FORMATION('-','PRE-RELEASE',#1105);
#1120=PERSON_AND_ORGANIZATION_ROLE('design supplier');
#1121=APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT(#1085,#1120,(#1119));
#1128=SECURITY_CLASSIFICATION_LEVEL('unclassified');3
#1129=SECURITY_CLASSIFICATION('','',#1128);
#1130=APPLIED_SECURITY_CLASSIFICATION_ASSIGNMENT(#1129,(#1119));
#1147=PERSON_AND_ORGANIZATION_ROLE('classification officer');
#1148=APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT(#1085,#1147,(#1129));
#1153=DATE_TIME_ROLE('classification date');
#1154=APPLIED_DATE_AND_TIME_ASSIGNMENT(#1087,#1153,(#1129));
#1155=PRODUCT_DEFINITION_CONTEXT('',#1102,'design');
#1156=PRODUCT_DEFINITION('design','',#1119,#1155);
#1175=PERSON_AND_ORGANIZATION_ROLE('creator');
#1176=APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT(#1085,#1175,(#1119,#1156));
#1181=DATE_TIME_ROLE('creation date');
#1182=APPLIED_DATE_AND_TIME_ASSIGNMENT(#1087,#1181,(#1156));
ENDSEC;
END-ISO-10303-21;
```

Appendix B Units Examples

This appendix is provided to give examples of instantiations of AP 209 ed2 global units for both standard international (SI) and English units.

When instantiated in a physical file, global SI units where length is expressed in centimeters with an overall model closure tolerance of .000001 of a centimeter would appear as:

```
#1=DIMENSIONAL_EXPONENTS(0.0,0.0,0.0,0.0,0.0,0.0,0.0);
#13=(LENGTH_UNIT()NAMED_UNIT(*)SI_UNIT(.CENTI.,.METRE.));
#14=(NAMED_UNIT(*)PLANE_ANGLE_UNIT()SI_UNIT($,.RADIAN.));
#15=(NAMED_UNIT(*)SI_UNIT($,.STERADIAN.)SOLID_ANGLE_UNIT());
#16=UNCERTAINTY_MEASURE_WITH_UNIT(LENGTH_MEASURE(0.000001),#13,'','');
/* Context for shape representations */
#17=(GEOMETRIC_REPRESENTATION_CONTEXT(3)
     GLOBAL_UNCERTAINTY_ASSIGNED_CONTEXT((#16))
     GLOBAL_UNIT_ASSIGNED_CONTEXT((#13,#14,#15))
     REPRESENTATION_CONTEXT('',''));
#21=DIMENSIONAL_EXPONENTS(0.0,0.0,0.0,0.0,1.0,0.0,0.0);
#22=(NAMED_UNIT(*)SI_UNIT($,.DEGREE_CELCIUS.)
     THERMODYNAMIC_TEMPERATURE_UNIT());
#23=DIMENSIONAL_EXPONENTS(0.0,1.0,0.0,0.0,0.0,0.0,0.0);
#24=(MASS_UNIT()NAMED_UNIT(*)SI_UNIT($,.GRAM.));
#25=DIMENSIONAL_EXPONENTS(0.0,0.0,1.0,0.0,0.0,0.0,0.0);
#26=(NAMED_UNIT(*)SI_UNIT($,.SECOND.)TIME_UNIT());
/* Context for FEA model */
#27=(GEOMETRIC_REPRESENTATION_CONTEXT(3)
     GLOBAL_UNIT_ASSIGNED_CONTEXT((#13,#14,#15,#22,#24,#26))
     REPRESENTATION_CONTEXT('',''));
```

When instantiated in a physical file, global English units where length is expressed in inches with an overall model closure tolerance of .000001 of an inch would appear as:

```
#1=DIMENSIONAL_EXPONENTS(0.0,0.0,0.0,0.0,0.0,0.0,0.0);
#13=(LENGTH_UNIT()NAMED_UNIT(*)SI_UNIT(.CENTI.,.METRE.));
#14=(NAMED_UNIT(*)PLANE_ANGLE_UNIT()SI_UNIT($,.RADIAN.));
#15=(NAMED_UNIT(*)SI_UNIT($,.STERADIAN.)SOLID_ANGLE_UNIT());
#16=LENGTH_MEASURE_WITH_UNIT(LENGTH_MEASURE(2.54),#13);
#17=DIMENSIONAL_EXPONENTS(1.0,0.0,0.0,0.0,0.0,0.0,0.0);
#18=(CONVERSION_BASED_UNIT('INCH',#16)LENGTH_UNIT()NAMED_UNIT(#17));
#19=UNCERTAINTY_MEASURE_WITH_UNIT(LENGTH_MEASURE(0.000001),#18,'','');
/* Context for shape representations */
#20=(GEOMETRIC_REPRESENTATION_CONTEXT(3)
     GLOBAL_UNCERTAINTY_ASSIGNED_CONTEXT((#19))
     GLOBAL_UNIT_ASSIGNED_CONTEXT((#18,#14,#15))
     REPRESENTATION_CONTEXT('',''));
```

Appendix C Finite Element Model Result Structures

This appendix is a draft collection of discussions, concepts, diagrams and examples resulting from efforts to generate ISO10303-209 e2 compliant structures for NASTRAN FEM results.

The entity diagrams are used to convey the overall concepts and important inheritance relationships. These diagrams were generated using a UML modeling tool (hence the UML look and feel). However, they are not necessarily equivalent to the EXPRESS schema. Many details are omitted for clarity. The reader can use the diagrams to find relevant entities in the ISO10303-209 e2 hypertext documentation.

The instance diagrams give clues as to how the data model is instantiated in the Part21 files. They are simple pictures generated in powerpoint that capture paper sketches of instance relationships and cardinality. They do not adhere to a rigorous mapping specification.

Lastly, fragments of the EXPRESS schema and Part21 file text are included to illustrate key concepts and referencing. The reader should always reference the full EXPRESS schema definition found in the hypertext documentation (**ISO/CD 10303-209 Application protocol: Multidisciplinary analysis and design**).

C.1 Quantity – Nodal Displacement

The displacement output quantity is the deflections at each node in the specified degree of freedom for the node. For general 3D problems, the displacement vector for each node will be the six degrees of freedom in rectilinear space.

C.1.1 NASTRAN Cards

Output deflection for the entire model. "All" is only set specification discussed.

```
$ request deflections for entire model
DISPLACEMENT = ALL
```

C.1.2 Entity Diagram - Nodal Displacement

The inheritance diagram is shown below. The bulk of the attributes are defined in the **nodal_freedom_and_value_definition** entity. The **nodal_freedom_action_defintion** leaf is used for specifying applied loads at the nodes and is not discussed here. The **nodal_freedom_values** leaf is used to specify output request and results.

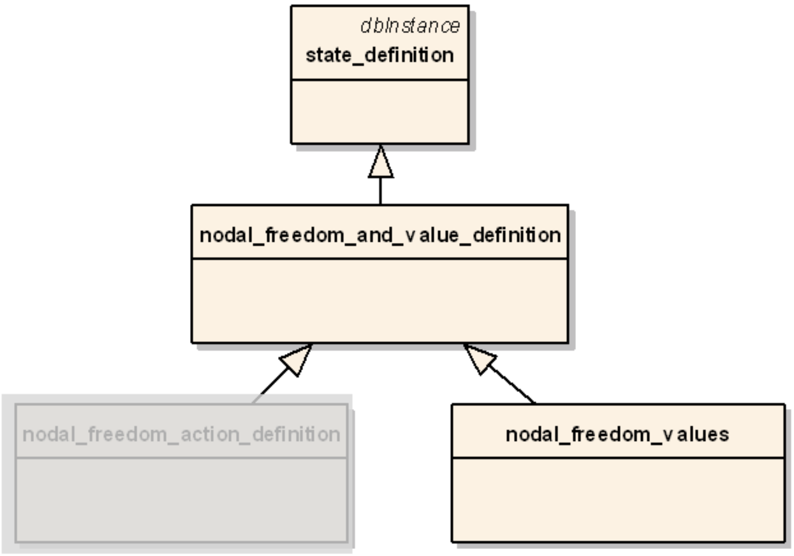

dbInstance
state_definition
nodal_freedom_and_value_definition
nodal_freedom_action_definition
nodal_freedom_values

Figure C.1 - Inheritance diagram for nodal displacement

C.1.3 EXPRESS Schema Text – Nodal Displacement

The relevant EXPRESS entity, type and enumeration specifications are listed. Red text is the selected subtype, select choice or enumeration. The green highlights identify the attributes, and the blue text indicates the type of the attribute..

```
ENTITY state_definition
  SUPERTYPE OF (ONEOF (field_variable_definition,
                       nodal_freedom_and_value_definition,
                       element_nodal_freedom_actions,
                       point_freedom_and_value_definition,
                       curve_freedom_and_value_definition,
                       surface_freedom_and_value_definition,
                       solid_freedom_and_value_definition,
                       linear_constraint_equation_element_value,
                       single_point_constraint_element_values,
                       analysis_message));
  defined_state : state;
END_ENTITY;

ENTITY nodal_freedom_and_value_definition
  SUPERTYPE OF (ONEOF (nodal_freedom_values,
                       nodal_freedom_action_definition))
  SUBTYPE OF (state_definition);
  node : node_output_reference;
  coordinate_system : fea_axis2_placement_3d;
  degrees_of_freedom : freedoms_list;
  values : LIST[1:?] OF measure_or_unspecified_value;
WHERE
  WR1: SIZEOF(degrees_of_freedom.freedoms) = SIZEOF (values);
END_ENTITY;

ENTITY nodal_freedom_values
  SUBTYPE OF (nodal_freedom_and_value_definition);
END_ENTITY;

TYPE node_output_reference = SELECT
   (analysis_item_within_representation,
    node_group,
    node_representation,
    substructure_node_reference);
END_TYPE;

ENTITY freedoms_list;
  freedoms : LIST[1:?] OF degree_of_freedom;
END_ENTITY;
```

C.1.4 Instance Diagram – Nodal Displacement

The AP 209 e2 instance names are presented in this simplified diagram as an example of the relationship between the **nodal_freedom_values** that represent the output request and the **nodal_freedom_values** that represent the actual deflection results data at each degree of freedom at each node [i].

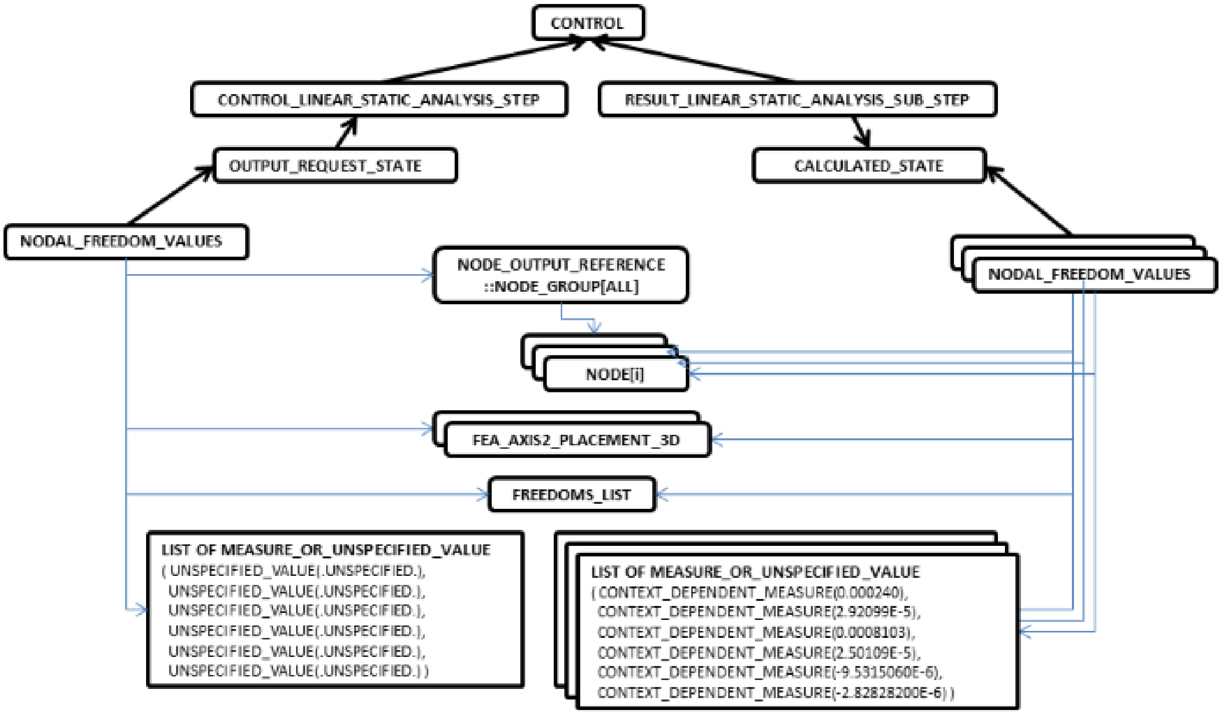

CONTROL
CONTROL_LINEAR_STATIC_ANALYSIS_STEP
RESULT_LINEAR_STATIC_ANALYSIS_SUB_STEP
OUTPUT_REQUEST_STATE
CALCULATED_STATE
NODAL_FREEDOM_VALUES
NODE_OUTPUT_REFERENCE
::NODE_GROUP[ALL]
NODAL_FREEDOM_VALUES
NODE[i]
FEA_AXIS2_PLACEMENT_3D
FREEDOMS_LIST
LIST OF MEASURE_OR_UNSPECIFIED_VALUE
(UNSPECIFIED_VALUE(.UNSPECIFIED.),
UNSPECIFIED_VALUE(.UNSPECIFIED.),
UNSPECIFIED_VALUE(.UNSPECIFIED.),
UNSPECIFIED_VALUE(.UNSPECIFIED.),
UNSPECIFIED_VALUE(.UNSPECIFIED.),
UNSPECIFIED_VALUE(.UNSPECIFIED.))
LIST OF MEASURE_OR_UNSPECIFIED_VALUE
(CONTEXT_DEPENDENT_MEASURE(0.000240),
CONTEXT_DEPENDENT_MEASURE(2.92099E-5),
CONTEXT_DEPENDENT_MEASURE(0.0008103),
CONTEXT_DEPENDENT_MEASURE(2.50109E-5),
CONTEXT_DEPENDENT_MEASURE(-9.5315060E-6),
CONTEXT_DEPENDENT_MEASURE(-2.82828200E-6))

Figure C.2 - Instance diagram for nodal displacement

C.1.5 AP 209 ed2 Instance Text – Nodal Displacement

The color highlighting is an aid to following the entity references. Other methods, such as HTML formatted Part 21 files can be generated but for this document only the the selected attribute references are highlighted. Some additional text formatting (extra carriage returns and tabs) has been performed that maynot be valid Part 21 formatting but it aids in seeing the structure of the entity instances.

```
#637590773= CONTROL(#637538024,'Control.0','FemConvert',
        'MSC.NASTRAN JOB CREATED ON 05-JUL-00 AT 17:49:48',('101',
        'TIME 999999.0','CEND','0',
        '1','2','ENDDATA'),('NASTRAN'));

#637590785= CONTROL_LINEAR_STATIC_ANALYSIS_STEP(#637590773,
        'STATIC STEP 1',1,#637590780,'',#637590786);

/*********************************************************************************/
#637590824= OUTPUT_REQUEST_STATE('Step 1 Output Request State',      'SUBCASE 1',(#637590785));

#637590846= NODAL_FREEDOM_VALUES(#637590824,#637590847,#637538026,#637590849,
        (UNSPECIFIED_VALUE(.UNSPECIFIED.),
        UNSPECIFIED_VALUE(.UNSPECIFIED.),
        UNSPECIFIED_VALUE(.UNSPECIFIED.),
        UNSPECIFIED_VALUE(.UNSPECIFIED.),
        UNSPECIFIED_VALUE(.UNSPECIFIED.),
        UNSPECIFIED_VALUE(.UNSPECIFIED.)));

#637590847= NODE_GROUP('','ALL',#637538024,(#637538817,#637538831,
        #637538835,#637538839,#637538843,#637538847,#637538851,#637538855,
        #637538859,#637538863,#637538867,#637538871,#637538875,#637538879,
        ...  <removed for clarity>
        #637544179,#637544183,#637544187,#637544191,#637544195,#637544199,
        #637544203));

#637590849= FREEDOMS_LIST((
        ENUMERATED_DEGREE_OF_FREEDOM(.X_TRANSLATION.),
        ENUMERATED_DEGREE_OF_FREEDOM(.Y_TRANSLATION.),
        ENUMERATED_DEGREE_OF_FREEDOM(.Z_TRANSLATION.),
        ENUMERATED_DEGREE_OF_FREEDOM(.X_ROTATION.),
        ENUMERATED_DEGREE_OF_FREEDOM(.Y_ROTATION.),
        ENUMERATED_DEGREE_OF_FREEDOM(.Z_ROTATION.)));

/*********************************************************************************/
#637544293= CALCULATED_STATE('Calculated State for Step 1','SUBCASE 1');

#637590884= NODAL_FREEDOM_VALUES(#637544293,#637538817,#637538821,#637590849,
        (CONTEXT_DEPENDENT_MEASURE(0.0002405151),
        CONTEXT_DEPENDENT_MEASURE(2.92099000000000E-5),
        CONTEXT_DEPENDENT_MEASURE(0.0008121154),
        CONTEXT_DEPENDENT_MEASURE(2.50109000000000E-5),
        CONTEXT_DEPENDENT_MEASURE(-9.53150600000000E-6),
        CONTEXT_DEPENDENT_MEASURE(-2.82828200000000E-6)));

#637538817= NODE_WITH_SOLUTION_COORDINATE_SYSTEM('1000',(#637538819,
        #637538821),#637538828,#637538024);
```

C.2 Quantity - Element Stress

The element stress output quantity is represented by a second order tensors in 3D space. The actual representation is dependent upon the requested options found on the NASTRAN stress request case control card. The representation is also dependent upon which dimensional class of element is being output. In all cases, the representation of the stress request must match the results representation. Many of the entities and types used in these representations are shared and reused.

C.2.1 NASTRAN Cards

The stress output requests are shown below. There are other options on the stress card but they do not directly affect the AP 209 representation of element stresses. "All" is only set specification supported at this time.

```
$ A request for top/bottom stresses at the center of elements
STRESS(CENTER) = ALL

$ A request for top/bottom stresses at the corners of elements
STRESS(CORNER) = ALL
```

C.2.2 Entity Diagram - Surface Element Stresses

The high level entity diagram highlights in green the entity hierarchy used for surface element stress representation. The other entities are discussed as in following sections.

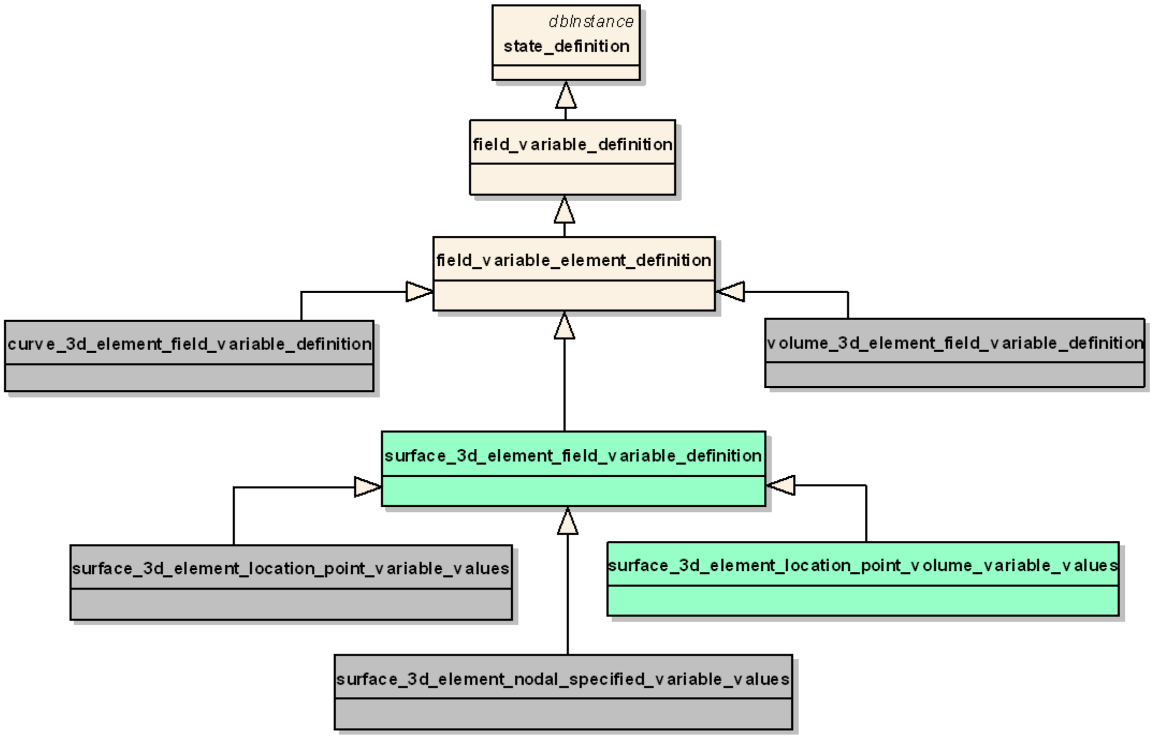
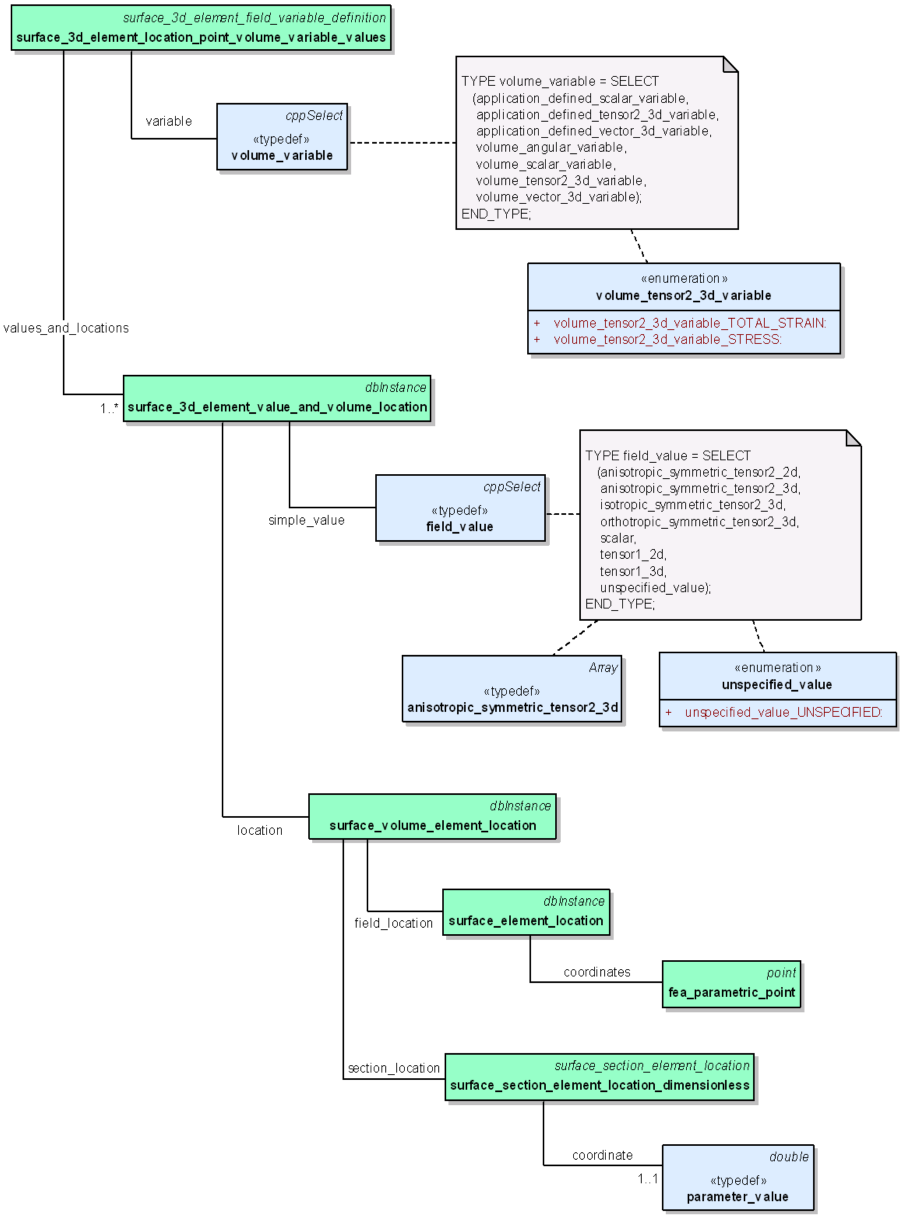

dbInstance
state_definition
field_variable_definition
field_variable_element_definition
curve_3d_element_field_variable_definition
volume_3d_element_field_variable_definition
surface_3d_element_field_variable_definition
surface_3d_element_location_point_variable_values
surface_3d_element_location_point_volume_variable_values
surface_3d_element_nodal_specified_variable_values

Figure C.3 - Inheritance diagram for surface element stresses

The attribute for the element output reference is in the surface_3d_element_field_variable_definition supertype. The type of this attribute is a SELECT type shown in the express text section. Either a group, an element or an element descriptor is typically used.

The **surface_3d_element_location_point_volume_variable_values** subtype specifies the construct that represents data at locations in the 2D plane of the element and at specified locations through the abstracted thickness of the surface element. In general the construct defines the type of output variable (*variable*), the data for the output variable (*simple_value*), and the location of the output variable (*location*). In the diagram below, SELECT type choices are shown in note boxes.



surface_3d_element_field_variable_definition
surface_3d_element_location_point_volume_variable_values
variable
cppSelect
«typedef»
volume_variable
TYPE volume_variable = SELECT
(application_defined_scalar_variable,
application_defined_tensor2_3d_variable,
application_defined_vector_3d_variable,
volume_angular_variable,
volume_scalar_variable,
volume_tensor2_3d_variable,
volume_vector_3d_variable);
END_TYPE;
«enumeration»
volume_tensor2_3d_variable
+ volume_tensor2_3d_variable_TOTAL_STRAIN:
+ volume_tensor2_3d_variable_STRESS:
values_and_locations
1..*
dbInstance
surface_3d_element_value_and_volume_location
simple_value
cppSelect
«typedef»
field_value
TYPE field_value = SELECT
(anisotropic_symmetric_tensor2_2d,
anisotropic_symmetric_tensor2_3d,
isotropic_symmetric_tensor2_3d,
orthotropic_symmetric_tensor2_3d,
scalar,
tensor1_2d,
tensor1_3d,
unspecified_value);
END_TYPE;
Array
«typedef»
anisotropic_symmetric_tensor2_3d
«enumeration»
unspecified_value
+ unspecified_value_UNSPECIFIED:
location
dbInstance
surface_volume_element_location
field_location
dbInstance
surface_element_location
coordinates
point
fea_parametric_point
section_location
surface_section_element_location
surface_section_element_location_dimensionless
coordinate
1..1
double
«typedef»
parameter_value

Figure C.4 - Class diagram for **surface_3d_element_location_point_volume_variable_values**

C.2.3 EXPRESS Schema Text - Surface Element Stresses

The relevant Express entity, type and enumeration specifications are listed. Red text is the selected subtype, select choice or enumeration. The green highlights identify the attributes, and the blue text indicates the type of the attribute.

```
ENTITY state_definition
  SUPERTYPE OF (ONEOF (field_variable_definition,
                       nodal_freedom_and_value_definition,
                       element_nodal_freedom_actions,
                       point_freedom_and_value_definition,
                       curve_freedom_and_value_definition,
                       surface_freedom_and_value_definition,
                       solid_freedom_and_value_definition,
                       linear_constraint_equation_element_value,
                       single_point_constraint_element_values,
                       analysis_message));
  defined_state : state;
END_ENTITY;

ENTITY field_variable_definition
  SUPERTYPE OF (ONEOF (field_variable_element_definition,
                       field_variable_element_group_value,
                       field_variable_whole_model_value,
                       field_variable_node_definition))
  SUBTYPE OF (state_definition);
END_ENTITY;

ENTITY field_variable_element_definition
  SUPERTYPE OF (ONEOF (volume_3d_element_field_variable_definition,
                       volume_2d_element_field_variable_definition,
                       surface_3d_element_field_variable_definition,
                       surface_2d_element_field_variable_definition,
                       curve_3d_element_field_variable_definition,
                       curve_2d_element_field_variable_definition))
  SUBTYPE OF (field_variable_definition);
END_ENTITY;

ENTITY surface_3d_element_field_variable_definition
  SUPERTYPE OF (ONEOF (surface_3d_element_location_point_volume_variable_values,
                       surface_3d_element_location_point_variable_values,
                       surface_3d_whole_element_variable_value,
                       surface_3d_element_constant_specified_variable_value,
                       surface_3d_element_constant_specified_volume_variable_value,
                       surface_3d_element_nodal_specified_variable_values,
                       surface_3d_element_boundary_location_point_surface_variable_values,
                       surface_3d_element_boundary_whole_face_variable_value,
                       surface_3d_element_boundary_constant_specified_variable_value,
                       surface_3d_element_boundary_constant_specified_surface_variable_value,
                       surface_3d_element_boundary_nodal_specified_variable_values,
                       surface_3d_element_boundary_edge_location_point_surface_variable_values,
                       surface_3d_element_boundary_edge_location_point_variable_values,
                       surface_3d_element_boundary_edge_whole_edge_variable_value,
                       surface_3d_element_boundary_edge_constant_specified_variable_value,
                       surface_3d_element_boundary_edge_constant_specified_surface_variable_value,
                       surface_3d_element_boundary_edge_nodal_specified_variable_values))
  SUBTYPE OF (field_variable_element_definition);
  element : surface_3d_element_output_reference;
END_ENTITY;

TYPE surface_3d_element_output_reference = SELECT
   (analysis_item_within_representation,
    surface_3d_element_descriptor,
    surface_3d_element_group,
    surface_3d_element_representation,
    surface_3d_substructure_element_reference);
END_TYPE;

ENTITY surface_3d_element_location_point_volume_variable_values
  SUBTYPE OF (surface_3d_element_field_variable_definition);
  basis : BOOLEAN;
  values_and_locations : SET[1:?] OF surface_3d_element_value_and_volume_location;
  variable : volume_variable;
WHERE
  WR1: consistent_set_values (values_and_locations, variable);
  WR2: appropriate_set_value_existence (values_and_locations, TYPEOF
(SELF\state_definition.defined_state));
END_ENTITY;

ENTITY surface_3d_element_value_and_volume_location;
  simple_value : field_value;
  location : surface_volume_element_location;
  coordinate_system : OPTIONAL surface_3d_state_coordinate_system;
WHERE
  WR1: necessary_value_coordinate_system (simple_value, coordinate_system);
END_ENTITY;

TYPE field_value = SELECT
   (anisotropic_symmetric_tensor2_2d,
    anisotropic_symmetric_tensor2_3d,
    isotropic_symmetric_tensor2_3d,
    orthotropic_symmetric_tensor2_3d,
    scalar,
    tensor1_2d,
    tensor1_3d,
    unspecified_value);
END_TYPE;

ENTITY surface_volume_element_location;
  field_location : surface_element_location;
  section_location : surface_section_element_location;
END_ENTITY;

ENTITY surface_element_location;
  coordinates : fea_parametric_point;
END_ENTITY;

ENTITY surface_section_element_location
  SUPERTYPE OF (ONEOF (surface_section_element_location_absolute,
                        surface_section_element_location_dimensionless));
  above_material_discontinuity : LOGICAL;
END_ENTITY;

ENTITY surface_section_element_location_dimensionless
  SUBTYPE OF (surface_section_element_location);
  coordinate : LIST[1:1] OF parameter_value;
WHERE
  WR1: valid_parametric_coordinate (coordinate);
END_ENTITY;

TYPE volume_variable = SELECT
   (application_defined_scalar_variable,
    application_defined_tensor2_3d_variable,
    application_defined_vector_3d_variable,
    volume_angular_variable,
    volume_scalar_variable,
    volume_tensor2_3d_variable,
    volume_vector_3d_variable);
END_TYPE;

TYPE volume_tensor2_3d_variable = ENUMERATION OF
  (total_strain,    stress);
END_TYPE;
```

C.2.4 Instance Diagram - Surface Element Stresses

A simplified instance diagram is shown below. The purpose of this diagram is to show which items are reusable between the output request and the calculated state. It also illustrates the results representation and the general structure of the instance graph. Some details are omitted. The **volume_variable** select type is the enumerated value of STRESS. This enumeration is not a separate instance as shown, but is part of the aggregating entity.

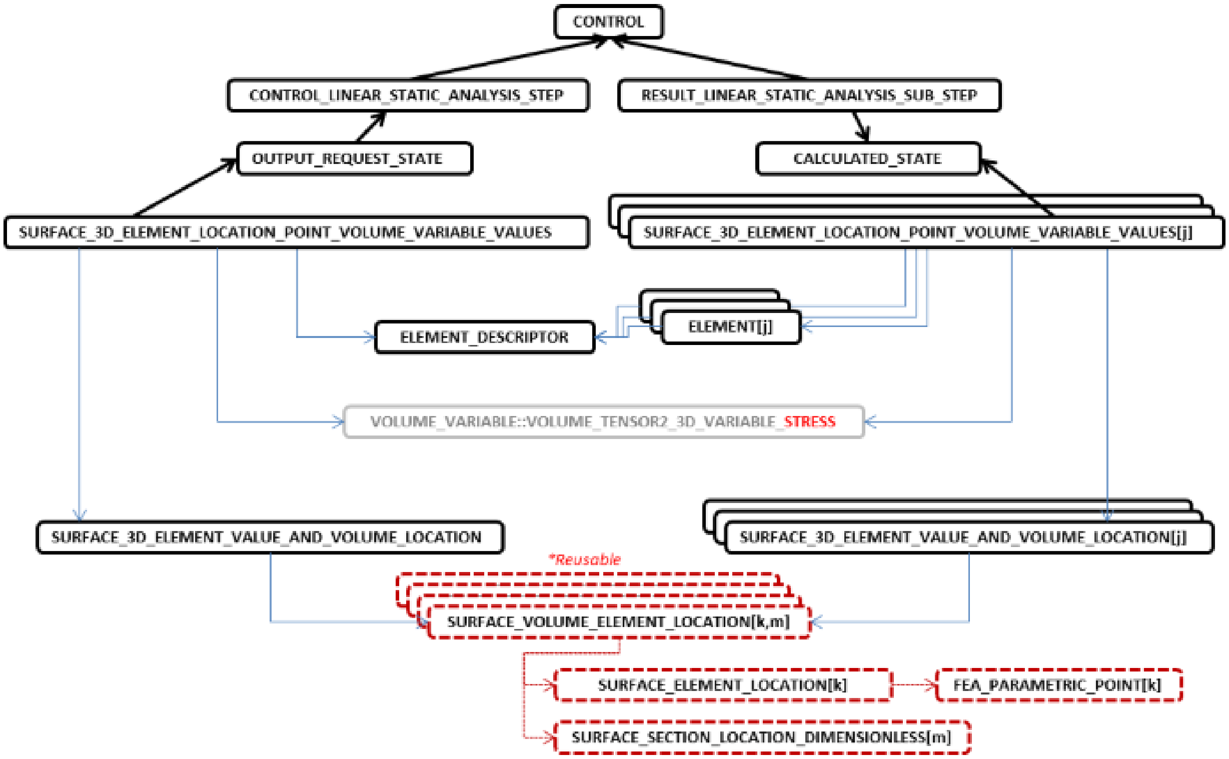

CONTROL
CONTROL_LINEAR_STATIC_ANALYSIS_STEP
RESULT_LINEAR_STATIC_ANALYSIS_SUB_STEP
OUTPUT_REQUEST_STATE
CALCULATED_STATE
SURFACE_3D_ELEMENT_LOCATION_POINT_VOLUME_VARIABLE_VALUES
SURFACE_3D_ELEMENT_LOCATION_POINT_VOLUME_VARIABLE_VALUES[j]
ELEMENT_DESCRIPTOR
ELEMENT[j]
VOLUME_VARIABLE::VOLUME_TENSOR2_3D_VARIABLE_STRESS
SURFACE_3D_ELEMENT_VALUE_AND_VOLUME_LOCATION
SURFACE_3D_ELEMENT_VALUE_AND_VOLUME_LOCATION[j]
*Reusable
SURFACE_VOLUME_ELEMENT_LOCATION[k,m]
SURFACE_ELEMENT_LOCATION[k]
FEA_PARAMETRIC_POINT[k]
SURFACE_SECTION_LOCATION_DIMENSIONLESS[m]

Figure C.5 - Instance diagram for surface element stress

Multiple **surface_volume_element_locations** are created for each element type at the requested location. For example, if quadrilateral element stresses are requested at the corners [k] of the element for the top a bottom surfaces [m], there would be 8 unique instances of **surface_volume_element_location** that would be shared between all the requests and results. Obviously, unique instances **surface_3d_element_value_and_volume_location** are required to house the stress tensor data at each location for each element [j].

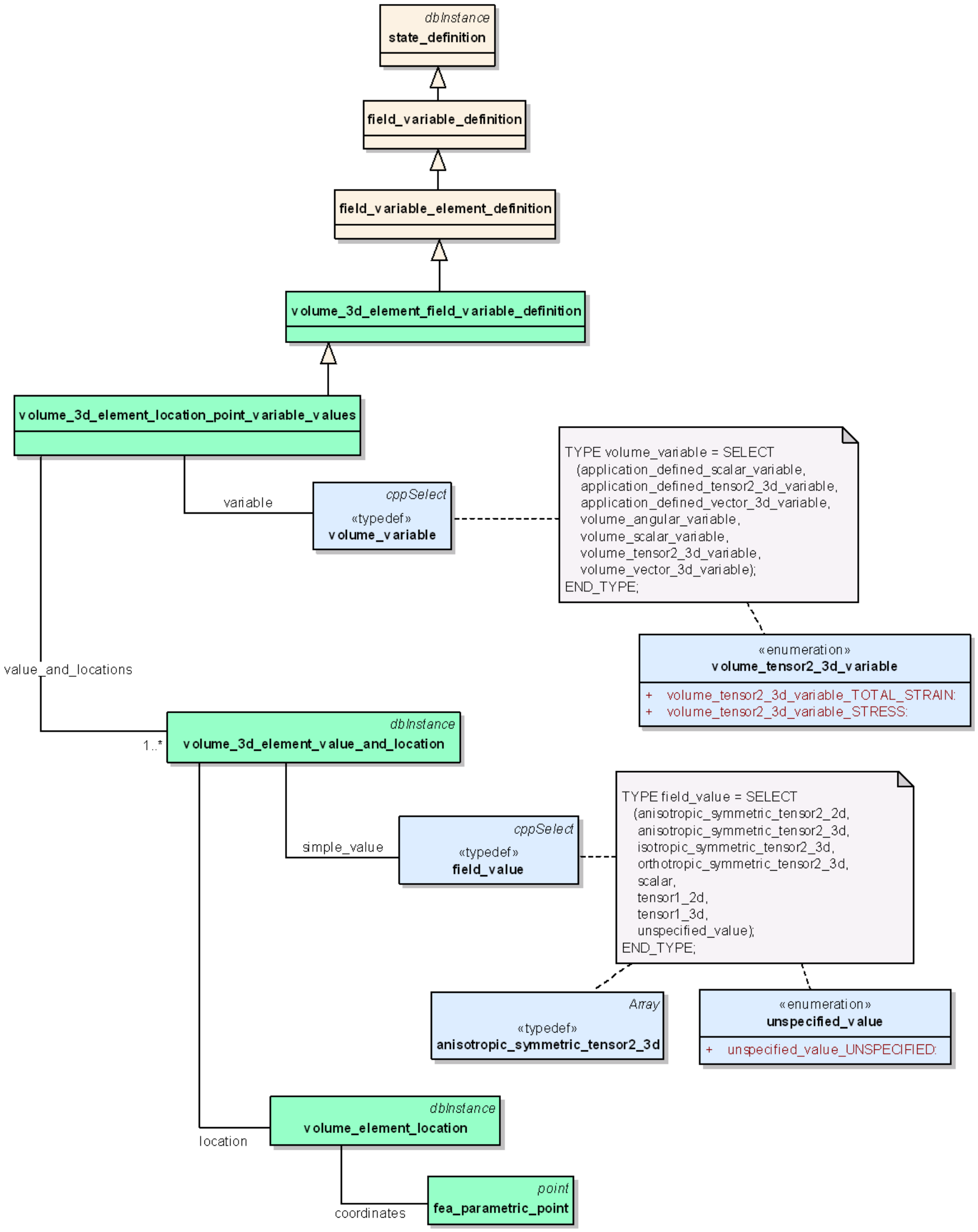
C.2.5 Instance Text - Surface Element Stresses

The color highlighting is an aid to following the entity references. Other methods, such as HTML formatted Part 21 files can be generated but for this document only the the selected attribute references are highlighted. Some additional text formatting has been performed that maynot be valid Part 21 formatting but it aids in seeing the structure of the entity instances.

```
#637590773= CONTROL(#637538024,'Control.0','FemConvert',
        'MSC.NASTRAN JOB CREATED ON 05-JUL-00 AT 17:49:48',('101',
        'TIME 999999.0','CEND','0',
        '1','2','ENDDATA'),('NASTRAN'));
#637590785= CONTROL_LINEAR_STATIC_ANALYSIS_STEP(#637590773,
        'STATIC STEP 1',1,#637590780,'',#637590786);
/******************************************************************************************
* Output request information for CQUAD4 stress tensor at center of element, top and bottom surface
******************************************************************************************/
#637590824= OUTPUT_REQUEST_STATE('Step 1 Output Request State',        'SUBCASE 1',(#637590785));

#637590842= SURFACE_3D_ELEMENT_LOCATION_POINT_VOLUME_VARIABLE_VALUES(#637590824,#637544703,.F.,
        (#637590844,#637590845),VOLUME_TENSOR2_3D_VARIABLE(.STRESS.));
#637544703= SURFACE_3D_ELEMENT_DESCRIPTOR(.LINEAR.,'LINEAR_QUADRILATERAL.CQUAD4',
        ((ENUMERATED_SURFACE_ELEMENT_PURPOSE(.MEMBRANE_DIRECT.),
        ENUMERATED_SURFACE_ELEMENT_PURPOSE(.MEMBRANE_SHEAR.)),
        (ENUMERATED_SURFACE_ELEMENT_PURPOSE(.BENDING_DIRECT.),
        ENUMERATED_SURFACE_ELEMENT_PURPOSE(.BENDING_TORSION.)),
        (ENUMERATED_SURFACE_ELEMENT_PURPOSE(.NORMAL_TO_PLANE_SHEAR.))),
        .QUADRILATERAL.);

#637590844= SURFACE_3D_ELEMENT_VALUE_AND_VOLUME_LOCATION(
        UNSPECIFIED_VALUE(.UNSPECIFIED.),#637544712,$);
#637590845= SURFACE_3D_ELEMENT_VALUE_AND_VOLUME_LOCATION(
        UNSPECIFIED_VALUE(.UNSPECIFIED.),#637544720,$);

#637544712= SURFACE_VOLUME_ELEMENT_LOCATION(#637544713,#637544716);
#637544720= SURFACE_VOLUME_ELEMENT_LOCATION(#637544713,#637544721);

#637544713= SURFACE_ELEMENT_LOCATION(#637544714);
#637544714= FEA_PARAMETRIC_POINT('sQuadN_0',(0.,0.,0.));

#637544716= SURFACE_SECTION_ELEMENT_LOCATION_DIMENSIONLESS(.F.,(-1.));
#637544721= SURFACE_SECTION_ELEMENT_LOCATION_DIMENSIONLESS(.F.,(1.));
/******************************************************************************************
*Calculated state data for CQUAD4 element stress tensor at center of element, top and bottom surface
******************************************************************************************/
#637544293= CALCULATED_STATE('Calculated State for Step 1','SUBCASE 1');

#637547777= SURFACE_3D_ELEMENT_LOCATION_POINT_VOLUME_VARIABLE_VALUES(#637544293,#637547768,.F.,
        (#637547779,#637547781),VOLUME_TENSOR2_3D_VARIABLE(.STRESS.));

#637547768= SURFACE_3D_ELEMENT_REPRESENTATION('2481',(#637544701),#637544265,
        (#637541667,#637542119,#637542639,#637543967),#637538024,#637544703,#637544273,#637544249);

#637547779= SURFACE_3D_ELEMENT_VALUE_AND_VOLUME_LOCATION(
        ANISOTROPIC_SYMMETRIC_TENSOR2_3D((-1.149405,-1.691526,0.,-1.042304,0.,0.)),#637544712,#637544701);
#637547781= SURFACE_3D_ELEMENT_VALUE_AND_VOLUME_LOCATION(
        ANISOTROPIC_SYMMETRIC_TENSOR2_3D((-0.2547834,-1.85213,0.,1.237044,0.,0.)),#637544720,#637544701)
```

C.2.6 Entity Diagram - Volume Element Stresses

The high level entity diagram highlights in green the entity hierarchy used for volume element stress representation. The other entities are discussed as in following sections.


dbInstance
state_definition
field_variable_definition
field_variable_element_definition
volume_3d_element_field_variable_definition
volume_3d_element_location_point_variable_values
variable
cppSelect
«typedef»
volume_variable
«enumeration»
volume_tensor2_3d_variable
+ volume_tensor2_3d_variable_TOTAL_STRAIN:
+ volume_tensor2_3d_variable_STRESS:
value_and_locations
1..*
dbInstance
volume_3d_element_value_and_location
simple_value
cppSelect
«typedef»
field_value
Array
«typedef»
anisotropic_symmetric_tensor2_3d
«enumeration»
unspecified_value
+ unspecified_value_UNSPECIFIED:
location
dbInstance
volume_element_location
coordinates
point
fea_parametric_point

```
TYPE volume_variable = SELECT
  (application_defined_scalar_variable,
   application_defined_tensor2_3d_variable,
   application_defined_vector_3d_variable,
   volume_angular_variable,
   volume_scalar_variable,
   volume_tensor2_3d_variable,
   volume_vector_3d_variable);
END_TYPE;
```

```
TYPE field_value = SELECT
  (anisotropic_symmetric_tensor2_2d,
   anisotropic_symmetric_tensor2_3d,
   isotropic_symmetric_tensor2_3d,
   orthotropic_symmetric_tensor2_3d,
   scalar,
   tensor1_2d,
   tensor1_3d,
   unspecified_value);
END_TYPE;
```

Figure C.6 - Class and inheritance diagram for **volume_3d_element_location_point_variable_values**

In general the construct defines the type of output variable (*variable*), the data for the output variable (*simple_value*), and the location of the output variable (*location*). This organization is very similar to the surface elements stresses but with different entity types used to aggregate the lower level types.

C.2.7 Express Text - Volume Element Stresses

The relevant Express entity, type and enumeration specifications are listed. Red text is the selected subtype, select choice or enumeration. The green highlights identify the attributes, and the blue text indicates the type of the attribute.

```
ENTITY state_definition
  SUPERTYPE OF (ONEOF (field_variable_definition,
                       nodal_freedom_and_value_definition,
                       element_nodal_freedom_actions,
                       point_freedom_and_value_definition,
                       curve_freedom_and_value_definition,
                       surface_freedom_and_value_definition,
                       solid_freedom_and_value_definition,
                       linear_constraint_equation_element_value,
                       single_point_constraint_element_values,
                       analysis_message));
  defined_state : state;
END_ENTITY;

ENTITY field_variable_definition
  SUPERTYPE OF (ONEOF (field_variable_element_definition,
                       field_variable_element_group_value,
                       field_variable_whole_model_value,
                       field_variable_node_definition))
  SUBTYPE OF (state_definition);
END_ENTITY;

ENTITY field_variable_element_definition
  SUPERTYPE OF (ONEOF (volume_3d_element_field_variable_definition,
                       volume_2d_element_field_variable_definition,
                       surface_3d_element_field_variable_definition,
                       surface_2d_element_field_variable_definition,
                       curve_3d_element_field_variable_definition,
                       curve_2d_element_field_variable_definition))
  SUBTYPE OF (field_variable_definition);
END_ENTITY;

ENTITY volume_3d_element_field_variable_definition
  SUPERTYPE OF (ONEOF (volume_3d_element_location_point_variable_values,
                       volume_3d_whole_element_variable_value,
                       volume_3d_element_constant_specified_variable_value,
                       volume_3d_element_nodal_specified_variable_values,
                       volume_3d_element_boundary_location_point_variable_values,
                       volume_3d_element_boundary_whole_face_variable_value,
                       volume_3d_element_boundary_constant_specified_variable_value,
                       volume_3d_element_boundary_nodal_specified_variable_values,
                       volume_3d_element_boundary_edge_location_point_volume_variable_values,
                       volume_3d_element_boundary_edge_whole_edge_variable_value,
                       volume_3d_element_boundary_edge_constant_specified_volume_variable_value,
                       volume_3d_element_boundary_edge_nodal_specified_variable_values))
  SUBTYPE OF (field_variable_element_definition);
  element : volume_3d_element_output_reference;
END_ENTITY;

ENTITY volume_3d_element_location_point_variable_values
  SUBTYPE OF (volume_3d_element_field_variable_definition);
  basis : BOOLEAN;
  values_and_locations : SET[1:?] OF volume_3d_element_value_and_location;
  variable : volume_variable;
WHERE
  WR1: consistent_set_values (values_and_locations, variable);
  WR2:          appropriate_set_value_existence          (values_and_locations,          TYPEOF
(SELF\state_definition.defined_state));
END_ENTITY;
```

```
ENTITY volume_3d_element_value_and_location;
  simple_value : field_value;
  location : volume_element_location;
  coordinate_system : OPTIONAL volume_3d_element_coordinate_system;
WHERE
  WR1: necessary_value_coordinate_system (simple_value, coordinate_system);
END_ENTITY;

ENTITY volume_element_location;
  coordinates : fea_parametric_point;
END_ENTITY;

TYPE volume_3d_element_output_reference = SELECT
   (analysis_item_within_representation,
    volume_3d_element_descriptor,
    volume_3d_element_group,
    volume_3d_element_representation,
    volume_3d_substructure_element_reference);
END_TYPE;
TYPE field_value = SELECT
   (anisotropic_symmetric_tensor2_2d,
    anisotropic_symmetric_tensor2_3d,
    isotropic_symmetric_tensor2_3d,
    orthotropic_symmetric_tensor2_3d,
    scalar,
    tensor1_2d,
    tensor1_3d,
    unspecified_value);
END_TYPE;

TYPE volume_variable = SELECT
   (application_defined_scalar_variable,
    application_defined_tensor2_3d_variable,
    application_defined_vector_3d_variable,
    volume_angular_variable,
    volume_scalar_variable,
    volume_tensor2_3d_variable,
    volume_vector_3d_variable);
END_TYPE;

TYPE volume_tensor2_3d_variable = ENUMERATION OF
  (total_strain,    stress);
END_TYPE;
```

C.2.8 Instance Diagram - Volume Element Stresses

A simplified instance diagram is shown below. The purpose of this diagram is to show which items are reusable between the output request and the calculated state. It also illustrates the results representation and the general structure of the instance graph. Some details are omitted. The **volume_variable** select type is the enumerated value of STRESS. This enumeration is not a separate instance as shown, but is part of the aggregating entity.

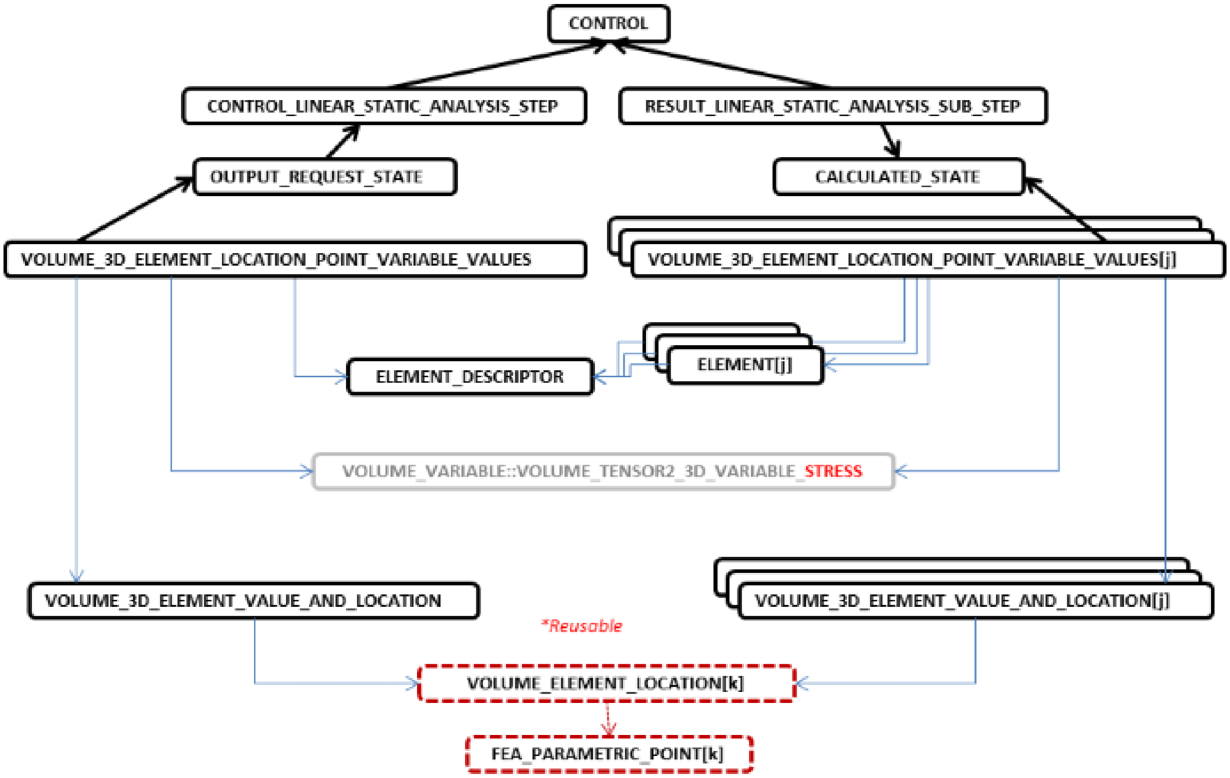

CONTROL
CONTROL_LINEAR_STATIC_ANALYSIS_STEP
RESULT_LINEAR_STATIC_ANALYSIS_SUB_STEP
OUTPUT_REQUEST_STATE
CALCULATED_STATE
VOLUME_3D_ELEMENT_LOCATION_POINT_VARIABLE_VALUES
VOLUME_3D_ELEMENT_LOCATION_POINT_VARIABLE_VALUES[j]
ELEMENT_DESCRIPTOR
ELEMENT[j]
VOLUME_VARIABLE::VOLUME_TENSOR2_3D_VARIABLE_STRESS
VOLUME_3D_ELEMENT_VALUE_AND_LOCATION
VOLUME_3D_ELEMENT_VALUE_AND_LOCATION[j]
*Reusable
VOLUME_ELEMENT_LOCATION[k]
FEA_PARAMETRIC_POINT[k]

Figure C.7 - Instance diagram for volume element stress

Multiple volume_element_locations are created for each element type at the requested location. For example, if hexahedral element stresses are requested, there would be 9 unique instances of volume_element_location that would be shared between all the requests and results. These fea_parametric points are defined for each volume element topology in the Part104 document./ Unique instances volume_3d_element_value_and _location are required to house the stress tensor data at each location for each element [j]. The tensor value entities are not shown on this diagram.

C.2.9 Instance Text - Volume Element Stresses

The color highlighting is an aid to following the entity references. Other methods, such as HTML formatted Part 21 files can be generated but for this document only the the selected attribute references are highlighted. Some additional text formatting has been performed that maynot be valid Part 21 formatting but it aids in seeing the structure of the entity instances.

No instance text is included for volume element stresses at this time.

C.3 Quantity - Element Strain

The element strain output quantity is identical to the stress requests when top and bottom surface strains are requested. If midplane strain and curvature is requested, a different representation is used. In all cases, the representation of the strain request must match the results representation. Many of the entities and types used in these representations are shared and reused.

C.3.1 NASTRAN Cards

The stress output requests are shown below. "All" is only set specification supported at this time. Note that NASTRAN will use the option for center or corner defined on the stress request (if present) in the first subcase (or global case) to set the option used for all subsequent stress, strain and elforce requests.

```
    $ A request for membrane strain and curvatures at the center
    STRAIN(CENTER,STRCUR) = ALL

    $ A request for top/bottom strains at the corners of elements
    STRAIN(CORNER,FIBER) = ALL
```

C.3.2 Entity Diagram - Surface Element Strains

If the FIBER option is used, the only difference between stress and strain representation is the enumerated value for the *variable* attribute. In this case, the TOTAL-STRAIN value is used for the volume_tensor2_3d_variable enumeration. If the STRCUR option is used, the strain data at the midplane surface is reported as a membrane strain and a surface curvature. The high level entity diagram highlights in green the entity hierarchy used for surface element strain representation when midplane strains and curvatures are requested.

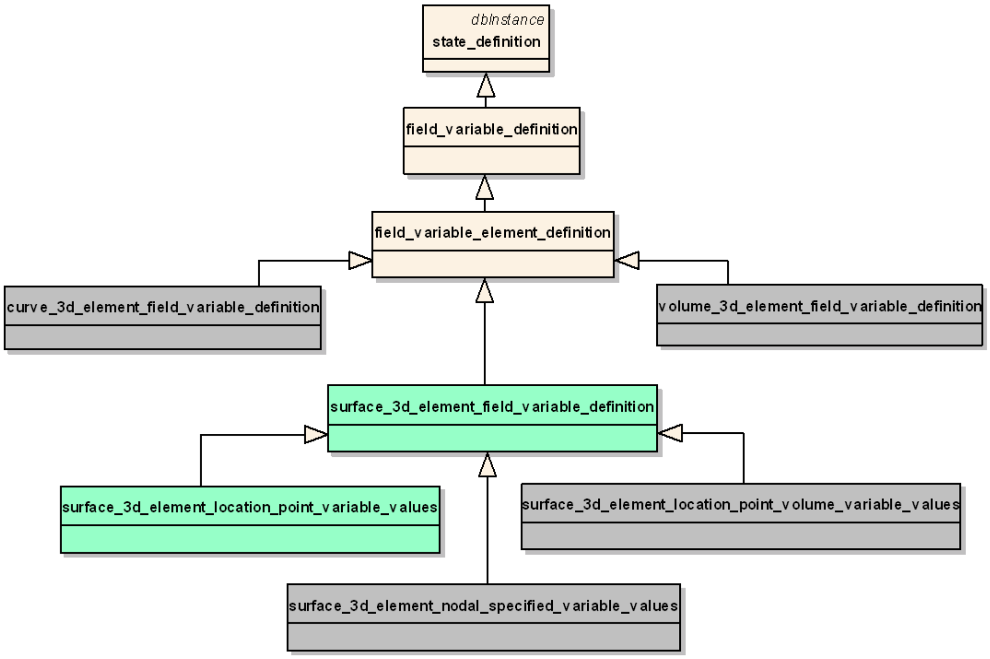
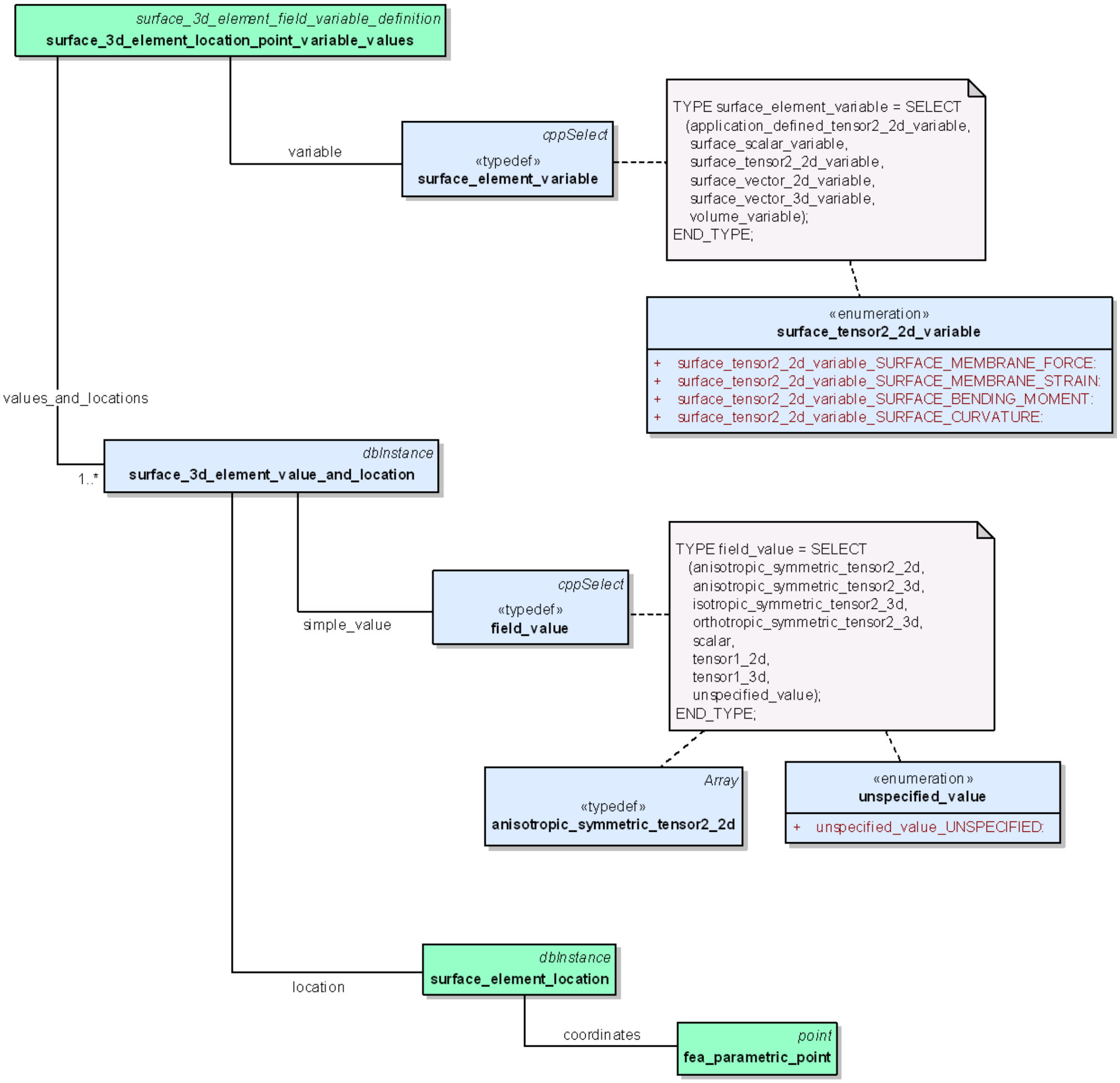

dbInstance
state_definition
field_variable_definition
field_variable_element_definition
curve_3d_element_field_variable_definition
volume_3d_element_field_variable_definition
surface_3d_element_field_variable_definition
surface_3d_element_location_point_variable_values
surface_3d_element_location_point_volume_variable_values
surface_3d_element_nodal_specified_variable_values

Figure C.8 - Inheritance diagram for surface element strains

The attribute for the element output reference is in the surface_3d_element_field_variable_definition supertype. The type of this attribute is a SELECT type shown in the express text section. Either a group, an element or an element descriptor is typically used.

The surface_3d_element_location_point_variable_values subtype specifies the construct that represents data at locations on the 2D midplane of the element. In general the construct defines the type of output variable (*variable*), the data for the output variable (*simple_value*), and the location of the output variable (*location*). In the diagram below, SELECT type choices are shown in note boxes.



surface_3d_element_field_variable_definition
surface_3d_element_location_point_variable_values
variable
cppSelect
«typedef»
surface_element_variable
«enumeration»
surface_tensor2_2d_variable
+ surface_tensor2_2d_variable_SURFACE_MEMBRANE_FORCE:
+ surface_tensor2_2d_variable_SURFACE_MEMBRANE_STRAIN:
+ surface_tensor2_2d_variable_SURFACE_BENDING_MOMENT:
+ surface_tensor2_2d_variable_SURFACE_CURVATURE:
values_and_locations
1..*
dbInstance
surface_3d_element_value_and_location
simple_value
cppSelect
«typedef»
field_value
Array
«typedef»
anisotropic_symmetric_tensor2_2d
«enumeration»
unspecified_value
+ unspecified_value_UNSPECIFIED:
location
dbInstance
surface_element_location
coordinates
point
fea_parametric_point

```
TYPE surface_element_variable = SELECT
  (application_defined_tensor2_2d_variable,
   surface_scalar_variable,
   surface_tensor2_2d_variable,
   surface_vector_2d_variable,
   surface_vector_3d_variable,
   volume_variable);
END_TYPE;
```

```
TYPE field_value = SELECT
  (anisotropic_symmetric_tensor2_2d,
   anisotropic_symmetric_tensor2_3d,
   isotropic_symmetric_tensor2_3d,
   orthotropic_symmetric_tensor2_3d,
   scalar,
   tensor1_2d,
   tensor1_3d,
   unspecified_value);
END_TYPE;
```

Figure C.9 - Class diagram for **surface_3d_element_location_point_variable_values**

C.3.3 Express Text - Surface Element Strains

The relevant Express entity, type and enumeration specifications are listed. Red text is the selected subtype, select choice or enumeration. The green highlights identify the attributes, and the blue text indicates the type of the attribute.

```
ENTITY state_definition
  SUPERTYPE OF (ONEOF (field_variable_definition,
                       nodal_freedom_and_value_definition,
                       element_nodal_freedom_actions,
                       point_freedom_and_value_definition,
                       curve_freedom_and_value_definition,
                       surface_freedom_and_value_definition,
                       solid_freedom_and_value_definition,
                       linear_constraint_equation_element_value,
                       single_point_constraint_element_values,
                       analysis_message));
  defined_state : state;
END_ENTITY;

ENTITY field_variable_definition
  SUPERTYPE OF (ONEOF (field_variable_element_definition,
                       field_variable_element_group_value,
                       field_variable_whole_model_value,
                       field_variable_node_definition))
  SUBTYPE OF (state_definition);
END_ENTITY;

ENTITY field_variable_element_definition
  SUPERTYPE OF (ONEOF (volume_3d_element_field_variable_definition,
                       volume_2d_element_field_variable_definition,
                       surface_3d_element_field_variable_definition,
                       surface_2d_element_field_variable_definition,
                       curve_3d_element_field_variable_definition,
                       curve_2d_element_field_variable_definition))
  SUBTYPE OF (field_variable_definition);
END_ENTITY;

ENTITY surface_3d_element_field_variable_definition
  SUPERTYPE OF (ONEOF (surface_3d_element_location_point_volume_variable_values,
                       surface_3d_element_location_point_variable_values,
                       surface_3d_whole_element_variable_value,
                       surface_3d_element_constant_specified_variable_value,
                       surface_3d_element_constant_specified_volume_variable_value,
                       surface_3d_element_nodal_specified_variable_values,
                       surface_3d_element_boundary_location_point_surface_variable_values,
                       surface_3d_element_boundary_whole_face_variable_value,
                       surface_3d_element_boundary_constant_specified_variable_value,
                       surface_3d_element_boundary_constant_specified_surface_variable_value,
                       surface_3d_element_boundary_nodal_specified_variable_values,
                       surface_3d_element_boundary_edge_location_point_surface_variable_values,
                       surface_3d_element_boundary_edge_location_point_variable_values,
                       surface_3d_element_boundary_edge_whole_edge_variable_value,
                       surface_3d_element_boundary_edge_constant_specified_variable_value,

surface_3d_element_boundary_edge_constant_specified_surface_variable_value,
                       surface_3d_element_boundary_edge_nodal_specified_variable_values))
  SUBTYPE OF (field_variable_element_definition);
  element : surface_3d_element_output_reference;
END_ENTITY;

TYPE surface_3d_element_output_reference = SELECT
   (analysis_item_within_representation,
    surface_3d_element_descriptor,
    surface_3d_element_group,
    surface_3d_element_representation,
    surface_3d_substructure_element_reference);
END_TYPE;

ENTITY surface_3d_element_location_point _variable_values
  SUBTYPE OF (surface_3d_element_field_variable_definition);
  basis : BOOLEAN;
  values_and_locations : SET[1:?] OF surface_3d_element_value_and_location;
  variable : surface_element_variable;
WHERE
  WR1: consistent_set_values (values_and_locations, variable);
  WR2:           appropriate_set_value_existence           (values_and_locations,           TYPEOF
(SELF\state_definition.defined_state));
END_ENTITY;

ENTITY surface_3d_element_value_and_location;
  simple_value : field_value;
  location : surface_element_location;
  coordinate_system : OPTIONAL surface_3d_state_coordinate_system;
WHERE
  WR1: necessary_value_coordinate_system (simple_value, coordinate_system);
END_ENTITY;

TYPE field_value = SELECT
   (anisotropic_symmetric_tensor2_2d,
    anisotropic_symmetric_tensor2_3d,
    isotropic_symmetric_tensor2_3d,
    orthotropic_symmetric_tensor2_3d,
    scalar,
    tensor1_2d,
    tensor1_3d,
    unspecified_value);
END_TYPE;

ENTITY surface_element_location;
  coordinates : fea_parametric_point;
END_ENTITY;

TYPE surface_element_variable = SELECT
   (application_defined_tensor2_2d_variable,
    surface_scalar_variable,
    surface_tensor2_2d_variable,
    surface_vector_2d_variable,
    surface_vector_3d_variable,
    volume_variable);
END_TYPE;

TYPE surface_tensor2_2d_variable = ENUMERATION OF
   (surface_membrane_force,
    surface_membrane_strain,
    surface_bending_moment,
    surface_curvature);
END_TYPE;
```

C.3.4 Instance Diagram - Surface Element Strains

A simplified instance diagram is shown below. The purpose of this diagram is to show which items are reusable between the output request and the calculated state. It also illustrates how the results are represented and the general structure of the instance graph. Some details are omitted.

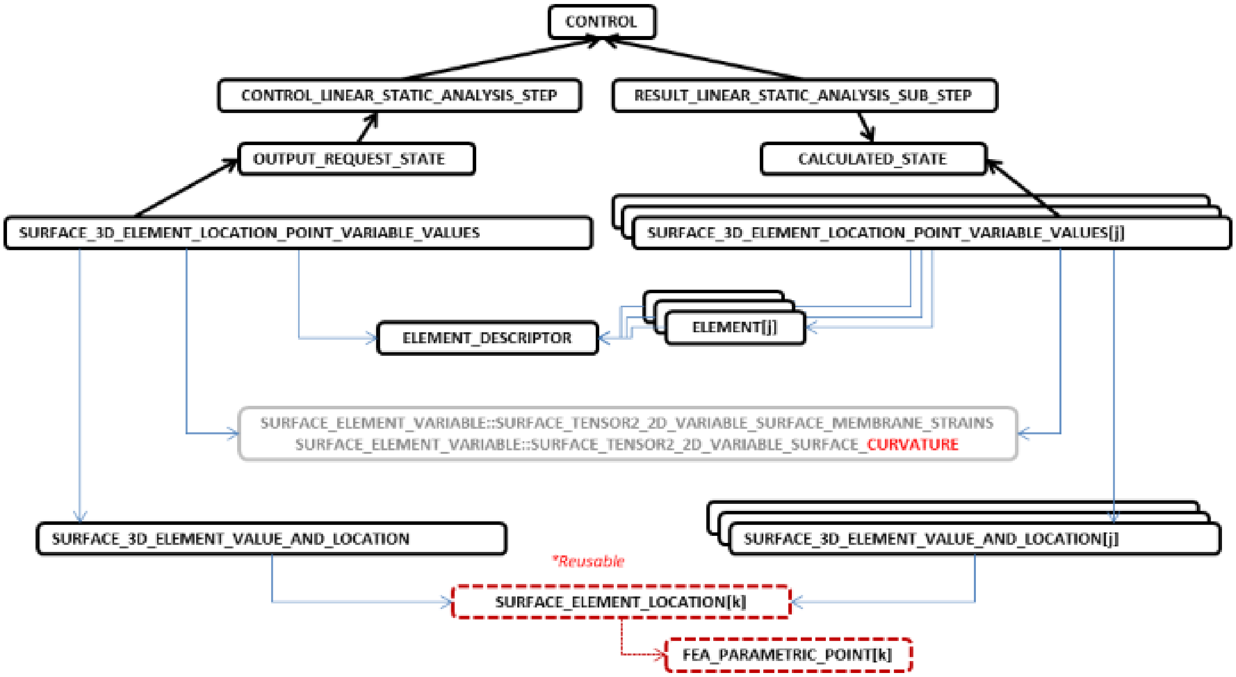

CONTROL
CONTROL_LINEAR_STATIC_ANALYSIS_STEP
RESULT_LINEAR_STATIC_ANALYSIS_SUB_STEP
OUTPUT_REQUEST_STATE
CALCULATED_STATE
SURFACE_3D_ELEMENT_LOCATION_POINT_VARIABLE_VALUES
SURFACE_3D_ELEMENT_LOCATION_POINT_VARIABLE_VALUES[j]
ELEMENT_DESCRIPTOR
ELEMENT[j]
SURFACE_ELEMENT_VARIABLE::SURFACE_TENSOR2_2D_VARIABLE_SURFACE_MEMBRANE_STRAINS
SURFACE_ELEMENT_VARIABLE::SURFACE_TENSOR2_2D_VARIABLE_SURFACE_CURVATURE
SURFACE_3D_ELEMENT_VALUE_AND_LOCATION
SURFACE_3D_ELEMENT_VALUE_AND_LOCATION[j]
*Reusable
SURFACE_ELEMENT_LOCATION[k]
FEA_PARAMETRIC_POINT[k]

Figure C.10 - Instance diagram for surface element strain

Multiple surface_element_locations are created for each element type at the requested location. For example, if quadrilateral element strains and curvature are requested at the corners [k] of the element, there would be 4 unique instances of surface_element_location that would be shared between all the requests and results. Obviously, unique instances surface_3d_element_value_and_location are required to house the strain tensor data at each location for each element [j].

C.3.5 Instance Text - Surface Element Strains

The color highlighting is an aid to following the entity references. Other methods, such as HTML formatted Part 21 files can be generated but for this document only the the selected attribute references are highlighted. Some additional text formatting has been performed that maynot be valid Part 21 formatting but it aids in seeing the structure of the entity instances.

```
#637590773= CONTROL(#637538024,'Control.0','FemConvert',
        'MSC.NASTRAN JOB CREATED ON 05-JUL-00 AT 17:49:48',('101',
        'TIME 999999.0','CEND','0',
        '1','2','ENDDATA'),('NASTRAN'));

#637590779= CONTROL_LINEAR_STATIC_ANALYSIS_STEP(#637590773,
        'STATIC STEP 2',2,#637590780,'',#637590781);
/***************************************************************************************/
* Output request information for CQUAD4 strain and curvature at center of element
***************************************************************************************/
#637590860= OUTPUT_REQUEST_STATE('Step 2 Output Request State','SUBCASE 2',(#637590779));
#637590868= SURFACE_3D_ELEMENT_LOCATION_POINT_VARIABLE_VALUES(#637590860,#637544703,.F.,
        (#637590870),SURFACE_TENSOR2_2D_VARIABLE(.SURFACE_CURVATURE.));
#637590871= SURFACE_3D_ELEMENT_LOCATION_POINT_VARIABLE_VALUES(#637590860,#637544703,.F.,
        (#637590873),SURFACE_TENSOR2_2D_VARIABLE(.SURFACE_MEMBRANE_STRAIN.));

#637544703= SURFACE_3D_ELEMENT_DESCRIPTOR(.LINEAR.,'LINEAR_QUADRILATERAL.CQUAD4',
        ((ENUMERATED_SURFACE_ELEMENT_PURPOSE(.MEMBRANE_DIRECT.),
        ENUMERATED_SURFACE_ELEMENT_PURPOSE(.MEMBRANE_SHEAR.)),
        (ENUMERATED_SURFACE_ELEMENT_PURPOSE(.BENDING_DIRECT.),
        ENUMERATED_SURFACE_ELEMENT_PURPOSE(.BENDING_TORSION.)),
        (ENUMERATED_SURFACE_ELEMENT_PURPOSE(.NORMAL_TO_PLANE_SHEAR.))),
        .QUADRILATERAL.);

#637590870= SURFACE_3D_ELEMENT_VALUE_AND_LOCATION(
        UNSPECIFIED_VALUE(.UNSPECIFIED.),#637544733,$);
#637590873= SURFACE_3D_ELEMENT_VALUE_AND_LOCATION(
        UNSPECIFIED_VALUE(.UNSPECIFIED.),#637544733,$);

#637544733= SURFACE_ELEMENT_LOCATION(#637544734);
#637544734= FEA_PARAMETRIC_POINT('eQuadN_0',(0.,0.,0.));
/***************************************************************************************/
*Calculated state data for CQUAD4 element strain and curvature at center of element
***************************************************************************************/
#637544277= CALCULATED_STATE('Calculated State for Step 2','SUBCASE 2');

#637547783= SURFACE_3D_ELEMENT_LOCATION_POINT_VARIABLE_VALUES(#637544277,#637547768,.F.,
        (#637547785),SURFACE_TENSOR2_2D_VARIABLE(.SURFACE_CURVATURE.));
#637547787= SURFACE_3D_ELEMENT_LOCATION_POINT_VARIABLE_VALUES(#637544277,#637547768,.F.,
        (#637547789),SURFACE_TENSOR2_2D_VARIABLE(.SURFACE_MEMBRANE_STRAIN.));

#637547768= SURFACE_3D_ELEMENT_REPRESENTATION('2481',(#637544701),#637544265,
        (#637541667,#637542119,#637542639,#637543967),#637538024,#637544703,#637544273,#637544249);

#637547785= SURFACE_3D_ELEMENT_VALUE_AND_LOCATION(ANISOTROPIC_SYMMETRIC_TENSOR2_2D(
        (-4.88978200000000E-7,-2.14783100000000E-6,-6.88526500000000E-7)),#637544733,#637544701);
#637547789= SURFACE_3D_ELEMENT_VALUE_AND_LOCATION(ANISOTROPIC_SYMMETRIC_TENSOR2_2D(
        (-1.98397200000000E-7,7.83949000000000E-8,1.93111400000000E-6)),#637544733,#637544701);
```

C.3.6 Entity Diagram - Volume Element Strains

Volume element strains are identical to volume element stresses except for the selected variable type enumeration. For strains, the value is **volume_tensor2_3d_variable_TOTAL_STRAIN**.

C.3.7 Express Text - Volume Element Strains

The Express listing for volume element strains is identical to volume element stresses.

C.3.8 Instance Diagram - Volume Element Strains

The volume element strain instance diagram is identical to the volume element stresses except for the selected variable type enumeration. The value is **volume_tensor2_3d_variable_TOTAL_STRAIN** for strains.

C.3.9 Instance Text - Volume Element Strains

The color highlighting is an aid to following the entity references. Other methods, such as HTML formatted Part 21 files can be generated but for this document only the the selected attribute references are highlighted. Some additional text formatting has been performed that maynot be valid Part 21 formatting but it aids in seeing the structure of the entity instances.

No instance text is included for volume element strains at this time.

C.4 Quantity - Grid Point Force Balance

The grid point force balance output quantity is the forces at each node of each element. The quantity is used to generate free body forces for selected element groups.

C.4.1 NASTRAN Cards

Grid point force balance output for the entire model.

```
$ request grid point force balance data for entire model
GPFORCE = ALL
```

C.4.2 Entity Diagram - Grid Point Force Balance

The diagram shows the top level inheritance and also the lower level associations of the relevant entities.

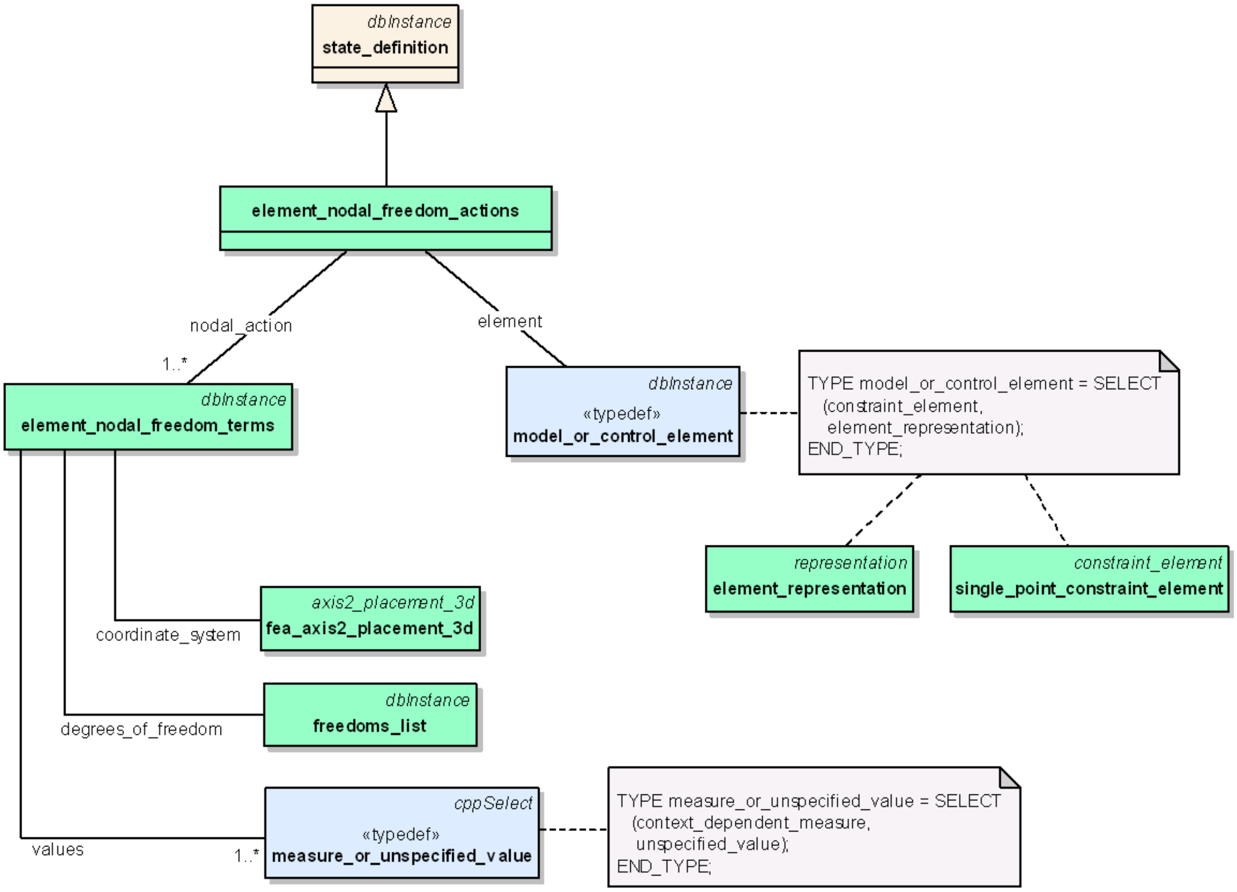


Figure C.11 - Class and inheritance diagram for **element_nodel_freedom_actions**

C.4.3 Express Text - Grid Point Force Balance

The relevant Express entity, type and enumeration specifications are listed. Red text is the selected subtype, select choice or enumeration. The green highlights identify the attributes, and the blue text indicates the type of the attribute.

```
ENTITY state_definition
  SUPERTYPE OF (ONEOF (field_variable_definition,
                       nodal_freedom_and_value_definition,
                       element_nodal_freedom_actions,
                       point_freedom_and_value_definition,
                       curve_freedom_and_value_definition,
                       surface_freedom_and_value_definition,
                       solid_freedom_and_value_definition,
                       linear_constraint_equation_element_value,
                       single_point_constraint_element_values,
                       analysis_message));
  defined_state : state;
END_ENTITY;

ENTITY element_nodal_freedom_actions
  SUBTYPE OF (state_definition);
  element  : model_or_control_element;
  nodal_action : LIST[1:?] OF element_nodal_freedom_terms;
END_ENTITY;

TYPE model_or_control_element = SELECT
   (constraint_element,
    element_representation);
END_TYPE;

ENTITY element_nodal_freedom_terms;
  coordinate_system : fea_axis2_placement_3d;
  degrees_of_freedom : freedoms_list;
  values : LIST[1:?] OF measure_or_unspecified_value;
WHERE
  WR1: SIZEOF(degrees_of_freedom.freedoms) = SIZEOF (values);
END_ENTITY;

ENTITY freedoms_list;
  freedoms  : LIST[1:?] OF degree_of_freedom;
END_ENTITY;

TYPE degree_of_freedom = SELECT
   (application_defined_degree_of_freedom,
    enumerated_degree_of_freedom);
END_TYPE;

TYPE enumerated_degree_of_freedom = ENUMERATION OF
   (x_translation,
    y_translation,
    z_translation,
    x_rotation,
    y_rotation,
    z_rotation,
    warp);
END_TYPE;

TYPE measure_or_unspecified_value = SELECT
   (context_dependent_measure,
    unspecified_value);
END_TYPE;

TYPE context_dependent_measure = REAL;
END_TYPE;
```

C.4.4 Instance Diagram - Grid Point Force Balance

The AP 209 e2 instance diagram shows how the results for the grid point force balance data are represented. As of this writing, the form of the output request for the GPFB data has not been defined. The issue is one of an efficient way to define the requests short of duplicating the entire results structure with a list of **`UNSPECIFIED_VALUE(.UNSPECIFIED.)`** values for every **element_nodal_freedom_terms** instance. Currently, the calculated state data is generated but no matching output request is present.
In the diagram below, the list of **element_nodal_freedom_terms** must match the order of the nodes in the node list of the element connectivity.

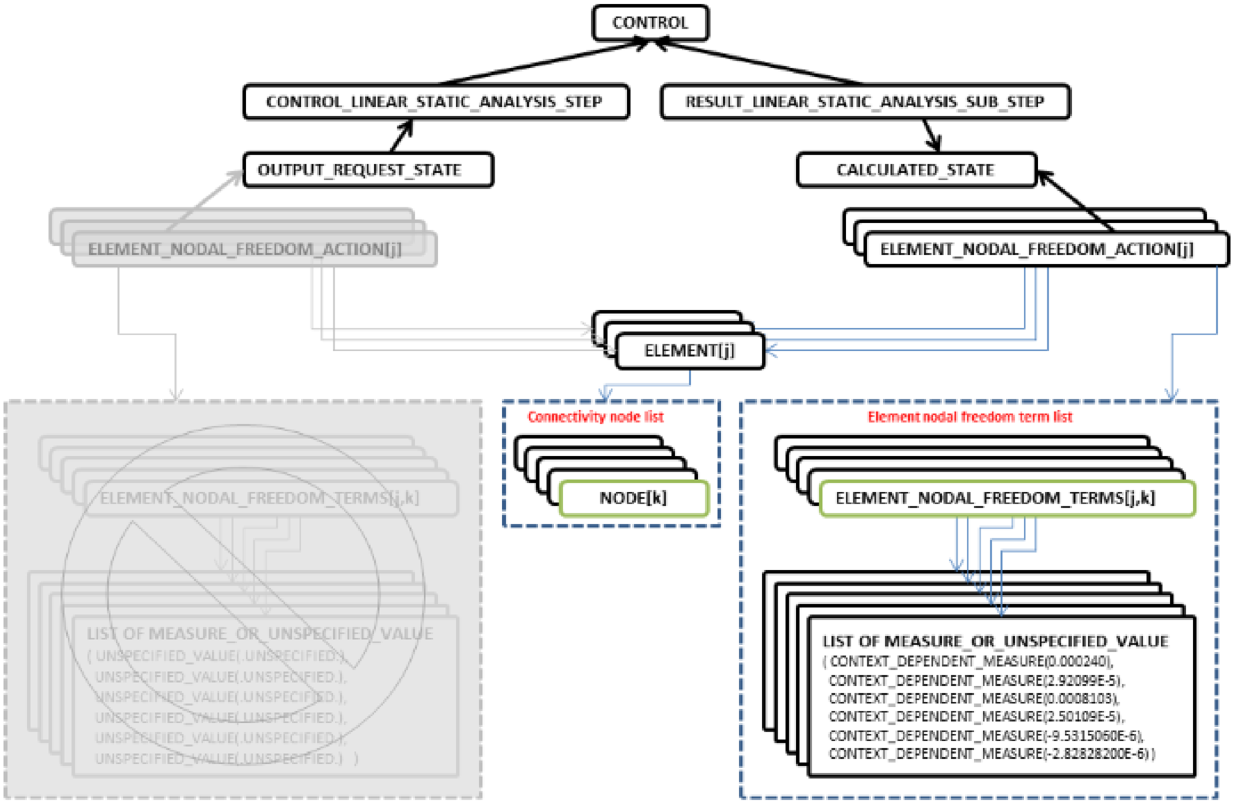

CONTROL
CONTROL_LINEAR_STATIC_ANALYSIS_STEP
RESULT_LINEAR_STATIC_ANALYSIS_SUB_STEP
OUTPUT_REQUEST_STATE
CALCULATED_STATE
ELEMENT_NODAL_FREEDOM_ACTION[j]
ELEMENT_NODAL_FREEDOM_ACTION[j]
ELEMENT[j]
Connectivity node list
NODE[k]
Element nodal freedom term list
ELEMENT_NODAL_FREEDOM_TERMS[j,k]
ELEMENT_NODAL_FREEDOM_TERMS[j,k]
LIST OF MEASURE_OR_UNSPECIFIED_VALUE
(UNSPECIFIED_VALUE(.UNSPECIFIED.),
UNSPECIFIED_VALUE(.UNSPECIFIED.),
UNSPECIFIED_VALUE(.UNSPECIFIED.),
UNSPECIFIED_VALUE(.UNSPECIFIED.),
UNSPECIFIED_VALUE(.UNSPECIFIED.),
UNSPECIFIED_VALUE(.UNSPECIFIED.))
LIST OF MEASURE_OR_UNSPECIFIED_VALUE
(CONTEXT_DEPENDENT_MEASURE(0.000240),
CONTEXT_DEPENDENT_MEASURE(2.92099E-5),
CONTEXT_DEPENDENT_MEASURE(0.0008103),
CONTEXT_DEPENDENT_MEASURE(2.50109E-5),
CONTEXT_DEPENDENT_MEASURE(-9.5315060E-6),
CONTEXT_DEPENDENT_MEASURE(-2.82828200E-6))

Figure C.12 - Instance diagram for grid point force balance

*note - the grayed out items are identical to calculated state structures except for the **`UNSPECIFIED_VALUE(.UNSPECIFIED.)`** values and are not created.

C.4.5 Instance Text - Grid Point Force Balance

The color highlighting is an aid to following the entity references. Other methods, such as HTML formatted Part 21 files can be generated but for this document only the the selected attribute references are highlighted. Some additional text formatting has been performed that maynot be valid Part 21 formatting but it aids in seeing the structure of the entity instances.

```
#637569343= CONTROL(#637538001,'Control.0','FemConvert',
        'MSC/NASTRAN JOB CREATED ON 09-JUN-00 AT 09:35:26',
        ('<SOL>101</SOL>','TIME 600.0','CEND','<GLOBALCASE>0</GLOBALCASE>',
        '<SUBCASE>1</SUBCASE>','<SUBCASE>2</SUBCASE>','ENDDATA'),('NASTRAN'));

#637569349= CONTROL_LINEAR_STATIC_ANALYSIS_STEP(#637569343,'STATIC STEP 2',2,#637569350,'',#637569351);

#637569346= RESULT_LINEAR_STATIC_ANALYSIS_SUB_STEP(#637569343,#637569347,#637539241);
#637569348= CONTROL_RESULT_RELATIONSHIP(#637569349,#637569346);

/***** no output request at this time *****************************************************************/

#637539241= CALCULATED_STATE('Calculated State for Step 2','SUBCASE 2');

#637539232= VOLUME_3D_ELEMENT_REPRESENTATION('368',(#637539234),#637539235,
        (#637538996,#637539000,#637539028,#637539136,#637539140,#637539168),
        #637538001,#637539238,#637539205);

#637539244= FREEDOMS_LIST(
        (ENUMERATED_DEGREE_OF_FREEDOM(.X_TRANSLATION.),
        ENUMERATED_DEGREE_OF_FREEDOM(.Y_TRANSLATION.),
        ENUMERATED_DEGREE_OF_FREEDOM(.Z_TRANSLATION.),
        ENUMERATED_DEGREE_OF_FREEDOM(.X_ROTATION.),
        ENUMERATED_DEGREE_OF_FREEDOM(.Y_ROTATION.),
        ENUMERATED_DEGREE_OF_FREEDOM(.Z_ROTATION.)));

#637539240= ELEMENT_NODAL_FREEDOM_ACTIONS(#637539241,#637539232,
        (#637539243,#637539247,#637539249,#637539251,#637539253,#637539255));

#637539243= ELEMENT_NODAL_FREEDOM_TERMS(#637538003,#637539244,(
        CONTEXT_DEPENDENT_MEASURE(-1.151063),CONTEXT_DEPENDENT_MEASURE(0.4314198),
        CONTEXT_DEPENDENT_MEASURE(0.7986243),CONTEXT_DEPENDENT_MEASURE(0.),
        CONTEXT_DEPENDENT_MEASURE(0.),CONTEXT_DEPENDENT_MEASURE(0.)));

#637539247= ELEMENT_NODAL_FREEDOM_TERMS(#637538003,#637539244,(
        CONTEXT_DEPENDENT_MEASURE(0.9437712),CONTEXT_DEPENDENT_MEASURE(-0.7879028),
        CONTEXT_DEPENDENT_MEASURE(0.2526518),CONTEXT_DEPENDENT_MEASURE(0.),
        CONTEXT_DEPENDENT_MEASURE(0.),  CONTEXT_DEPENDENT_MEASURE(0.)));

#637539249= ELEMENT_NODAL_FREEDOM_TERMS(#637538003,#637539244,(
        CONTEXT_DEPENDENT_MEASURE(-0.4999262),CONTEXT_DEPENDENT_MEASURE(0.555863),
        CONTEXT_DEPENDENT_MEASURE(0.6385123),CONTEXT_DEPENDENT_MEASURE(0.),
        CONTEXT_DEPENDENT_MEASURE(0.),CONTEXT_DEPENDENT_MEASURE(0.)));

#637539251= ELEMENT_NODAL_FREEDOM_TERMS(#637538003,#637539244,(
        CONTEXT_DEPENDENT_MEASURE(-2.688304),CONTEXT_DEPENDENT_MEASURE(-0.08705387),
        CONTEXT_DEPENDENT_MEASURE(-0.0914064),CONTEXT_DEPENDENT_MEASURE(0.),
        CONTEXT_DEPENDENT_MEASURE(0.),CONTEXT_DEPENDENT_MEASURE(0.)));
```

```
#637539253= ELEMENT_NODAL_FREEDOM_TERMS(#637538003,#637539244,(
        CONTEXT_DEPENDENT_MEASURE(3.239961),CONTEXT_DEPENDENT_MEASURE(-0.8987678),
        CONTEXT_DEPENDENT_MEASURE(-1.15925),CONTEXT_DEPENDENT_MEASURE(0.),
        CONTEXT_DEPENDENT_MEASURE(0.),CONTEXT_DEPENDENT_MEASURE(0.)));

#637539255= ELEMENT_NODAL_FREEDOM_TERMS(#637538003,#637539244,(
        CONTEXT_DEPENDENT_MEASURE(0.1555603),CONTEXT_DEPENDENT_MEASURE(0.7864418),
        CONTEXT_DEPENDENT_MEASURE(-0.4391323),CONTEXT_DEPENDENT_MEASURE(0.),
        CONTEXT_DEPENDENT_MEASURE(0.),CONTEXT_DEPENDENT_MEASURE(0.)));.
etc...
```

Appendix D Directionally Explicit Element Layout

This examples discusses the mapping from NASTRAN CBUSH(i) and CELAS(i) elements to the AP 209 ed2 data model. These examples need to be further developed with textual descriptions and diagrams similar to the preceeding Appendix C. They are included here to initiate discussion as to the proper mapping of these directionally explicit elements.

The basic FE model used in this example uses 3 nodes connected by various NASTRAN elastic element representations. This is not an actual executable model but a model fragment that is intended to aid understanding and interpretation of the AP 209 ed2 mapping. The numeric fields are filled with values that represent the type and field location. For example, the first real value on the PBUSH1D entry is populated with 1.1 and the following fields with 2.2 and 3.3. This helps locate the mapped value in the Part 21 file fragment. Other variation of this can be found on the PBUSH continuation lines where values such as 11.1, 11.2, 11.3 etc… represent the positional location of the appropriate real value. Each example may use a slightly different numbering method to aid in verifying the mapping to the AIM.

```
////BASIC MODEL FRAGMENTS////

GRID           1       0.0     0.0     0.0
GRID           2       0.0     0.0     5.0
GRID           3       0.0     0.0     0.0
$
CBUSH1D,100,199,1,2
CBUSH1D,101,199,1,2,0
CBUSH1D,102,199,1,3,0
PBUSH1D,199,1.1,2.2,3.3

CBUSH,200,299,1,2,
CBUSH,201,299,1,2,0.0,1.0,0.0
CBUSH,202,299,1,3,0.0,1.0,0.0,0
PBUSH,299,K,10.1,10.2,10.3,10.4,10.5,10.6
,,B,11.1,11.2,11.3,11.4,11.5,11.6
,,GE,12.1,11.2,12.3,12.4,12.5,12.6
,,RCV,1.1,1.2,1.3,1.4

////NOTES: the identifier used in the deer is constructed as follows: AAA.B.C
////where AAA is the nastran element id, .B is the step instance, .C is the degree of freedom
////matrix enumeration has been fixed on cbush elements

////CBUSH1D////

#637538669= DIRECTIONALLY_EXPLICIT_ELEMENT_COORDINATE_SYSTEM_ALIGNED('',#637538670);
#637538670= PARAMETRIC_CURVE_3D_ELEMENT_COORDINATE_SYSTEM('',#637538671);
#637538671= PARAMETRIC_CURVE_3D_ELEMENT_COORDINATE_DIRECTION('',#637538672);
#637538672= DIRECTION('unspecified',(1.,0.,0.));

#637538842= SYSTEM_AND_FREEDOM(#637538669,ENUMERATED_DEGREE_OF_FREEDOM(.X_TRANSLATION.));
#637538846= SYSTEM_AND_FREEDOM(#637538669,ENUMERATED_DEGREE_OF_FREEDOM(.X_TRANSLATION.));
#637538843=
DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('100.2.1',$,$,(#637538253,#637538292),#637538282,(#
637538846,#637538842),#637538847);
#637538847= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT($,3.3);
#637538848=
DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('100.1.1',$,$,(#637538253,#637538292),#637538282,(#
637538846,#637538842),#637538851);
#637538851= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT($,2.2);
#637538852=
DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('100.0.1',$,$,(#637538253,#637538292),#637538282,(#
637538846,#637538842),#637538855);
#637538855= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT($,1.1);
```

```
////CBUSH////

#637538284=
FEA_AXIS2_PLACEMENT_3D('0',#637538285,#637538287,#637538289,.CARTESIAN.,'FEA_BASIC_COORD_SYSTEM.0
');
#637538856= DIRECTIONALLY_EXPLICIT_ELEMENT_COORDINATE_SYSTEM_ARBITRARY('',#637538284);

#637538857= SYSTEM_AND_FREEDOM(#637538856,ENUMERATED_DEGREE_OF_FREEDOM(.Z_ROTATION.));
#637538861= SYSTEM_AND_FREEDOM(#637538856,ENUMERATED_DEGREE_OF_FREEDOM(.Z_ROTATION.));
#637538858=
DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('202.2.6',$,$,(#637538253,#637538296),#637538282,(#
637538861,#637538857),#637538862);
#637538862= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT($,12.6);
#637538863=
DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('202.1.6',$,$,(#637538253,#637538296),#637538282,(#
637538861,#637538857),#637538866);
#637538866= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT($,11.6);
#637538867=
DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('202.0.6',$,$,(#637538253,#637538296),#637538282,(#
637538861,#637538857),#637538870);
#637538870= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT($,10.6);

#637538871= SYSTEM_AND_FREEDOM(#637538856,ENUMERATED_DEGREE_OF_FREEDOM(.Y_ROTATION.));
#637538875= SYSTEM_AND_FREEDOM(#637538856,ENUMERATED_DEGREE_OF_FREEDOM(.Y_ROTATION.));
#637538872=
DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('202.2.5',$,$,(#637538253,#637538296),#637538282,(#
637538875,#637538871),#637538876);
#637538876= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT($,12.5);
#637538877=
DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('202.1.5',$,$,(#637538253,#637538296),#637538282,(#
637538875,#637538871),#637538880);
#637538880= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT($,11.5);
#637538881=
DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('202.0.5',$,$,(#637538253,#637538296),#637538282,(#
637538875,#637538871),#637538884);
#637538884= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT($,10.5);

#637538885= SYSTEM_AND_FREEDOM(#637538856,ENUMERATED_DEGREE_OF_FREEDOM(.X_ROTATION.));
#637538889= SYSTEM_AND_FREEDOM(#637538856,ENUMERATED_DEGREE_OF_FREEDOM(.X_ROTATION.));
#637538886=
DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('202.2.4',$,$,(#637538253,#637538296),#637538282,(#
637538889,#637538885),#637538890);
#637538890= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT($,12.4);
#637538891=
DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('202.1.4',$,$,(#637538253,#637538296),#637538282,(#
637538889,#637538885),#637538894);
#637538894= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT($,11.4);
#637538895=
DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('202.0.4',$,$,(#637538253,#637538296),#637538282,(#
637538889,#637538885),#637538898);
#637538898= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT($,10.4);

#637538899= SYSTEM_AND_FREEDOM(#637538856,ENUMERATED_DEGREE_OF_FREEDOM(.Z_TRANSLATION.));
#637538903= SYSTEM_AND_FREEDOM(#637538856,ENUMERATED_DEGREE_OF_FREEDOM(.Z_TRANSLATION.));
#637538900=
DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('202.2.3',$,$,(#637538253,#637538296),#637538282,(#
637538903,#637538899),#637538904);
#637538904= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT($,12.3);
#637538905=
DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('202.1.3',$,$,(#637538253,#637538296),#637538282,(#
637538903,#637538899),#637538908);
#637538908= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT($,11.3);
#637538909=
DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('202.0.3',$,$,(#637538253,#637538296),#637538282,(#
637538903,#637538899),#637538912);
#637538912= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT($,10.3);

#637538913= SYSTEM_AND_FREEDOM(#637538856,ENUMERATED_DEGREE_OF_FREEDOM(.Y_TRANSLATION.));
```

```
#637538917= SYSTEM_AND_FREEDOM(#637538856,ENUMERATED_DEGREE_OF_FREEDOM(.Y_TRANSLATION.));
#637538914=
DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('202.2.2',$,$,(#637538253,#637538296),#637538282,(#
637538917,#637538913),#637538918);
#637538918= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT($,11.2);
#637538919=
DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('202.1.2',$,$,(#637538253,#637538296),#637538282,(#
637538917,#637538913),#637538922);
#637538922= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT($,11.2);
#637538923=
DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('202.0.2',$,$,(#637538253,#637538296),#637538282,(#
637538917,#637538913),#637538926);
#637538926= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT($,10.2);

#637538927= SYSTEM_AND_FREEDOM(#637538856,ENUMERATED_DEGREE_OF_FREEDOM(.X_TRANSLATION.));
#637538931= SYSTEM_AND_FREEDOM(#637538856,ENUMERATED_DEGREE_OF_FREEDOM(.X_TRANSLATION.));
#637538928=
DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('202.2.1',$,$,(#637538253,#637538296),#637538282,(#
637538931,#637538927),#637538932);
#637538932= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT($,12.1);
#637538933=
DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('202.1.1',$,$,(#637538253,#637538296),#637538282,(#
637538931,#637538927),#637538936);
#637538936= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT($,11.1);
#637538937=
DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('202.0.1',$,$,(#637538253,#637538296),#637538282,(#
637538931,#637538927),#637538940);
#637538940= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT($,10.1);
```

If standard NASTRAN elastic spring elements are specified connecting the same GRID points, the resulting STEP instances are constructed as shown. The second set of spring elements specify 2 distinct elements on one bulk data card entry. Again, the real and integer values are ficticious to indicate position on the card and differenciate the same field from 2 similar card entries.

```
CELAS1,300,394,1,1,3,1
CELAS1,301,395,1,2,3,2
CELAS1,302,396,1,3,3,3
CELAS1,303,397,1,4,3,4
CELAS1,304,398,1,5,3,5
CELAS1,305,399,1,6,3,6
$
PELAS,394,1.0,1.1,1.2,395,2.0,2.1,2.2
PELAS,396,3.0,3.1,3.2,397,4.0,4.1,4.2
PELAS,398,5.0,5.1,5.2,399,6.0,6.1,6.2
$
CELAS2,400,7.0,1,1,3,1,7.1,7.2
CELAS2,401,8.0,1,1,3,1,8.1,8.2
CELAS2,402,9.0,1,1,3,1,9.1,9.2
CELAS2,403,10.0,1,1,3,1,10.1,10.2
CELAS2,404,11.0,1,1,3,1,11.1,11.2
CELAS2,405,12.0,1,1,3,1,12.1,12.2
$
```

```
#637538284=
FEA_AXIS2_PLACEMENT_3D('0',#637538285,#637538287,#637538289,.CARTESIAN.,'FEA_BASIC_COORD_SYSTEM.0');
#637538969= DIRECTIONALLY_EXPLICIT_ELEMENT_COORDINATE_SYSTEM_ARBITRARY('',#637538284);

#637538971= DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('305.1.6-
6',$,$,(#637538253,#637538296),#637538282,(#637538974,#637538970),#637538975);
#637538976= DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('305.0.6-
6',$,$,(#637538253,#637538296),#637538282,(#637538974,#637538970),#637538979);
#637538970= SYSTEM_AND_FREEDOM(#637538969,ENUMERATED_DEGREE_OF_FREEDOM(.Z_ROTATION.));
#637538974= SYSTEM_AND_FREEDOM(#637538969,ENUMERATED_DEGREE_OF_FREEDOM(.Z_ROTATION.));
#637538975= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT(ENUMERATED_MATRIX_PROPERTY_TYPE(.DAMPING.),6.1);
#637538979= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT(ENUMERATED_MATRIX_PROPERTY_TYPE(.STIFFNESS.),6.);
```

```
#637538986= DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('304.0.5-
5',$,$,(#637538253,#637538296),#637538282,(#637538984,#637538980),#637538989);
#637538981= DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('304.1.5-
5',$,$,(#637538253,#637538296),#637538282,(#637538984,#637538980),#637538985);
#637538980= SYSTEM_AND_FREEDOM(#637538969,ENUMERATED_DEGREE_OF_FREEDOM(.Y_ROTATION.));
#637538984= SYSTEM_AND_FREEDOM(#637538969,ENUMERATED_DEGREE_OF_FREEDOM(.Y_ROTATION.));
#637538985= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT(ENUMERATED_MATRIX_PROPERTY_TYPE(.DAMPING.),5.1);
#637538989= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT(ENUMERATED_MATRIX_PROPERTY_TYPE(.STIFFNESS.),5.);

#637538996= DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('303.0.4-
4',$,$,(#637538253,#637538296),#637538282,(#637538994,#637538990),#637538999);
#637538991= DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('303.1.4-
4',$,$,(#637538253,#637538296),#637538282,(#637538994,#637538990),#637538995);
#637538990= SYSTEM_AND_FREEDOM(#637538969,ENUMERATED_DEGREE_OF_FREEDOM(.X_ROTATION.));
#637538994= SYSTEM_AND_FREEDOM(#637538969,ENUMERATED_DEGREE_OF_FREEDOM(.X_ROTATION.));
#637538995= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT(ENUMERATED_MATRIX_PROPERTY_TYPE(.DAMPING.),4.1);
#637538999= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT(ENUMERATED_MATRIX_PROPERTY_TYPE(.STIFFNESS.),4.);

#637539001= DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('302.1.3-
3',$,$,(#637538253,#637538296),#637538282,(#637539004,#637539000),#637539005);
#637539006= DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('302.0.3-
3',$,$,(#637538253,#637538296),#637538282,(#637539004,#637539000),#637539009);
#637539000= SYSTEM_AND_FREEDOM(#637538969,ENUMERATED_DEGREE_OF_FREEDOM(.Z_TRANSLATION.));
#637539004= SYSTEM_AND_FREEDOM(#637538969,ENUMERATED_DEGREE_OF_FREEDOM(.Z_TRANSLATION.));
#637539005= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT(ENUMERATED_MATRIX_PROPERTY_TYPE(.DAMPING.),3.1);
#637539009= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT(ENUMERATED_MATRIX_PROPERTY_TYPE(.STIFFNESS.),3.);

#637539011= DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('301.1.2-
2',$,$,(#637538253,#637538296),#637538282,(#637539014,#637539010),#637539015);
#637539016= DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('301.0.2-
2',$,$,(#637538253,#637538296),#637538282,(#637539014,#637539010),#637539019);
#637539010= SYSTEM_AND_FREEDOM(#637538969,ENUMERATED_DEGREE_OF_FREEDOM(.Y_TRANSLATION.));
#637539014= SYSTEM_AND_FREEDOM(#637538969,ENUMERATED_DEGREE_OF_FREEDOM(.Y_TRANSLATION.));
#637539015= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT(ENUMERATED_MATRIX_PROPERTY_TYPE(.DAMPING.),2.1);
#637539019= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT(ENUMERATED_MATRIX_PROPERTY_TYPE(.STIFFNESS.),2.);

#637539021= DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('300.1.1-
1',$,$,(#637538253,#637538296),#637538282,(#637539024,#637539020),#637539025);
#637539026= DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('300.0.1-
1',$,$,(#637538253,#637538296),#637538282,(#637539024,#637539020),#637539029);
#637539020= SYSTEM_AND_FREEDOM(#637538969,ENUMERATED_DEGREE_OF_FREEDOM(.X_TRANSLATION.));
#637539024= SYSTEM_AND_FREEDOM(#637538969,ENUMERATED_DEGREE_OF_FREEDOM(.X_TRANSLATION.));
#637539025= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT(ENUMERATED_MATRIX_PROPERTY_TYPE(.DAMPING.),1.1);
#637539029= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT(ENUMERATED_MATRIX_PROPERTY_TYPE(.STIFFNESS.),1.);
```

If the degree of freedoms are changed on the bulk data card like below, the STEP instance data below represents the mapped information.

```
CELAS1,300,394,1,1,3,2
CELAS1,301,395,1,2,3,3
CELAS1,302,396,1,3,3,1
$
```

```
#637539001= DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('302.1.3-
1',$,$,(#637538253,#637538296),#637538282,(#637539004,#637539000),#637539005);
#637539006= DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('302.0.3-
1',$,$,(#637538253,#637538296),#637538282,(#637539004,#637539000),#637539009);
#637539000= SYSTEM_AND_FREEDOM(#637538969,ENUMERATED_DEGREE_OF_FREEDOM(.X_TRANSLATION.));
#637539004= SYSTEM_AND_FREEDOM(#637538969,ENUMERATED_DEGREE_OF_FREEDOM(.Z_TRANSLATION.));
#637539005= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT(ENUMERATED_MATRIX_PROPERTY_TYPE(.DAMPING.),3.1);
#637539009= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT(ENUMERATED_MATRIX_PROPERTY_TYPE(.STIFFNESS.),3.);

#637539011= DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('301.1.2-
3',$,$,(#637538253,#637538296),#637538282,(#637539014,#637539010),#637539015);
#637539016= DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('301.0.2-
3',$,$,(#637538253,#637538296),#637538282,(#637539014,#637539010),#637539019);
#637539010= SYSTEM_AND_FREEDOM(#637538969,ENUMERATED_DEGREE_OF_FREEDOM(.Z_TRANSLATION.));
#637539014= SYSTEM_AND_FREEDOM(#637538969,ENUMERATED_DEGREE_OF_FREEDOM(.Y_TRANSLATION.));
#637539015= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT(ENUMERATED_MATRIX_PROPERTY_TYPE(.DAMPING.),2.1);
#637539019= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT(ENUMERATED_MATRIX_PROPERTY_TYPE(.STIFFNESS.),2.);

#637539021= DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('300.1.1-
2',$,$,(#637538253,#637538296),#637538282,(#637539024,#637539020),#637539025);
```

```
#637539026= DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('300.0.1-
2',$,$,(#637538253,#637538296),#637538282,(#637539024,#637539020),#637539029);
#637539020= SYSTEM_AND_FREEDOM(#637538969,ENUMERATED_DEGREE_OF_FREEDOM(.Y_TRANSLATION.));
#637539024= SYSTEM_AND_FREEDOM(#637538969,ENUMERATED_DEGREE_OF_FREEDOM(.X_TRANSLATION.));
#637539025= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT(ENUMERATED_MATRIX_PROPERTY_TYPE(.DAMPING.),1.1);
#637539029= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT(ENUMERATED_MATRIX_PROPERTY_TYPE(.STIFFNESS.),1.);
```

These are the outputs from the celas2 elements:

```
CELAS2,400,7.0,1,1,3,1,7.1,7.2
CELAS2,401,8.0,1,1,3,1,8.1,8.2
CELAS2,402,9.0,1,1,3,1,9.1,9.2
CELAS2,403,10.0,1,1,3,1,10.1,10.2
CELAS2,404,11.0,1,1,3,1,11.1,11.2
CELAS2,405,12.0,1,1,3,1,12.1,12.2
```

```
#637538971= DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('405.1.1-
1',$,$,(#637538253,#637538296),#637538282,(#637538974,#637538970),#637538975);
#637538976= DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('405.0.1-
1',$,$,(#637538253,#637538296),#637538282,(#637538974,#637538970),#637538979);
#637538970= SYSTEM_AND_FREEDOM(#637538969,ENUMERATED_DEGREE_OF_FREEDOM(.X_TRANSLATION.));
#637538974= SYSTEM_AND_FREEDOM(#637538969,ENUMERATED_DEGREE_OF_FREEDOM(.X_TRANSLATION.));
#637538975= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT(ENUMERATED_MATRIX_PROPERTY_TYPE(.DAMPING.),12.1);
#637538979= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT(ENUMERATED_MATRIX_PROPERTY_TYPE(.STIFFNESS.),12.);

#637538981= DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('404.1.1-
1',$,$,(#637538253,#637538296),#637538282,(#637538984,#637538980),#637538985);
#637538986= DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('404.0.1-
1',$,$,(#637538253,#637538296),#637538282,(#637538984,#637538980),#637538989);
#637538980= SYSTEM_AND_FREEDOM(#637538969,ENUMERATED_DEGREE_OF_FREEDOM(.X_TRANSLATION.));
#637538984= SYSTEM_AND_FREEDOM(#637538969,ENUMERATED_DEGREE_OF_FREEDOM(.X_TRANSLATION.));
#637538985= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT(ENUMERATED_MATRIX_PROPERTY_TYPE(.DAMPING.),11.1);
#637538989= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT(ENUMERATED_MATRIX_PROPERTY_TYPE(.STIFFNESS.),11.);

#637538991= DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('403.1.1-
1',$,$,(#637538253,#637538296),#637538282,(#637538994,#637538990),#637538995);
#637538996= DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('403.0.1-
1',$,$,(#637538253,#637538296),#637538282,(#637538994,#637538990),#637538999);
#637538990= SYSTEM_AND_FREEDOM(#637538969,ENUMERATED_DEGREE_OF_FREEDOM(.X_TRANSLATION.));
#637538994= SYSTEM_AND_FREEDOM(#637538969,ENUMERATED_DEGREE_OF_FREEDOM(.X_TRANSLATION.));
#637538995= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT(ENUMERATED_MATRIX_PROPERTY_TYPE(.DAMPING.),10.1);
#637538999= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT(ENUMERATED_MATRIX_PROPERTY_TYPE(.STIFFNESS.),10.);

#637539001= DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('402.1.1-
1',$,$,(#637538253,#637538296),#637538282,(#637539004,#637539000),#637539005);
#637539006= DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('402.0.1-
1',$,$,(#637538253,#637538296),#637538282,(#637539004,#637539000),#637539009);
#637539000= SYSTEM_AND_FREEDOM(#637538969,ENUMERATED_DEGREE_OF_FREEDOM(.X_TRANSLATION.));
#637539004= SYSTEM_AND_FREEDOM(#637538969,ENUMERATED_DEGREE_OF_FREEDOM(.X_TRANSLATION.));
#637539005= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT(ENUMERATED_MATRIX_PROPERTY_TYPE(.DAMPING.),9.1);
#637539009= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT(ENUMERATED_MATRIX_PROPERTY_TYPE(.STIFFNESS.),9.);

#637539011= DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('401.1.1-
1',$,$,(#637538253,#637538296),#637538282,(#637539014,#637539010),#637539015);
#637539016= DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('401.0.1-
1',$,$,(#637538253,#637538296),#637538282,(#637539014,#637539010),#637539019);
#637539010= SYSTEM_AND_FREEDOM(#637538969,ENUMERATED_DEGREE_OF_FREEDOM(.X_TRANSLATION.));
#637539014= SYSTEM_AND_FREEDOM(#637538969,ENUMERATED_DEGREE_OF_FREEDOM(.X_TRANSLATION.));
#637539015= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT(ENUMERATED_MATRIX_PROPERTY_TYPE(.DAMPING.),8.1);
#637539019= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT(ENUMERATED_MATRIX_PROPERTY_TYPE(.STIFFNESS.),8.);

#637539021= DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('400.1.1-
1',$,$,(#637538253,#637538296),#637538282,(#637539024,#637539020),#637539025);
#637539026= DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION('400.0.1-
1',$,$,(#637538253,#637538296),#637538282,(#637539024,#637539020),#637539029);
#637539020= SYSTEM_AND_FREEDOM(#637538969,ENUMERATED_DEGREE_OF_FREEDOM(.X_TRANSLATION.));
#637539024= SYSTEM_AND_FREEDOM(#637538969,ENUMERATED_DEGREE_OF_FREEDOM(.X_TRANSLATION.));
#637539025= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT(ENUMERATED_MATRIX_PROPERTY_TYPE(.DAMPING.),7.1);
#637539029= DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT(ENUMERATED_MATRIX_PROPERTY_TYPE(.STIFFNESS.),7.);
```

Appendix E Entity/Type to STEP Parts

This appendix lists many of the commonly used the entities and types used in AP 209 ed2 and cross references them to the STEP part in which they are defined. The first column is the entity or type name. The second column defines whether it is an entity or type. The last column lists the STEP part in which the full textual definition of the item exists. A complete list can be found in ISO 10303-209:2014.

NAME	KIND	PART
ACTION	ENTITY	41
ACTION_ASSIGNMENT	ENTITY	41
ACTION_DIRECTIVE	ENTITY	41
ACTION_ITEM	TYPE	209
ACTION_METHOD	ENTITY	41
ACTION_REQUEST_ASSIGNMENT	ENTITY	41
ACTION_REQUEST_ITEM	TYPE	209
ACTION_REQUEST_SOLUTION	ENTITY	41
ACTION_REQUEST_STATUS	ENTITY	41
ACTION_STATUS	ENTITY	41
ACTION_TYPE	TYPE	104
ADDRESS	ENTITY	41
ADVANCED_BREP_SHAPE_REPRESENTATION	ENTITY	514
ADVANCED_FACE	ENTITY	511
AGGREGATED_ANGULAR_VARIABLE	TYPE	104
AGGREGATED_SCALAR_VARIABLE	TYPE	104
AGGREGATED_TENSOR2_3D_VARIABLE	TYPE	104
AGGREGATED_VECTOR_3D_VARIABLE	TYPE	104
AHEAD_OR_BEHIND	TYPE	41
ALIGNED_AXIS_TOLERANCE	ENTITY	104
ALIGNED_CURVE_3D_ELEMENT_COORDINATE_SYSTEM	ENTITY	104
ALIGNED_SURFACE_2D_ELEMENT_COORDINATE_SYSTEM	ENTITY	104
ALIGNED_SURFACE_3D_ELEMENT_COORDINATE_SYSTEM	ENTITY	104
ALTERNATE_PRODUCT_RELATIONSHIP	ENTITY	41
AMOUNT_OF_SUBSTANCE_MEASURE	TYPE	41
AMOUNT_OF_SUBSTANCE_MEASURE_WITH_UNIT	ENTITY	41
AMOUNT_OF_SUBSTANCE_UNIT	ENTITY	41
ANALYSIS_ITEM_WITHIN_REPRESENTATION	ENTITY	104
ANALYSIS_MESSAGE	ENTITY	104
ANALYSIS_STEP	ENTITY	104
ANGULAR_VALUE	TYPE	104
ANISOTROPIC_SYMMETRIC_TENSOR2_2D	TYPE	104
ANISOTROPIC_SYMMETRIC_TENSOR2_3D	TYPE	104
ANISOTROPIC_SYMMETRIC_TENSOR4_2D	TYPE	104
ANISOTROPIC_SYMMETRIC_TENSOR4_3D	TYPE	104
APPLICATION_CONTEXT	ENTITY	41
APPLICATION_CONTEXT_ELEMENT	ENTITY	41
APPLICATION_DEFINED_DEGREE_OF_FREEDOM	TYPE	104
APPLICATION_DEFINED_ELEMENT_PURPOSE	TYPE	104
APPLICATION_DEFINED_MATRIX_PROPERTY_TYPE	TYPE	104
APPLICATION_DEFINED_SCALAR_VARIABLE	TYPE	104
APPLICATION_DEFINED_TENSOR2_2D_VARIABLE	TYPE	104

NAME	KIND	PART
APPLICATION_DEFINED_TENSOR2_3D_VARIABLE	TYPE	104
APPLICATION_DEFINED_VECTOR_2D_VARIABLE	TYPE	104
APPLICATION_DEFINED_VECTOR_3D_VARIABLE	TYPE	104
APPLICATION_PROTOCOL_DEFINITION	ENTITY	41
APPLIED_ACTION_ASSIGNMENT	ENTITY	209
APPLIED_ACTION_REQUEST_ASSIGNMENT	ENTITY	209
APPLIED_APPROVAL_ASSIGNMENT	ENTITY	209
APPLIED_CERTIFICATION_ASSIGNMENT	ENTITY	209
APPLIED_CONTRACT_ASSIGNMENT	ENTITY	209
APPLIED_DATE_AND_TIME_ASSIGNMENT	ENTITY	209
APPLIED_DATE_ASSIGNMENT	ENTITY	209
APPLIED_DOCUMENT_REFERENCE	ENTITY	209
APPLIED_ORGANIZATION_ASSIGNMENT	ENTITY	209
APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT	ENTITY	209
APPLIED_SECURITY_CLASSIFICATION_ASSIGNMENT	ENTITY	209
APPROVAL	ENTITY	41
APPROVAL_ASSIGNMENT	ENTITY	41
APPROVAL_DATE_TIME	ENTITY	41
APPROVAL_ITEM	TYPE	209
APPROVAL_PERSON_ORGANIZATION	ENTITY	41
APPROVAL_RELATIONSHIP	ENTITY	41
APPROVAL_ROLE	ENTITY	41
APPROVAL_STATUS	ENTITY	41
ARBITRARY_VOLUME_2D_ELEMENT_COORDINATE_SYSTEM	ENTITY	104
ARBITRARY_VOLUME_3D_ELEMENT_COORDINATE_SYSTEM	ENTITY	104
AREA_MEASURE	TYPE	41
AREA_MEASURE_WITH_UNIT	ENTITY	41
AREA_UNIT	ENTITY	41
ASSEMBLY_COMPONENT_USAGE	ENTITY	41
ASSEMBLY_COMPONENT_USAGE_SUBSTITUTE	ENTITY	41
AXI_OR_PLANE	TYPE	104
AXIS1_PLACEMENT	ENTITY	42
AXIS2_PLACEMENT	TYPE	42
AXIS2_PLACEMENT_2D	ENTITY	42
AXIS2_PLACEMENT_3D	ENTITY	42
AXISYMMETRIC_2D_ELEMENT_PROPERTY	ENTITY	104
AXISYMMETRIC_CURVE_2D_ELEMENT_DESCRIPTOR	ENTITY	104
AXISYMMETRIC_CURVE_2D_ELEMENT_REPRESENTATION	ENTITY	104
AXISYMMETRIC_SURFACE_2D_ELEMENT_DESCRIPTOR	ENTITY	104
AXISYMMETRIC_SURFACE_2D_ELEMENT_REPRESENTATION	ENTITY	104
AXISYMMETRIC_VOLUME_2D_ELEMENT_DESCRIPTOR	ENTITY	104
AXISYMMETRIC_VOLUME_2D_ELEMENT_REPRESENTATION	ENTITY	104
B_SPLINE_CURVE	ENTITY	42
B_SPLINE_CURVE_FORM	TYPE	42
B_SPLINE_CURVE_WITH_KNOTS	ENTITY	42
B_SPLINE_SURFACE	ENTITY	42
B_SPLINE_SURFACE_FORM	TYPE	42
B_SPLINE_SURFACE_WITH_KNOTS	ENTITY	42
BEVELED_SHEET_REPRESENTATION	ENTITY	209
BEZIER_CURVE	ENTITY	42

NAME	KIND	PART
BEZIER_SURFACE	ENTITY	42
BOOLEAN_OPERAND	TYPE	42
BOUNDARY_AGGREGATED_VARIABLE	TYPE	104
BOUNDARY_CURVE	ENTITY	42
BOUNDARY_CURVE_SCALAR_VARIABLE	TYPE	104
BOUNDARY_CURVE_VECTOR_3D_VARIABLE	TYPE	104
BOUNDARY_EDGE_VARIABLE	TYPE	104
BOUNDARY_SURFACE_SCALAR_VARIABLE	TYPE	104
BOUNDARY_SURFACE_VECTOR_3D_VARIABLE	TYPE	104
BOUNDARY_VARIABLE	TYPE	104
BOUNDED_CURVE	ENTITY	42
BOUNDED_PCURVE	ENTITY	42
BOUNDED_SURFACE	ENTITY	42
BOUNDED_SURFACE_CURVE	ENTITY	42
BREP_WITH_VOIDS	ENTITY	42
CALCULATED_STATE	ENTITY	104
CALENDAR_DATE	ENTITY	41
CARTESIAN_POINT	ENTITY	42
CARTESIAN_TRANSFORMATION_OPERATOR	ENTITY	42
CARTESIAN_TRANSFORMATION_OPERATOR_2D	ENTITY	42
CARTESIAN_TRANSFORMATION_OPERATOR_3D	ENTITY	42
CERTIFICATION	ENTITY	41
CERTIFICATION_ASSIGNMENT	ENTITY	41
CERTIFICATION_ITEM	TYPE	209
CERTIFICATION_TYPE	ENTITY	41
CHARACTERIZED_DEFINITION	TYPE	41
CHARACTERIZED_MATERIAL_PROPERTY	TYPE	45
CHARACTERIZED_OBJECT	ENTITY	41
CHARACTERIZED_PRODUCT_DEFINITION	TYPE	41
CIRCLE	ENTITY	42
CLOSED_SHELL	ENTITY	42
COMPOSITE_ASSEMBLY_DEFINITION	ENTITY	209
COMPOSITE_ASSEMBLY_SEQUENCE_DEFINITION	ENTITY	209
COMPOSITE_CURVE	ENTITY	42
COMPOSITE_CURVE_ON_SURFACE	ENTITY	42
COMPOSITE_CURVE_SEGMENT	ENTITY	42
COMPOSITE_SHEET_REPRESENTATION	ENTITY	209
CONFIGURATION_DESIGN	ENTITY	44
CONFIGURATION_EFFECTIVITY	ENTITY	44
CONFIGURATION_ITEM	ENTITY	44
CONIC	ENTITY	42
CONICAL_SURFACE	ENTITY	42
CONNECTED_EDGE_SET	ENTITY	42
CONNECTED_FACE_SET	ENTITY	42
CONSTANT_SURFACE_3D_ELEMENT_COORDINATE_SYSTEM	ENTITY	104
CONSTRAINT_ELEMENT	ENTITY	104
CONTEXT_DEPENDENT_MEASURE	TYPE	41
CONTEXT_DEPENDENT_SHAPE_REPRESENTATION	ENTITY	41
CONTEXT_DEPENDENT_UNIT	ENTITY	41
CONTRACT	ENTITY	41

NAME	**KIND**	**PART**
CONTRACT_ASSIGNMENT	ENTITY	41
CONTRACT_ITEM	TYPE	209
CONTRACT_TYPE	ENTITY	41
CONTROL	ENTITY	104
CONTROL_ANALYSIS_STEP	ENTITY	104
CONTROL_LINEAR_MODES_AND_FREQUENCIES_ANALYSIS_STEP	ENTITY	104
CONTROL_LINEAR_MODES_AND_FREQUENCIES_PROCESS	ENTITY	104
CONTROL_LINEAR_STATIC_ANALYSIS_STEP	ENTITY	104
CONTROL_LINEAR_STATIC_ANALYSIS_STEP_WITH_HARMONIC	ENTITY	104
CONTROL_LINEAR_STATIC_LOAD_INCREMENT_PROCESS	ENTITY	104
CONTROL_PROCESS	ENTITY	104
CONTROL_RESULT_RELATIONSHIP	ENTITY	104
CONVERSION_BASED_UNIT	ENTITY	41
COORDINATE_SYSTEM_TYPE	TYPE	104
COORDINATED_UNIVERSAL_TIME_OFFSET	ENTITY	41
COUNT_MEASURE	TYPE	41
CURVE	ENTITY	42
CURVE_2D_ELEMENT_BASIS	ENTITY	104
CURVE_2D_ELEMENT_CONSTANT_SPECIFIED_VARIABLE_VALUE	ENTITY	104
CURVE_2D_ELEMENT_CONSTANT_SPECIFIED_VOLUME_VARIABLE_ VALUE	ENTITY	104
CURVE_2D_ELEMENT_COORDINATE_SYSTEM	ENTITY	104
CURVE_2D_ELEMENT_DESCRIPTOR	TYPE	104
CURVE_2D_ELEMENT_FIELD_VARIABLE_DEFINITION	ENTITY	104
CURVE_2D_ELEMENT_GROUP	ENTITY	104
CURVE_2D_ELEMENT_INTEGRATED_MATRIX	ENTITY	104
CURVE_2D_ELEMENT_INTEGRATED_MATRIX_WITH_DEFINITION	ENTITY	104
CURVE_2D_ELEMENT_INTEGRATION	ENTITY	104
CURVE_2D_ELEMENT_LOCATION_POINT_VARIABLE_VALUES	ENTITY	104
CURVE_2D_ELEMENT_LOCATION_POINT_VOLUME_VARIABLE_VALUES	ENTITY	104
CURVE_2D_ELEMENT_OUTPUT_REFERENCE	TYPE	104
CURVE_2D_ELEMENT_PROPERTY	ENTITY	104
CURVE_2D_ELEMENT_REPRESENTATION	TYPE	104
CURVE_2D_ELEMENT_VALUE_AND_LOCATION	ENTITY	104
CURVE_2D_ELEMENT_VALUE_AND_VOLUME_LOCATION	ENTITY	104
CURVE_2D_NODE_FIELD_AGGREGATED_VARIABLE_VALUES	ENTITY	104
CURVE_2D_NODE_FIELD_SECTION_VARIABLE_VALUES	ENTITY	104
CURVE_2D_NODE_FIELD_VARIABLE_DEFINITION	ENTITY	104
CURVE_2D_STATE_COORDINATE_SYSTEM	TYPE	104
CURVE_2D_SUBSTRUCTURE_ELEMENT_REFERENCE	ENTITY	104
CURVE_2D_WHOLE_ELEMENT_VARIABLE_VALUE	ENTITY	104
CURVE_3D_ELEMENT_BASIS	ENTITY	104
CURVE_3D_ELEMENT_CONSTANT_SPECIFIED_VARIABLE_VALUE	ENTITY	104
CURVE_3D_ELEMENT_CONSTANT_SPECIFIED_VOLUME_VARIABLE_ VALUE	ENTITY	104
CURVE_3D_ELEMENT_COORDINATE_SYSTEM	TYPE	104
CURVE_3D_ELEMENT_DESCRIPTOR	ENTITY	104
CURVE_3D_ELEMENT_FIELD_VARIABLE_DEFINITION	ENTITY	104
CURVE_3D_ELEMENT_GROUP	ENTITY	104
CURVE_3D_ELEMENT_INTEGRATED_MATRIX	ENTITY	104

NAME	**KIND**	**PART**
CURVE_3D_ELEMENT_INTEGRATED_MATRIX_WITH_DEFINITION	ENTITY	104
CURVE_3D_ELEMENT_INTEGRATION	ENTITY	104
CURVE_3D_ELEMENT_LENGTH_INTEGRATION	TYPE	104
CURVE_3D_ELEMENT_LENGTH_INTEGRATION_EXPLICIT	ENTITY	104
CURVE_3D_ELEMENT_LENGTH_INTEGRATION_RULE	ENTITY	104
CURVE_3D_ELEMENT_LOCATION_POINT_VARIABLE_VALUES	ENTITY	104
CURVE_3D_ELEMENT_LOCATION_POINT_VOLUME_VARIABLE_VALUES	ENTITY	104
CURVE_3D_ELEMENT_NODAL_SPECIFIED_VARIABLE_VALUES	ENTITY	104
CURVE_3D_ELEMENT_OUTPUT_REFERENCE	TYPE	104
CURVE_3D_ELEMENT_POSITION_WEIGHT	ENTITY	104
CURVE_3D_ELEMENT_PROPERTY	ENTITY	104
CURVE_3D_ELEMENT_REPRESENTATION	ENTITY	104
CURVE_3D_ELEMENT_VALUE_AND_LOCATION	ENTITY	104
CURVE_3D_ELEMENT_VALUE_AND_VOLUME_LOCATION	ENTITY	104
CURVE_3D_NODE_FIELD_AGGREGATED_VARIABLE_VALUES	ENTITY	104
CURVE_3D_NODE_FIELD_SECTION_VARIABLE_VALUES	ENTITY	104
CURVE_3D_NODE_FIELD_VARIABLE_DEFINITION	ENTITY	104
CURVE_3D_STATE_COORDINATE_SYSTEM	TYPE	104
CURVE_3D_SUBSTRUCTURE_ELEMENT_REFERENCE	ENTITY	104
CURVE_3D_WHOLE_ELEMENT_VARIABLE_VALUE	ENTITY	104
CURVE_BOUNDED_SURFACE	ENTITY	42
CURVE_CONSTRAINT	ENTITY	104
CURVE_EDGE	TYPE	104
CURVE_ELEMENT_END_COORDINATE_SYSTEM	TYPE	104
CURVE_ELEMENT_END_OFFSET	ENTITY	104
CURVE_ELEMENT_END_RELEASE	ENTITY	104
CURVE_ELEMENT_END_RELEASE_PACKET	ENTITY	104
CURVE_ELEMENT_FREEDOM	TYPE	104
CURVE_ELEMENT_INTERVAL	ENTITY	104
CURVE_ELEMENT_INTERVAL_CONSTANT	ENTITY	104
CURVE_ELEMENT_INTERVAL_LINEARLY_VARYING	ENTITY	104
CURVE_ELEMENT_LOCATION	ENTITY	104
CURVE_ELEMENT_PURPOSE	TYPE	104
CURVE_ELEMENT_SECTION_DEFINITION	ENTITY	104
CURVE_ELEMENT_SECTION_DERIVED_DEFINITIONS	ENTITY	104
CURVE_ELEMENT_VARIABLE	TYPE	104
CURVE_FREEDOM_ACTION_DEFINITION	ENTITY	104
CURVE_FREEDOM_AND_VALUE_DEFINITION	ENTITY	104
CURVE_FREEDOM_VALUES	ENTITY	104
CURVE_MATRIX_PROPERTY_TYPE	TYPE	104
CURVE_ON_SURFACE	TYPE	42
CURVE_REPLICA	ENTITY	42
CURVE_SCALAR_VARIABLE	TYPE	104
CURVE_SECTION_ELEMENT_LOCATION	ENTITY	104
CURVE_SECTION_INTEGRATION_EXPLICIT	ENTITY	104
CURVE_VECTOR_2D_VARIABLE	TYPE	104
CURVE_VECTOR_3D_VARIABLE	TYPE	104
CURVE_VOLUME_ELEMENT_LOCATION	ENTITY	104
CYLINDRICAL_HARMONIC_NUMBER	TYPE	104
CYLINDRICAL_POINT	ENTITY	104

NAME	KIND	PART
CYLINDRICAL_SURFACE	ENTITY	42
CYLINDRICAL_SYMMETRY_CONTROL	ENTITY	104
DATA_ENVIRONMENT	ENTITY	45
DATA_ENVIRONMENT_RELATIONSHIP	ENTITY	45
DATE	ENTITY	41
DATE_AND_TIME	ENTITY	41
DATE_AND_TIME_ASSIGNMENT	ENTITY	41
DATE_ASSIGNMENT	ENTITY	41
DATE_ITEM	TYPE	209
DATE_ROLE	ENTITY	41
DATE_TIME_ITEM	TYPE	209
DATE_TIME_ROLE	ENTITY	41
DATE_TIME_SELECT	TYPE	41
DATED_EFFECTIVITY	ENTITY	41
DAY_IN_MONTH_NUMBER	TYPE	41
DAY_IN_WEEK_NUMBER	TYPE	41
DAY_IN_YEAR_NUMBER	TYPE	41
DEFINITIONAL_REPRESENTATION	ENTITY	43
DEGENERATE_PCURVE	ENTITY	42
DEGENERATE_TOROIDAL_SURFACE	ENTITY	42
DEGREE_OF_FREEDOM	TYPE	104
DERIVED_UNIT	ENTITY	41
DERIVED_UNIT_ELEMENT	ENTITY	41
DESCRIPTIVE_MEASURE	TYPE	41
DESCRIPTIVE_REPRESENTATION_ITEM	ENTITY	45
DESIGN_MAKE_FROM_RELATIONSHIP	ENTITY	209
DIMENSION_COUNT	TYPE	42
DIMENSIONAL_CHARACTERISTIC	TYPE	47
DIMENSIONAL_EXPONENTS	ENTITY	41
DIMENSIONAL_SIZE	ENTITY	47
DIRECTED_ACTION	ENTITY	41
DIRECTION	ENTITY	42
DIRECTION_NODE	ENTITY	104
DIRECTIONALLY_EXPLICIT_ELEMENT_COEFFICIENT	ENTITY	104
DIRECTIONALLY_EXPLICIT_ELEMENT_COORDINATE_SYSTEM	TYPE	104
DIRECTIONALLY_EXPLICIT_ELEMENT_COORDINATE_SYSTEM_ALIGNED	ENTITY	104
DIRECTIONALLY_EXPLICIT_ELEMENT_COORDINATE_SYSTEM_ ARBITRARY	ENTITY	104
DIRECTIONALLY_EXPLICIT_ELEMENT_REPRESENTATION	ENTITY	104
DOCUMENT	ENTITY	41
DOCUMENT_REFERENCE	ENTITY	41
DOCUMENT_REFERENCE_ITEM	TYPE	209
DOCUMENT_RELATIONSHIP	ENTITY	41
DOCUMENT_TYPE	ENTITY	41
DOCUMENT_USAGE_CONSTRAINT	ENTITY	41
DOCUMENT_WITH_CLASS	ENTITY	41
DRAPED_DEFINED_TRANSFORMATION	ENTITY	209
DUMMY_NODE	ENTITY	104
EDGE	ENTITY	42
EDGE_BASED_WIREFRAME_MODEL	ENTITY	42

NAME	KIND	PART
EDGE_BASED_WIREFRAME_SHAPE_REPRESENTATION	ENTITY	501
EDGE_CURVE	ENTITY	42
EDGE_LOOP	ENTITY	42
EFFECTIVITY	ENTITY	41
ELECTRIC_CURRENT_MEASURE	TYPE	41
ELEMENT_2D_SHAPE	TYPE	104
ELEMENT_ANALYSIS_MESSAGE	ENTITY	104
ELEMENT_ASPECT	TYPE	104
ELEMENT_DEFINITION	ENTITY	104
ELEMENT_DESCRIPTOR	ENTITY	104
ELEMENT_GEOMETRIC_RELATIONSHIP	ENTITY	104
ELEMENT_GROUP	ENTITY	104
ELEMENT_GROUP_ANALYSIS_MESSAGE	ENTITY	104
ELEMENT_INTEGRATION_ALGEBRAIC	TYPE	104
ELEMENT_MATERIAL	ENTITY	104
ELEMENT_NODAL_FREEDOM_ACTIONS	ENTITY	104
ELEMENT_NODAL_FREEDOM_TERMS	ENTITY	104
ELEMENT_ORDER	TYPE	104
ELEMENT_REPRESENTATION	ENTITY	104
ELEMENT_SEQUENCE	ENTITY	104
ELEMENT_VOLUME	TYPE	104
ELEMENTARY_SURFACE	ENTITY	42
ELLIPSE	ENTITY	42
ENUMERATED_CURVE_ELEMENT_FREEDOM	TYPE	104
ENUMERATED_CURVE_ELEMENT_PURPOSE	TYPE	104
ENUMERATED_CURVE_MATRIX_PROPERTY_TYPE	TYPE	104
ENUMERATED_DEGREE_OF_FREEDOM	TYPE	104
ENUMERATED_MATRIX_PROPERTY_TYPE	TYPE	104
ENUMERATED_PLANE_2D_ELEMENT_PURPOSE	TYPE	104
ENUMERATED_SURFACE_ELEMENT_PURPOSE	TYPE	104
ENUMERATED_SURFACE_MATRIX_PROPERTY_TYPE	TYPE	104
ENUMERATED_VOLUME_ELEMENT_PURPOSE	TYPE	104
EULER_ANGLES	ENTITY	104
EVALUATED_DEGENERATE_PCURVE	ENTITY	42
EXECUTED_ACTION	ENTITY	41
EXPANDED_UNCERTAINTY	ENTITY	45
EXPLICIT_ELEMENT_MATRIX	ENTITY	104
EXPLICIT_ELEMENT_REPRESENTATION	ENTITY	104
FACE	ENTITY	42
FACE_BOUND	ENTITY	42
FACE_OUTER_BOUND	ENTITY	42
FACE_SURFACE	ENTITY	42
FACETED_BREP	ENTITY	42
FACETED_BREP_SHAPE_REPRESENTATION	ENTITY	512
FEA_AREA_DENSITY	ENTITY	104
FEA_AXIS2_PLACEMENT_2D	ENTITY	104
FEA_AXIS2_PLACEMENT_3D	ENTITY	104
FEA_COLUMN_NORMALISED_MONOCLINIC_SYMMETRIC_TENSOR4_3D	TYPE	104
FEA_COLUMN_NORMALISED_ORTHOTROPIC_SYMMETRIC_TENSOR4_3D	TYPE	104
FEA_GROUP	ENTITY	104

NAME	KIND	PART
FEA_GROUP_RELATION	ENTITY	104
FEA_ISO_ORTHOTROPIC_SYMMETRIC_TENSOR4_3D	TYPE	104
FEA_ISOTROPIC_SYMMETRIC_TENSOR4_3D	TYPE	104
FEA_LINEAR_ELASTICITY	ENTITY	104
FEA_MASS_DENSITY	ENTITY	104
FEA_MATERIAL_PROPERTY_GEOMETRIC_RELATIONSHIP	ENTITY	104
FEA_MATERIAL_PROPERTY_REPRESENTATION	ENTITY	104
FEA_MATERIAL_PROPERTY_REPRESENTATION_ITEM	ENTITY	104
FEA_MODEL	ENTITY	104
FEA_MODEL_2D	ENTITY	104
FEA_MODEL_3D	ENTITY	104
FEA_MODEL_DEFINITION	ENTITY	104
FEA_MOISTURE_ABSORPTION	ENTITY	104
FEA_PARAMETRIC_POINT	ENTITY	104
FEA_REPRESENTATION_ITEM	ENTITY	104
FEA_SECANT_COEFFICIENT_OF_LINEAR_THERMAL_EXPANSION	ENTITY	104
FEA_SHELL_BENDING_STIFFNESS	ENTITY	104
FEA_SHELL_MEMBRANE_BENDING_COUPLING_STIFFNESS	ENTITY	104
FEA_SHELL_MEMBRANE_STIFFNESS	ENTITY	104
FEA_SHELL_SHEAR_STIFFNESS	ENTITY	104
FEA_TANGENTIAL_COEFFICIENT_OF_LINEAR_THERMAL_EXPANSION	ENTITY	104
FEA_TRANSVERSE_ISOTROPIC_SYMMETRIC_TENSOR4_3D	TYPE	104
FIELD_VALUE	TYPE	104
FIELD_VARIABLE_DEFINITION	ENTITY	104
FIELD_VARIABLE_ELEMENT_DEFINITION	ENTITY	104
FIELD_VARIABLE_ELEMENT_GROUP_VALUE	ENTITY	104
FIELD_VARIABLE_NODE_DEFINITION	ENTITY	104
FIELD_VARIABLE_WHOLE_MODEL_VALUE	ENTITY	104
FLAT_PATTERN_PLY_REPRESENTATION_RELATIONSHIP	ENTITY	209
FREEDOM_AND_COEFFICIENT	ENTITY	104
FREEDOMS_LIST	ENTITY	104
FUNCTIONALLY_DEFINED_TRANSFORMATION	ENTITY	43
GEOMETRIC_CURVE_SET	ENTITY	42
GEOMETRIC_NODE	ENTITY	104
GEOMETRIC_REPRESENTATION_CONTEXT	ENTITY	42
GEOMETRIC_REPRESENTATION_ITEM	ENTITY	42
GEOMETRIC_SET	ENTITY	42
GEOMETRIC_SET_SELECT	TYPE	42
GEOMETRICALLY_BOUNDED_SURFACE_SHAPE_REPRESENTATION	ENTITY	507
GEOMETRICALLY_BOUNDED_WIREFRAME_SHAPE_REPRESENTATION	ENTITY	510
GLOBAL_UNCERTAINTY_ASSIGNED_CONTEXT	ENTITY	43
GLOBAL_UNIT_ASSIGNED_CONTEXT	ENTITY	41
GROUNDED_DAMPER	ENTITY	104
GROUNDED_SPRING	ENTITY	104
GROUP	ENTITY	41
GROUP_RELATIONSHIP	ENTITY	41
HOUR_IN_DAY	TYPE	41
HYPERBOLA	ENTITY	42
IDENTIFIER	TYPE	41
INTEGRATION_RULE	TYPE	104

NAME	KIND	PART
INTERSECTION_CURVE	ENTITY	42
ISOTROPIC_SYMMETRIC_TENSOR2_3D	TYPE	104
ITEM_DEFINED_TRANSFORMATION	ENTITY	43
KNOT_TYPE	TYPE	42
LABEL	TYPE	41
LAID_DEFINED_TRANSFORMATION	ENTITY	209
LENGTH_MEASURE	TYPE	41
LENGTH_MEASURE_WITH_UNIT	ENTITY	41
LENGTH_UNIT	ENTITY	41
LINE	ENTITY	42
LINEAR_CONSTRAINT_EQUATION_ELEMENT	ENTITY	104
LINEAR_CONSTRAINT_EQUATION_ELEMENT_VALUE	ENTITY	104
LINEAR_CONSTRAINT_EQUATION_NODAL_TERM	ENTITY	104
LINEARLY_SUPERIMPOSED_STATE	ENTITY	104
LIST_OF_REVERSIBLE_TOPOLOGY_ITEM	TYPE	42
LOCAL_TIME	ENTITY	41
LOOP	ENTITY	42
LOT_EFFECTIVITY	ENTITY	41
LUMINOUS_INTENSITY_MEASURE	TYPE	41
MAKE_FROM_USAGE_OPTION	ENTITY	44
MANIFOLD_SOLID_BREP	ENTITY	42
MANIFOLD_SURFACE_SHAPE_REPRESENTATION	ENTITY	509
MAPPED_ITEM	ENTITY	43
MASS_MEASURE	TYPE	41
MASS_MEASURE_WITH_UNIT	ENTITY	41
MASS_UNIT	ENTITY	41
MATERIAL_DESIGNATION	ENTITY	45
MATERIAL_DESIGNATION_CHARACTERIZATION	ENTITY	45
MATERIAL_PROPERTY	ENTITY	45
MATERIAL_PROPERTY_REPRESENTATION	ENTITY	45
MATRIX_PROPERTY_TYPE	TYPE	104
MATRIX_SYMMETRY	TYPE	104
MEASURE_OR_UNSPECIFIED_VALUE	TYPE	104
MEASURE_QUALIFICATION	ENTITY	45
MEASURE_REPRESENTATION_ITEM	ENTITY	45
MEASURE_VALUE	TYPE	41
MEASURE_WITH_UNIT	ENTITY	41
MESSAGE_LEVEL	TYPE	104
MINUTE_IN_HOUR	TYPE	41
MODEL_OR_CONTROL_ELEMENT	TYPE	104
MONTH_IN_YEAR_NUMBER	TYPE	41
NAMED_UNIT	ENTITY	41
NEXT_ASSEMBLY_USAGE_OCCURRENCE	ENTITY	44
NO_SYMMETRY_CONTROL	ENTITY	104
NODAL_DOF_REDUCTION	ENTITY	104
NODAL_FREEDOM_ACTION_DEFINITION	ENTITY	104
NODAL_FREEDOM_AND_VALUE_DEFINITION	ENTITY	104
NODAL_FREEDOM_VALUES	ENTITY	104
NODE	ENTITY	104
NODE_ANALYSIS_MESSAGE	ENTITY	104

NAME	**KIND**	**PART**
NODE_DEFINITION	ENTITY	104
NODE_GEOMETRIC_RELATIONSHIP	ENTITY	104
NODE_GROUP	ENTITY	104
NODE_OUTPUT_REFERENCE	TYPE	104
NODE_REPRESENTATION	ENTITY	104
NODE_SEQUENCE	ENTITY	104
NODE_SET	ENTITY	104
NODE_WITH_SOLUTION_COORDINATE_SYSTEM	ENTITY	104
NODE_WITH_VECTOR	ENTITY	104
NUMERIC_MEASURE	TYPE	41
OFFSET_CURVE_2D	ENTITY	42
OFFSET_CURVE_3D	ENTITY	42
OFFSET_SURFACE	ENTITY	42
OPEN_SHELL	ENTITY	42
ORDINAL_DATE	ENTITY	41
ORGANIZATION	ENTITY	41
ORGANIZATION_ASSIGNMENT	ENTITY	41
ORGANIZATION_ITEM	TYPE	209
ORGANIZATION_RELATIONSHIP	ENTITY	41
ORGANIZATION_ROLE	ENTITY	41
ORGANIZATIONAL_ADDRESS	ENTITY	41
ORGANIZATIONAL_PROJECT	ENTITY	41
ORIENTED_CLOSED_SHELL	ENTITY	42
ORIENTED_EDGE	ENTITY	42
ORIENTED_FACE	ENTITY	42
ORIENTED_OPEN_SHELL	ENTITY	42
ORIENTED_PATH	ENTITY	42
ORTHOTROPIC_SYMMETRIC_TENSOR2_3D	TYPE	104
OUTER_BOUNDARY_CURVE	ENTITY	42
OUTPUT_REQUEST_STATE	ENTITY	104
PARABOLA	ENTITY	42
PARAMETER_VALUE	TYPE	41
PARAMETRIC_CURVE_3D_ELEMENT_COORDINATE_DIRECTION	ENTITY	104
PARAMETRIC_CURVE_3D_ELEMENT_COORDINATE_SYSTEM	ENTITY	104
PARAMETRIC_REPRESENTATION_CONTEXT	ENTITY	43
PARAMETRIC_SURFACE_2D_ELEMENT_COORDINATE_SYSTEM	ENTITY	104
PARAMETRIC_SURFACE_3D_ELEMENT_COORDINATE_SYSTEM	ENTITY	104
PARAMETRIC_VOLUME_2D_ELEMENT_COORDINATE_SYSTEM	ENTITY	104
PARAMETRIC_VOLUME_3D_ELEMENT_COORDINATE_SYSTEM	ENTITY	104
PATH	ENTITY	42
PCURVE	ENTITY	42
PCURVE_OR_SURFACE	TYPE	42
PERCENTAGE_LAMINATE_DEFINITION	ENTITY	209
PERCENTAGE_PLY_DEFINITION	ENTITY	209
PERSON	ENTITY	41
PERSON_AND_ORGANIZATION	ENTITY	41
PERSON_AND_ORGANIZATION_ASSIGNMENT	ENTITY	41
PERSON_AND_ORGANIZATION_ROLE	ENTITY	41
PERSON_ORGANIZATION_ITEM	TYPE	209
PERSON_ORGANIZATION_SELECT	TYPE	41

NAME	KIND	PART
PERSONAL_ADDRESS	ENTITY	41
PLACEMENT	ENTITY	42
PLANE	ENTITY	42
PLANE_2D_ELEMENT_PROPERTY	ENTITY	104
PLANE_2D_ELEMENT_PURPOSE	TYPE	104
PLANE_ANGLE_MEASURE	TYPE	41
PLANE_ANGLE_MEASURE_WITH_UNIT	ENTITY	41
PLANE_ANGLE_UNIT	ENTITY	41
PLANE_CURVE_2D_ELEMENT_DESCRIPTOR	ENTITY	104
PLANE_CURVE_2D_ELEMENT_REPRESENTATION	ENTITY	104
PLANE_SURFACE_2D_ELEMENT_DESCRIPTOR	ENTITY	104
PLANE_SURFACE_2D_ELEMENT_REPRESENTATION	ENTITY	104
PLANE_VOLUME_2D_ELEMENT_DESCRIPTOR	ENTITY	104
PLANE_VOLUME_2D_ELEMENT_REPRESENTATION	ENTITY	104
PLY_LAMINATE_DEFINITION	ENTITY	209
PLY_LAMINATE_SEQUENCE_DEFINITION	ENTITY	209
POINT	ENTITY	42
POINT_AND_VECTOR	ENTITY	209
POINT_CONSTRAINT	ENTITY	104
POINT_ELEMENT_MATRIX	ENTITY	104
POINT_ELEMENT_REPRESENTATION	ENTITY	104
POINT_FREEDOM_ACTION_DEFINITION	ENTITY	104
POINT_FREEDOM_AND_VALUE_DEFINITION	ENTITY	104
POINT_FREEDOM_VALUES	ENTITY	104
POINT_ON_CURVE	ENTITY	42
POINT_ON_SURFACE	ENTITY	42
POINT_PATH	ENTITY	209
POINT_REPLICA	ENTITY	42
POINT_REPRESENTATION	ENTITY	209
POLY_LOOP	ENTITY	42
POLYLINE	ENTITY	42
POSITIVE_LENGTH_MEASURE	TYPE	41
POSITIVE_PLANE_ANGLE_MEASURE	TYPE	41
POSITIVE_RATIO_MEASURE	TYPE	41
PRECISION_QUALIFIER	ENTITY	45
PREFERRED_SURFACE_CURVE_REPRESENTATION	TYPE	42
PRODUCT	ENTITY	41
PRODUCT_CATEGORY	ENTITY	41
PRODUCT_CATEGORY_RELATIONSHIP	ENTITY	41
PRODUCT_CONCEPT	ENTITY	44
PRODUCT_CONCEPT_CONTEXT	ENTITY	41
PRODUCT_CONTEXT	ENTITY	41
PRODUCT_DEFINITION	ENTITY	41
PRODUCT_DEFINITION_CONTEXT	ENTITY	41
PRODUCT_DEFINITION_EFFECTIVITY	ENTITY	41
PRODUCT_DEFINITION_FORMATION	ENTITY	41
PRODUCT_DEFINITION_FORMATION_RELATIONSHIP	ENTITY	41
PRODUCT_DEFINITION_FORMATION_WITH_SPECIFIED_SOURCE	ENTITY	41
PRODUCT_DEFINITION_RELATIONSHIP	ENTITY	41
PRODUCT_DEFINITION_SHAPE	ENTITY	41

NAME	KIND	PART
PRODUCT_DEFINITION_USAGE	ENTITY	44
PRODUCT_DEFINITION_WITH_ASSOCIATED_DOCUMENTS	ENTITY	41
PRODUCT_MATERIAL_COMPOSITION_RELATIONSHIP	ENTITY	45
PRODUCT_RELATED_PRODUCT_CATEGORY	ENTITY	41
PROMISSORY_USAGE_OCCURRENCE	ENTITY	44
PROPERTY_DEFINITION	ENTITY	41
PROPERTY_DEFINITION_REPRESENTATION	ENTITY	41
QUALIFIED_REPRESENTATION_ITEM	ENTITY	45
QUALITATIVE_UNCERTAINTY	ENTITY	45
QUANTIFIED_ASSEMBLY_COMPONENT_USAGE	ENTITY	44
QUASI_UNIFORM_CURVE	ENTITY	42
QUASI_UNIFORM_SURFACE	ENTITY	42
RATIO_MEASURE	TYPE	41
RATIO_UNIT	ENTITY	41
RATIONAL_B_SPLINE_CURVE	ENTITY	42
RATIONAL_B_SPLINE_SURFACE	ENTITY	42
RECTANGULAR_COMPOSITE_SURFACE	ENTITY	42
RECTANGULAR_TRIMMED_SURFACE	ENTITY	42
REPARAMETRISED_COMPOSITE_CURVE_SEGMENT	ENTITY	42
REPRESENTATION	ENTITY	43
REPRESENTATION_CONTEXT	ENTITY	43
REPRESENTATION_ITEM	ENTITY	43
REPRESENTATION_MAP	ENTITY	43
REPRESENTATION_RELATIONSHIP	ENTITY	43
REPRESENTATION_RELATIONSHIP_WITH_TRANSFORMATION	ENTITY	43
RESULT	ENTITY	104
RESULT_ANALYSIS_STEP	ENTITY	104
RESULT_LINEAR_MODES_AND_FREQUENCIES_ANALYSIS_SUB_STEP	ENTITY	104
RESULT_LINEAR_STATIC_ANALYSIS_SUB_STEP	ENTITY	104
RETENTION	ENTITY	209
RETENTION_ASSIGNMENT	ENTITY	209
RETENTION_ITEM	TYPE	209
REVERSIBLE_TOPOLOGY	TYPE	42
REVERSIBLE_TOPOLOGY_ITEM	TYPE	42
SCALAR	TYPE	104
SEAM_CURVE	ENTITY	42
SECOND_IN_MINUTE	TYPE	41
SECURITY_CLASSIFICATION	ENTITY	41
SECURITY_CLASSIFICATION_ASSIGNMENT	ENTITY	41
SECURITY_CLASSIFICATION_ITEM	TYPE	209
SECURITY_CLASSIFICATION_LEVEL	ENTITY	41
SERIAL_NUMBERED_EFFECTIVITY	ENTITY	41
SET_OF_REVERSIBLE_TOPOLOGY_ITEM	TYPE	42
SHAPE_ASPECT	ENTITY	41
SHAPE_ASPECT_RELATIONSHIP	ENTITY	41
SHAPE_DEFINITION	TYPE	41
SHAPE_DEFINITION_REPRESENTATION	ENTITY	41
SHAPE_DIMENSION_REPRESENTATION	ENTITY	47
SHAPE_FUNCTION	TYPE	104
SHAPE_REPRESENTATION	ENTITY	41

NAME	KIND	PART
SHAPE_REPRESENTATION_RELATIONSHIP	ENTITY	41
SHELL	TYPE	42
SHELL_BASED_SURFACE_MODEL	ENTITY	42
SHELL_BASED_WIREFRAME_MODEL	ENTITY	42
SHELL_BASED_WIREFRAME_SHAPE_REPRESENTATION	ENTITY	502
SI_PREFIX	TYPE	41
SI_UNIT	ENTITY	41
SI_UNIT_NAME	TYPE	41
SIMPLE_PLANE_2D_ELEMENT_PROPERTY	ENTITY	104
SINGLE_POINT_CONSTRAINT_ELEMENT	ENTITY	104
SINGLE_POINT_CONSTRAINT_ELEMENT_VALUES	ENTITY	104
SMEARED_MATERIAL_DEFINITION	ENTITY	209
SOLID_ANGLE_MEASURE	TYPE	41
SOLID_ANGLE_MEASURE_WITH_UNIT	ENTITY	41
SOLID_ANGLE_UNIT	ENTITY	41
SOLID_CONSTRAINT	ENTITY	104
SOLID_FREEDOM_ACTION_DEFINITION	ENTITY	104
SOLID_FREEDOM_AND_VALUE_DEFINITION	ENTITY	104
SOLID_FREEDOM_VALUES	ENTITY	104
SOLID_MODEL	ENTITY	42
SOURCE	TYPE	41
SPECIFIED_HIGHER_USAGE_OCCURRENCE	ENTITY	44
SPECIFIED_STATE	ENTITY	104
SPHERICAL_POINT	ENTITY	104
SPHERICAL_SURFACE	ENTITY	42
STANDARD_UNCERTAINTY	ENTITY	45
STATE	ENTITY	104
STATE_COMPONENT	ENTITY	104
STATE_DEFINITION	ENTITY	104
STATE_RELATIONSHIP	ENTITY	104
STATE_WITH_HARMONIC	ENTITY	104
STATIONARY_MASS	ENTITY	104
STRUCTURAL_RESPONSE_PROPERTY	ENTITY	104
STRUCTURAL_RESPONSE_PROPERTY_DEFINITION_REPRESENTATION	ENTITY	104
SUBSTRUCTURE_ELEMENT_REPRESENTATION	ENTITY	104
SUBSTRUCTURE_NODE_REFERENCE	ENTITY	104
SUBSTRUCTURE_NODE_RELATIONSHIP	ENTITY	104
SUPPLIED_PART_RELATIONSHIP	ENTITY	209
SUPPORTED_ITEM	TYPE	41
SURFACE	ENTITY	42
SURFACE_2D_EDGE	TYPE	104
SURFACE_2D_ELEMENT_BASIS	ENTITY	104
SURFACE_2D_ELEMENT_BOUNDARY_CONSTANT_SPECIFIED_SURFACE_VARIABLE_VALUE	ENTITY	104
SURFACE_2D_ELEMENT_BOUNDARY_CONSTANT_SPECIFIED_VARIABLE_VALUE	ENTITY	104
SURFACE_2D_ELEMENT_BOUNDARY_EDGE_CONSTANT_SPECIFIED_SURFACE_VARIABLE_VALUE	ENTITY	104
SURFACE_2D_ELEMENT_BOUNDARY_EDGE_CONSTANT_SPECIFIED_VARIABLE_VALUE	ENTITY	104

NAME	KIND	PART
SURFACE_2D_ELEMENT_BOUNDARY_EDGE_LOCATION_POINT_SURFACE_VARIABLE_VALUES	ENTITY	104
SURFACE_2D_ELEMENT_BOUNDARY_EDGE_LOCATION_POINT_VARIABLE_VALUES	ENTITY	104
SURFACE_2D_ELEMENT_BOUNDARY_EDGE_NODAL_SPECIFIED_VARIABLE_VALUES	ENTITY	104
SURFACE_2D_ELEMENT_BOUNDARY_EDGE_WHOLE_EDGE_VARIABLE_VALUE	ENTITY	104
SURFACE_2D_ELEMENT_BOUNDARY_LOCATION_POINT_SURFACE_VARIABLE_VALUES	ENTITY	104
SURFACE_2D_ELEMENT_BOUNDARY_NODAL_SPECIFIED_VARIABLE_VALUES	ENTITY	104
SURFACE_2D_ELEMENT_BOUNDARY_WHOLE_FACE_VARIABLE_VALUE	ENTITY	104
SURFACE_2D_ELEMENT_CONSTANT_SPECIFIED_VARIABLE_VALUE	ENTITY	104
SURFACE_2D_ELEMENT_CONSTANT_SPECIFIED_VOLUME_VARIABLE_VALUE	ENTITY	104
SURFACE_2D_ELEMENT_COORDINATE_SYSTEM	TYPE	104
SURFACE_2D_ELEMENT_DESCRIPTOR	TYPE	104
SURFACE_2D_ELEMENT_FIELD_VARIABLE_DEFINITION	ENTITY	104
SURFACE_2D_ELEMENT_GROUP	ENTITY	104
SURFACE_2D_ELEMENT_INTEGRATED_MATRIX	ENTITY	104
SURFACE_2D_ELEMENT_INTEGRATED_MATRIX_WITH_DEFINITION	ENTITY	104
SURFACE_2D_ELEMENT_INTEGRATION	ENTITY	104
SURFACE_2D_ELEMENT_LENGTH_INTEGRATION	TYPE	104
SURFACE_2D_ELEMENT_LENGTH_INTEGRATION_EXPLICIT	ENTITY	104
SURFACE_2D_ELEMENT_LENGTH_INTEGRATION_RULE	ENTITY	104
SURFACE_2D_ELEMENT_LOCATION_POINT_VARIABLE_VALUES	ENTITY	104
SURFACE_2D_ELEMENT_LOCATION_POINT_VOLUME_VARIABLE_VALUES	ENTITY	104
SURFACE_2D_ELEMENT_NODAL_SPECIFIED_VARIABLE_VALUES	ENTITY	104
SURFACE_2D_ELEMENT_OUTPUT_REFERENCE	TYPE	104
SURFACE_2D_ELEMENT_REPRESENTATION	TYPE	104
SURFACE_2D_ELEMENT_VALUE_AND_LOCATION	ENTITY	104
SURFACE_2D_ELEMENT_VALUE_AND_VOLUME_LOCATION	ENTITY	104
SURFACE_2D_FACE	TYPE	104
SURFACE_2D_NODE_FIELD_AGGREGATED_VARIABLE_VALUES	ENTITY	104
SURFACE_2D_NODE_FIELD_SECTION_VARIABLE_VALUES	ENTITY	104
SURFACE_2D_NODE_FIELD_VARIABLE_DEFINITION	ENTITY	104
SURFACE_2D_STATE_COORDINATE_SYSTEM	TYPE	104
SURFACE_2D_SUBSTRUCTURE_ELEMENT_REFERENCE	ENTITY	104
SURFACE_2D_WHOLE_ELEMENT_VARIABLE_VALUE	ENTITY	104
SURFACE_3D_EDGE	TYPE	104
SURFACE_3D_ELEMENT_BASIS	ENTITY	104
SURFACE_3D_ELEMENT_BOUNDARY_CONSTANT_SPECIFIED_SURFACE_VARIABLE_VALUE	ENTITY	104
SURFACE_3D_ELEMENT_BOUNDARY_CONSTANT_SPECIFIED_VARIABLE_VALUE	ENTITY	104
SURFACE_3D_ELEMENT_BOUNDARY_EDGE_CONSTANT_SPECIFIED_SURFACE_VARIABLE_VALUE	ENTITY	104
SURFACE_3D_ELEMENT_BOUNDARY_EDGE_CONSTANT_SPECIFIED_VARIABLE_VALUE	ENTITY	104
SURFACE_3D_ELEMENT_BOUNDARY_EDGE_LOCATION_POINT_SURFACE_	ENTITY	104

NAME	KIND	PART
VARIABLE_VALUES		
SURFACE_3D_ELEMENT_BOUNDARY_EDGE_LOCATION_POINT_VARIABLE_VALUES	ENTITY	104
SURFACE_3D_ELEMENT_BOUNDARY_EDGE_NODAL_SPECIFIED_VARIABLE_VALUES	ENTITY	104
SURFACE_3D_ELEMENT_BOUNDARY_EDGE_WHOLE_EDGE_VARIABLE_VALUE	ENTITY	104
SURFACE_3D_ELEMENT_BOUNDARY_LOCATION_POINT_SURFACE_VARIABLE_VALUES	ENTITY	104
SURFACE_3D_ELEMENT_BOUNDARY_NODAL_SPECIFIED_VARIABLE_VALUES	ENTITY	104
SURFACE_3D_ELEMENT_BOUNDARY_WHOLE_FACE_VARIABLE_VALUE	ENTITY	104
SURFACE_3D_ELEMENT_CONSTANT_SPECIFIED_VARIABLE_VALUE	ENTITY	104
SURFACE_3D_ELEMENT_CONSTANT_SPECIFIED_VOLUME_VARIABLE_VALUE	ENTITY	104
SURFACE_3D_ELEMENT_COORDINATE_SYSTEM	TYPE	104
SURFACE_3D_ELEMENT_DESCRIPTOR	ENTITY	104
SURFACE_3D_ELEMENT_FIELD_INTEGRATION	TYPE	104
SURFACE_3D_ELEMENT_FIELD_INTEGRATION_EXPLICIT	ENTITY	104
SURFACE_3D_ELEMENT_FIELD_INTEGRATION_RULE	ENTITY	104
SURFACE_3D_ELEMENT_FIELD_VARIABLE_DEFINITION	ENTITY	104
SURFACE_3D_ELEMENT_GROUP	ENTITY	104
SURFACE_3D_ELEMENT_INTEGRATED_MATRIX	ENTITY	104
SURFACE_3D_ELEMENT_INTEGRATED_MATRIX_WITH_DEFINITION	ENTITY	104
SURFACE_3D_ELEMENT_INTEGRATION	ENTITY	104
SURFACE_3D_ELEMENT_LOCATION_POINT_VARIABLE_VALUES	ENTITY	104
SURFACE_3D_ELEMENT_LOCATION_POINT_VOLUME_VARIABLE_VALUES	ENTITY	104
SURFACE_3D_ELEMENT_NODAL_SPECIFIED_VARIABLE_VALUES	ENTITY	104
SURFACE_3D_ELEMENT_OUTPUT_REFERENCE	TYPE	104
SURFACE_3D_ELEMENT_REPRESENTATION	ENTITY	104
SURFACE_3D_ELEMENT_VALUE_AND_LOCATION	ENTITY	104
SURFACE_3D_ELEMENT_VALUE_AND_VOLUME_LOCATION	ENTITY	104
SURFACE_3D_FACE	TYPE	104
SURFACE_3D_NODE_FIELD_AGGREGATED_VARIABLE_VALUES	ENTITY	104
SURFACE_3D_NODE_FIELD_SECTION_VARIABLE_VALUES	ENTITY	104
SURFACE_3D_NODE_FIELD_VARIABLE_DEFINITION	ENTITY	104
SURFACE_3D_STATE_COORDINATE_SYSTEM	TYPE	104
SURFACE_3D_SUBSTRUCTURE_ELEMENT_REFERENCE	ENTITY	104
SURFACE_3D_WHOLE_ELEMENT_VARIABLE_VALUE	ENTITY	104
SURFACE_CONSTRAINT	ENTITY	104
SURFACE_CURVE	ENTITY	42
SURFACE_ELEMENT_LOCATION	ENTITY	104
SURFACE_ELEMENT_PROPERTY	ENTITY	104
SURFACE_ELEMENT_PURPOSE	TYPE	104
SURFACE_ELEMENT_VARIABLE	TYPE	104
SURFACE_FREEDOM_ACTION_DEFINITION	ENTITY	104
SURFACE_FREEDOM_AND_VALUE_DEFINITION	ENTITY	104
SURFACE_FREEDOM_VALUES	ENTITY	104
SURFACE_MATRIX_PROPERTY_TYPE	TYPE	104
SURFACE_MODEL	TYPE	42

NAME	**KIND**	**PART**
SURFACE_OF_LINEAR_EXTRUSION	ENTITY	42
SURFACE_OF_REVOLUTION	ENTITY	42
SURFACE_PATCH	ENTITY	42
SURFACE_POSITION_WEIGHT	ENTITY	104
SURFACE_REPLICA	ENTITY	42
SURFACE_SCALAR_VARIABLE	TYPE	104
SURFACE_SECTION	ENTITY	104
SURFACE_SECTION_ELEMENT_LOCATION	ENTITY	104
SURFACE_SECTION_ELEMENT_LOCATION_ABSOLUTE	ENTITY	104
SURFACE_SECTION_ELEMENT_LOCATION_DIMENSIONLESS	ENTITY	104
SURFACE_SECTION_FIELD	ENTITY	104
SURFACE_SECTION_FIELD_CONSTANT	ENTITY	104
SURFACE_SECTION_FIELD_VARYING	ENTITY	104
SURFACE_SECTION_INTEGRATION	TYPE	104
SURFACE_SECTION_INTEGRATION_EXPLICIT	ENTITY	104
SURFACE_SECTION_INTEGRATION_RULE	ENTITY	104
SURFACE_SECTION_POSITION_WEIGHT	ENTITY	104
SURFACE_TENSOR2_2D_VARIABLE	TYPE	104
SURFACE_VECTOR_2D_VARIABLE	TYPE	104
SURFACE_VECTOR_3D_VARIABLE	TYPE	104
SURFACE_VOLUME_ELEMENT_LOCATION	ENTITY	104
SWEPT_SURFACE	ENTITY	42
SYMMETRIC_TENSOR2_2D	TYPE	104
SYMMETRIC_TENSOR2_3D	TYPE	104
SYMMETRIC_TENSOR4_2D	TYPE	104
SYMMETRIC_TENSOR4_3D	TYPE	104
SYMMETRY_CONTROL	ENTITY	104
SYSTEM_AND_FREEDOM	ENTITY	104
TENSOR_REPRESENTATION_ITEM	ENTITY	104
TENSOR_TYPE	TYPE	104
TENSOR1	TYPE	104
TENSOR1_2D	TYPE	104
TENSOR1_3D	TYPE	104
TEXT	TYPE	41
THERMODYNAMIC_TEMPERATURE_MEASURE	TYPE	41
THERMODYNAMIC_TEMPERATURE_MEASURE_WITH_UNIT	ENTITY	41
THERMODYNAMIC_TEMPERATURE_UNIT	ENTITY	41
THICKNESS_LAMINATE_DEFINITION	ENTITY	209
TIME_MEASURE	TYPE	41
TIME_MEASURE_WITH_UNIT	ENTITY	41
TIME_UNIT	ENTITY	41
TOPOLOGICAL_REPRESENTATION_ITEM	ENTITY	42
TOROIDAL_SURFACE	ENTITY	42
TRANSFORMATION	TYPE	43
TRANSFORMATION_WITH_DERIVED_ANGLE	ENTITY	209
TRANSITION_CODE	TYPE	42
TRIMMED_CURVE	ENTITY	42
TRIMMING_PREFERENCE	TYPE	42
TRIMMING_SELECT	TYPE	42
TYPE_QUALIFIER	ENTITY	45

NAME	KIND	PART
UNCERTAINTY_MEASURE_WITH_UNIT	ENTITY	43
UNCERTAINTY_QUALIFIER	ENTITY	45
UNIFORM_CURVE	ENTITY	42
UNIFORM_SURFACE	ENTITY	42
UNIFORM_SURFACE_SECTION	ENTITY	104
UNIFORM_SURFACE_SECTION_LAYERED	ENTITY	104
UNIT	TYPE	41
UNSPECIFIED_VALUE	TYPE	104
VALUE_QUALIFIER	TYPE	45
VECTOR	ENTITY	42
VECTOR_OR_DIRECTION	TYPE	42
VERSIONED_ACTION_REQUEST	ENTITY	41
VERTEX	ENTITY	42
VERTEX_LOOP	ENTITY	42
VERTEX_POINT	ENTITY	42
VERTEX_SHELL	ENTITY	42
VOLUME_2D_EDGE	TYPE	104
VOLUME_2D_ELEMENT_BASIS	ENTITY	104
VOLUME_2D_ELEMENT_BOUNDARY_CONSTANT_SPECIFIED_VARIABLE_VALUE	ENTITY	104
VOLUME_2D_ELEMENT_BOUNDARY_EDGE_CONSTANT_SPECIFIED_VOLUME_VARIABLE_VALUE	ENTITY	104
VOLUME_2D_ELEMENT_BOUNDARY_EDGE_LOCATION_POINT_VOLUME_VARIABLE_VALUES	ENTITY	104
VOLUME_2D_ELEMENT_BOUNDARY_EDGE_NODAL_SPECIFIED_VARIABLE_VALUES	ENTITY	104
VOLUME_2D_ELEMENT_BOUNDARY_EDGE_WHOLE_EDGE_VARIABLE_VALUE	ENTITY	104
VOLUME_2D_ELEMENT_BOUNDARY_LOCATION_POINT_VARIABLE_VALUES	ENTITY	104
VOLUME_2D_ELEMENT_BOUNDARY_NODAL_SPECIFIED_VARIABLE_VALUES	ENTITY	104
VOLUME_2D_ELEMENT_BOUNDARY_WHOLE_FACE_VARIABLE_VALUE	ENTITY	104
VOLUME_2D_ELEMENT_CONSTANT_SPECIFIED_VARIABLE_VALUE	ENTITY	104
VOLUME_2D_ELEMENT_COORDINATE_SYSTEM	TYPE	104
VOLUME_2D_ELEMENT_DESCRIPTOR	TYPE	104
VOLUME_2D_ELEMENT_FIELD_INTEGRATION	TYPE	104
VOLUME_2D_ELEMENT_FIELD_INTEGRATION_EXPLICIT	ENTITY	104
VOLUME_2D_ELEMENT_FIELD_INTEGRATION_RULE	ENTITY	104
VOLUME_2D_ELEMENT_FIELD_VARIABLE_DEFINITION	ENTITY	104
VOLUME_2D_ELEMENT_GROUP	ENTITY	104
VOLUME_2D_ELEMENT_INTEGRATED_MATRIX	ENTITY	104
VOLUME_2D_ELEMENT_INTEGRATED_MATRIX_WITH_DEFINITION	ENTITY	104
VOLUME_2D_ELEMENT_LOCATION_POINT_VARIABLE_VALUES	ENTITY	104
VOLUME_2D_ELEMENT_NODAL_SPECIFIED_VARIABLE_VALUES	ENTITY	104
VOLUME_2D_ELEMENT_OUTPUT_REFERENCE	TYPE	104
VOLUME_2D_ELEMENT_REPRESENTATION	TYPE	104
VOLUME_2D_ELEMENT_VALUE_AND_LOCATION	ENTITY	104
VOLUME_2D_FACE	TYPE	104
VOLUME_2D_NODE_FIELD_VARIABLE_DEFINITION	ENTITY	104

NAME	KIND	PART
VOLUME_2D_SUBSTRUCTURE_ELEMENT_REFERENCE	ENTITY	104
VOLUME_2D_WHOLE_ELEMENT_VARIABLE_VALUE	ENTITY	104
VOLUME_3D_EDGE	TYPE	104
VOLUME_3D_ELEMENT_BASIS	ENTITY	104
VOLUME_3D_ELEMENT_BOUNDARY_CONSTANT_SPECIFIED_VARIABLE_VALUE	ENTITY	104
VOLUME_3D_ELEMENT_BOUNDARY_EDGE_CONSTANT_SPECIFIED_VOLUME_VARIABLE_VALUE	ENTITY	104
VOLUME_3D_ELEMENT_BOUNDARY_EDGE_LOCATION_POINT_VOLUME_VARIABLE_VALUES	ENTITY	104
VOLUME_3D_ELEMENT_BOUNDARY_EDGE_NODAL_SPECIFIED_VARIABLE_VALUES	ENTITY	104
VOLUME_3D_ELEMENT_BOUNDARY_EDGE_WHOLE_EDGE_VARIABLE_VALUE	ENTITY	104
VOLUME_3D_ELEMENT_BOUNDARY_LOCATION_POINT_VARIABLE_VALUES	ENTITY	104
VOLUME_3D_ELEMENT_BOUNDARY_NODAL_SPECIFIED_VARIABLE_VALUES	ENTITY	104
VOLUME_3D_ELEMENT_BOUNDARY_WHOLE_FACE_VARIABLE_VALUE	ENTITY	104
VOLUME_3D_ELEMENT_CONSTANT_SPECIFIED_VARIABLE_VALUE	ENTITY	104
VOLUME_3D_ELEMENT_COORDINATE_SYSTEM	TYPE	104
VOLUME_3D_ELEMENT_DESCRIPTOR	ENTITY	104
VOLUME_3D_ELEMENT_FIELD_INTEGRATION	TYPE	104
VOLUME_3D_ELEMENT_FIELD_INTEGRATION_EXPLICIT	ENTITY	104
VOLUME_3D_ELEMENT_FIELD_INTEGRATION_RULE	ENTITY	104
VOLUME_3D_ELEMENT_FIELD_VARIABLE_DEFINITION	ENTITY	104
VOLUME_3D_ELEMENT_GROUP	ENTITY	104
VOLUME_3D_ELEMENT_INTEGRATED_MATRIX	ENTITY	104
VOLUME_3D_ELEMENT_INTEGRATED_MATRIX_WITH_DEFINITION	ENTITY	104
VOLUME_3D_ELEMENT_LOCATION_POINT_VARIABLE_VALUES	ENTITY	104
VOLUME_3D_ELEMENT_NODAL_SPECIFIED_VARIABLE_VALUES	ENTITY	104
VOLUME_3D_ELEMENT_OUTPUT_REFERENCE	TYPE	104
VOLUME_3D_ELEMENT_REPRESENTATION	ENTITY	104
VOLUME_3D_ELEMENT_SHAPE	TYPE	104
VOLUME_3D_ELEMENT_VALUE_AND_LOCATION	ENTITY	104
VOLUME_3D_FACE	TYPE	104
VOLUME_3D_NODE_FIELD_VARIABLE_DEFINITION	ENTITY	104
VOLUME_3D_SUBSTRUCTURE_ELEMENT_REFERENCE	ENTITY	104
VOLUME_3D_WHOLE_ELEMENT_VARIABLE_VALUE	ENTITY	104
VOLUME_AGGREGATED_VARIABLE	TYPE	104
VOLUME_ANGULAR_VARIABLE	TYPE	104
VOLUME_ELEMENT_LOCATION	ENTITY	104
VOLUME_ELEMENT_PURPOSE	TYPE	104
VOLUME_MEASURE	TYPE	41
VOLUME_MEASURE_WITH_UNIT	ENTITY	41
VOLUME_POSITION_WEIGHT	ENTITY	104
VOLUME_SCALAR_VARIABLE	TYPE	104
VOLUME_TENSOR2_3D_VARIABLE	TYPE	104
VOLUME_UNIT	ENTITY	41
VOLUME_VARIABLE	TYPE	104

NAME	KIND	PART
VOLUME_VECTOR_3D_VARIABLE	TYPE	104
WEEK_IN_YEAR_NUMBER	TYPE	41
WEEK_OF_YEAR_AND_DAY_DATE	ENTITY	41
WHOLE_MODEL_ANALYSIS_MESSAGE	ENTITY	104
WHOLE_MODEL_MODES_AND_FREQUENCIES_ANALYSIS_MESSAGE	ENTITY	104
WIRE_SHELL	ENTITY	42
WIREFRAME_MODEL	TYPE	42
YEAR_NUMBER	TYPE	41