Advances in Engineering Software 127 (2019) 96–105

Contents lists available at ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

Research paper

Relating structural test and FEA data with STEP AP209

R. Lanza[*,a,b], J. Haenisch[a], K. Bengtsson[a], T. Rølvåg[b]

[a] Jotne EPM Technology AS, Grenseveien 107, Oslo 0663, Norway
[b] Norwegian University of Science and Technology, Richard Birkelandsvei 2B, Trondheim, Norway

ARTICLE INFO

Keywords:
STEP ISO 10303
FEM Analysis
Structural testing
Data exchange
Simulation data management

ABSTRACT

This paper proposes a method for incorporating FEA data and structural test data into one common standardized data model based on the ISO 10303 STEP Standard [1]. The proposed method takes advantage of data structures and elements defined in STEP AP209 Edition 2 [2] to provide traceability between analysis and testing phases; information such as sensor and finite elements, test and FEA load cases, and test and FEA results are included. It also presents an introduction to STEP and AP209e2, and discusses how it can be used in a Simulation Data Management environment.

1. Introduction

Simulation and structural testing plays a big role in the development of complex products. As Moore's Law continues to evolve, greater computational power and storage becomes available for use. This has led to an ever-increasing amount of simulations, especially as design optimization through simulation and analysis becomes more common. The higher computational power allows engineers to perform more complex analyses with higher fidelity than ever before. Properly applied, high fidelity methods can lead to more optimized and safer products.

Enormous amounts of data are generated by these methods that must be managed effectively and efficiently. Problems arise when these data must be stored for reuse in different domains or when they have to be archived for a longer term. The large amount of data means finding information becomes more difficult. Files in different formats, for different applications, spread over multiple locations and companies further complicates the situation. A popular solution to these difficulties is often declared to be Simulation Data Management (SDM) and Product Data Management (PDM). These solutions make organizing simulation and CAD data together with other engineering information more efficient, but have focused more on the CAD aspects of data management. The aerospace industry (among others) has recognized the growing challenges related to SDM and PDM for analysis and simulation data and have been active in promoting SDM and PDM solutions.

Still with SDM, users are often locked to proprietary formats of the software initially used for their design and simulation, causing complications when different partners are using different software. SDM is not the main focus of this paper, but as we will see, AP209 is not only used as a file format but it could also be the backbone of the data model for a software system (including Data Management tools).

The reliability of simulation data depends on their validation by physical tests. For safety critical systems, authorities may require this relationship to be traceable. Test data, therefore, need to be managed together with corresponding simulation data. This adds to the complexity of the data management task. A typical (and simplified) engineering process that involves structural testing is as follows:

1. A simulation is performed and results are saved in the CAE software's native format.
2. Based on the results, actuator and sensor locations are chosen for a structural test.
3. Parameters for controlling the test are developed based on simulation results.
4. Tests are performed and loads and results are exported from the test equipment to a test specific format.
5. Test results and simulated results are compared and reconciled.
6. Results are summarized in test reports and delivered to consuming organizations.

Companies often have their own internal work-flows to manage interactions between the analysis and testing organizations during test planning and preparations up and throughout test execution. Additional work-flows are used to compare, reconcile, document and distribute the product testing results.

These work-flows can be performed manually or through automation but both rely on sets of agreed-upon definitions. The following types of information are a few examples of these definitions:

* Corresponding author at: Norwegian University of Science and Technology, Richard Birkelandsvei 2B, Trondheim, Norway.
E-mail addresses: remi.lanza@jotne.com (R. Lanza), jochen.haenisch@jotne.com (J. Haenisch), terje.rolvag@ntnu.no (T. Rølvåg).

https://doi.org/10.1016/j.advengsoft.2018.08.005
Received 5 March 2018; Received in revised form 27 July 2018; Accepted 12 August 2018

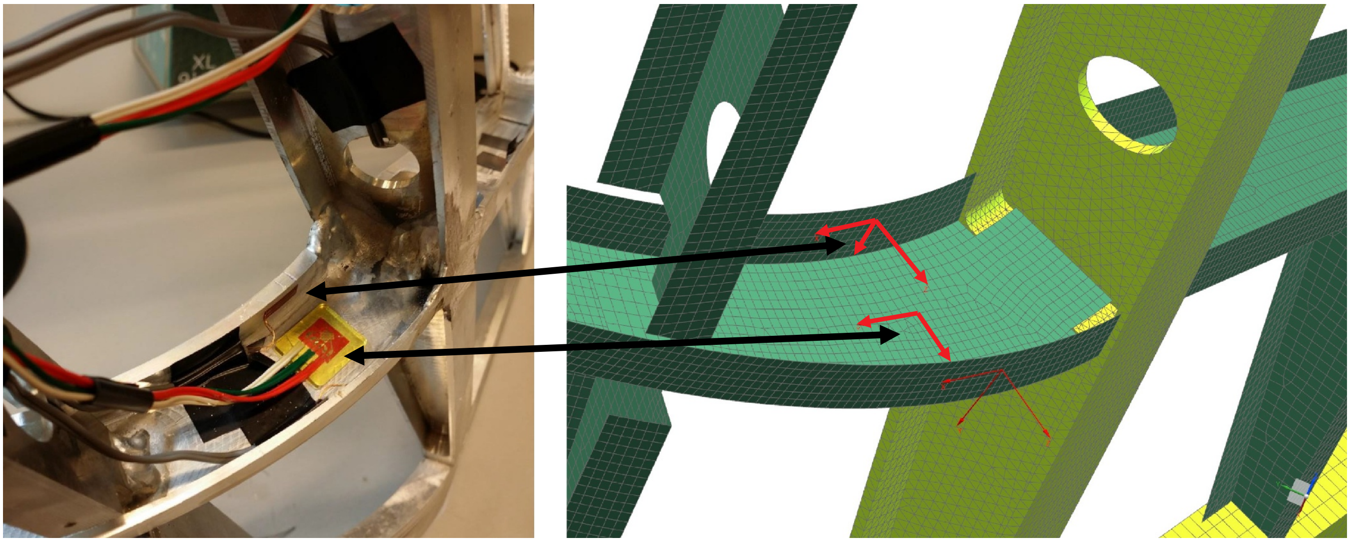

Fig. 1. Mapping sensor locations and orientations to FEM model.

1. Sensor distribution in the FE model frame of reference. (Figure 1)
2. Sensor orientations in the FE model frame of reference. (Figure 1)
3. Relation between corresponding test cases and analysis load cases.
4. Sensor mapping to channel IDs from the test equipment.
5. Information about applied filtering techniques on applied loads and sensor result data.

With these definitions, the correspondence of virtual and physical results can be validated against the testing requirements for data content and quality. Common data manipulation techniques, such as transforming the results to matching frame of reference, enable consistent predictions or comparisons and are highly dependent on the common understanding of the kinds of definitions described above.

These operations are performed in a variety of software tools with results typically output to Excel sheets for further analysis or reporting.

Data artifacts are generated at many of the steps in these workflows and must be retained to achieve full traceability. Examples of these data artifacts are the following:

1. Test Requirements
2. Test Plans and Procedures
3. FEM analysis files
4. FEM result files
5. Structural test output files
6. FEM-Structural test definition files
7. Comparison / correlation results
8. Reports

In certain industries there exist strong regulations on data retention of products. This is the case for the aerospace industry. As an example, the Federal Aviation Administration (FAA) in the United States, requires that *'Type design data must be retained and accessible for the lifespan of the product. It is possible that technical support for the original software will be terminated during the product lifespan, so your procedures manual must explain how access to the data will be retained or transitioned to a new software system.'* [3].

The goal of this paper is to validate that the AP209 data model has the capabilities to represent the above information, and keep the traceability between the different data fields. Thus, enabling the storage of a complete data set in a neutral and archive-friendly format.

Fig. 2presents an overview of the data which we want to represent in AP209, and how it relates together internally in a model.

In the next sections we briefly cover the background of the STEP ISO-10303 standard, followed by Section 3 where we present the outline of the proposed model, while Sections 4–6 go into specific details of the data model.

2. STEP ISO 10303

2.1. Background

Started in 1998, the goal of ISO 10303 was to standardize the representation of product data that are aggregated throughout the whole product life-cycle and across all relevant domains. The data model that STEP standardizes is written in the data modeling language EXPRESS [4], a lexical and graphical language which is both human and computer readable. EXPRESS is an object-oriented language using encapsulation and inheritance; it offers rich features for specifying population constraints.

Part 21 of the STEP standard [5] describes the ASCII representation of STEP, which is commonly known as the *STEP file format*. In addition, STEP defines an API to access product data in STEP compliant database repositories for data sharing. This is standardized Part 22 *SDAI, Standard Data Access Interface* [6]. Programming language interfaces for STEP data, so called language bindings, are specified in for example Part 23 [7] for C++. Having all these standardized methods for accessing STEP data, simplifies the creation of STEP based tools and software, and allows these to share a unified understanding of the data.

The standard is composed of a collection of parts, some of which covers the implementation methods of the standard, such as the parts mentioned above, while most parts specify the data models of the different product data domains supported by the standard, i.e. geometric representations, FEA, mathematical descriptions, product structures etc. Each of these are holding the definition of entities with their attributes and inheritance, which in an Object-Oriented Programming (OOP) view are essentially classes.

2.2. STEP Architecture

An important aspect of the STEP architecture is the use of higher level data models, which by using formal mapping specifications, maps to the integrated resources and the application resources of ISO 10303. Only a brief description of this process will be included in this paper. The reader is advised to study *STEP in a Nutshell* [8] and the *STEP Application Handbook* [9] for a thoroughly explanation of the STEP architecture.

The main idea is that an *Application Activity Model* (AAM) is used to describe the activities and data flows of a certain use case of the

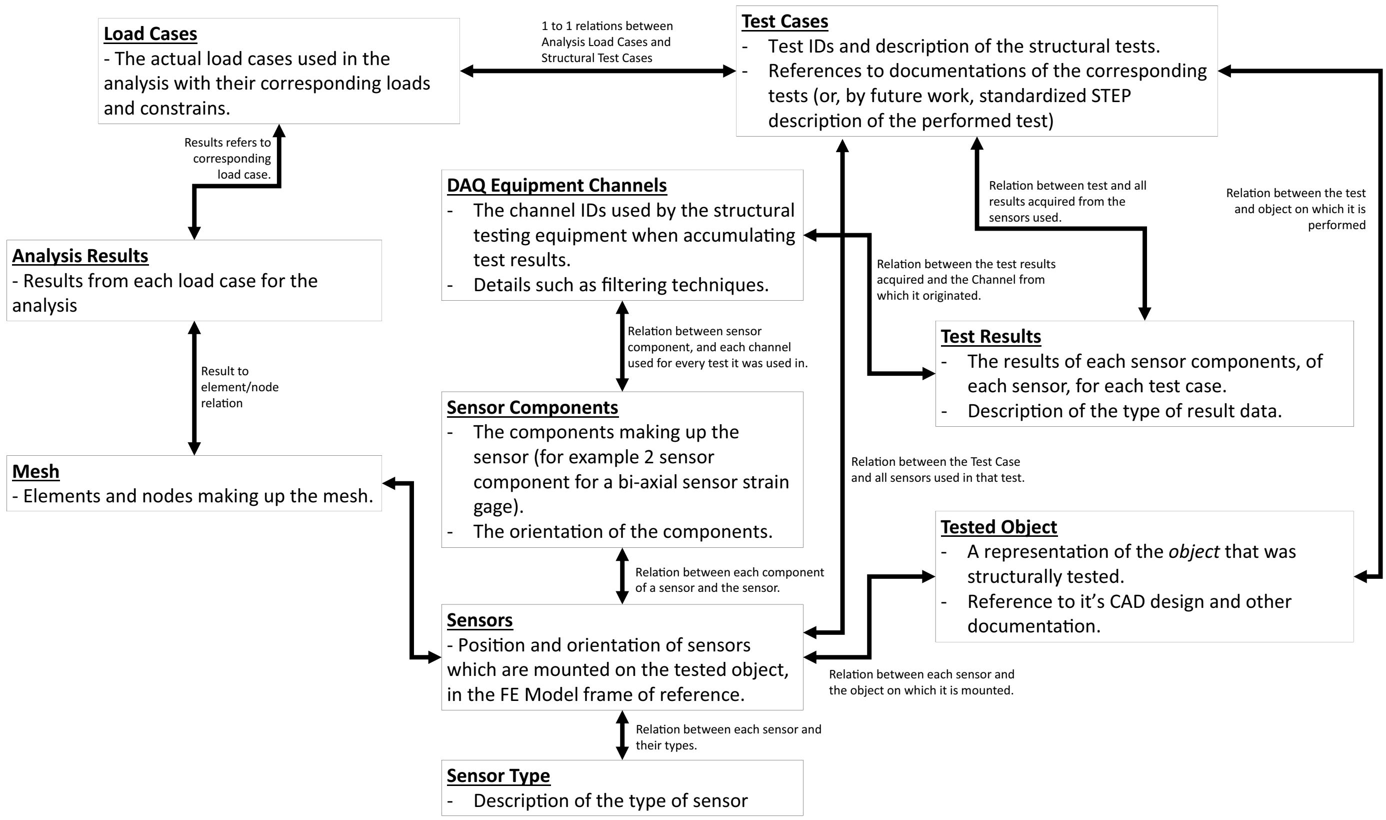

Fig. 2. Overview of main data represented in an AP209 model containing Analysis and Structural Test data, and how they relate.

Application Protocol (see Section 2.3). This is usually done by graphically illustrating the flows of types of data necessary for the to-be-developed STEP data model.

Based on the AAM, a formal *Application Reference Model* (ARM) can be designed; this may be modelled in the EXPRESS language. The intention is that this model is built by experts of the product data domain in question. Data objects and attributes are defined using terminology well understood in the specific domain.

The *Application Interpreted Model* (AIM) is the lowest level data model. AIMs contain the exact same information as the ARMs, but mapped by mapping specifications to a formally defined and generic format which is uniform across all usages of the STEP standard. Because of the complexity and genericness of the STEP standard, such mapping and encoding is usually done by a STEP expert in cooperation with domain experts.

The author of this paper has focused on the feasibility of using the AP209 AIM for structural test data; the formalities of publishing the findings of this study as part of ISO 10303 are not discussed in detail. This is a natural next task.

2.3. Application Protocols

Each Application Protocol (AP) focuses on a specific domain or phase in the product life-cycle. An AP specifies a single ARM to define its content, which, as described in the previous section, maps to an interoperable AIM. The AIM objects are defined in what are called *Integrated Resources* (IR). These IRs are in turn defined in the several parts of which the STEP standard consists.

A certain application or software supporting STEP, defines which AP it covers, that is, which share of the total STEP data model. APs are, thus, the view of the standard offered to implementors of data exchange, sharing and archiving solutions. STEP files refer one or several APs, but are all based on the same type of data structure, the so called PDM schema. They have the same high level definitions, allowing SDM and PDM tools to easily process files from different domains (i.e. CAD, FEA, manufacturing). STEP has also several managements concepts (such as requirements, assignments, classifications, roles, activities...) embedded within certain parts, which can be directly integrated within a Data Management tool.

Since the initial release of STEP in 1994, AP203 [10] and AP214 [11] have been the most successful Application Protocols, and are now widely used as exchange formats between CAD and PLM software.

In Fig. 3 we see how entities with inheritance and attributes are defined in an ISO 10,303 Part which in turn is used by an Application Protocol. The example shows two high level entities, **representation_item** and **representation** which belongs to Part 43 [12]. This Part has many generic entities that are used by all APs. Each entity may be a parent (*supertype*) of multiple entities which are defined in other Parts that further specializes them. For simplicity the figure shows a single inheritance branch (**representation_item** and **representation** actually have many child (*subtypes*) entities defined in other parts).

AP209, which covers the domain Analysis and Design, includes Part 42 [13], Part 43 and Part 104 [14], while AP242 [15], intended as a CAD format, includes only Part 42 and 43. Both AP209 and AP242 include shares of several other Parts which are not shown in the figure. Fig. 4 shows how **representation, element_representation** and **surface_3d_element_representation** are defined in the standard AP documents in the EXPRESS language.

A STEP file or database holds a population of instances of these entities, and can be interpreted by an application that implements the AP schema; an extract of such a STEP file is included in Section 3.1.

2.4. Application protocol 209

AP209 is called *Multidisciplinary analysis and design*, and is primarily meant to specify simulation solver relevant data for exchange, sharing and archival. An overview of the data that it can represent is shown in Fig. 5.

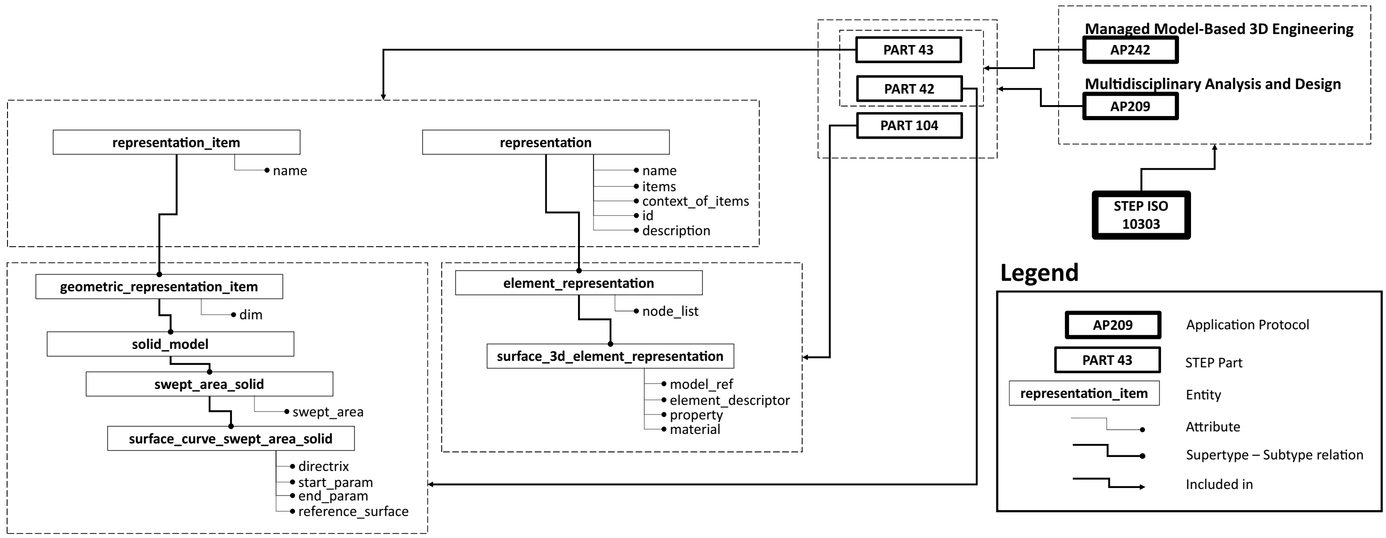

Fig. 3. Example of how entities are included in Parts which again are included in Application Protocols.

It covers the representation of composites, analysis definition and analysis results (FEM and CFD), design (CAD) and more. Note that an important aspect of AP209 is the capability of not only representing analysis and design *separately*, but also allowing the interconnection between both domains (such as relationships between mesh and loads, and the design geometry). Most modern FEM and CFD pre-processors have implemented geometry based mesh generation and load definitions in their applications, but the solver interfaces are still generally based on traditional ASCII cards formats.

Currently, only linear statics and linear modal analyses are completely supported in AP209. However, as noted by [16], the standard was designed to easily be updated to support non-linear analysis, as it already covers roughly 90% of this problem.

Multiple implementations of the standard have been performed, but these have been limited in scope, focusing on the exchange of composite data between design, analysis and manufacturing purposes or the basic FEA model entities. Several of these are summarized in [18].

Ongoing work and implementations of the standard are led by the LOTAR EAS: Engineering Analysis & Simulation Workgroup [19] which is co-chaired by Airbus and Boeing. They have been active in promoting commercial implementation of the AP209 standard with rigorous testing criteria.

As described earlier, an AP is composed of several STEP parts, which are principally schema that specify the content, form and structure of a set of entities (classes). Parts can be used in different APs, therefore many entities are general in nature. They can be viewed as *building-blocks* for representing certain classes of items or concepts. As we will see in the next section, these *building-blocks* or entities, can be used, not only to represent FEA and CAD, but also information concerning structural testing, as long as the new use of the existing structures is defined accordingly. The next sections describes an outline of a proposed structure for using AP209 to represent the additional data required for representing structural testing information. No extensions of the AP209 standard are suggested, but as will be discussed, future work may recommend changes or extensions.

3. The higher structure of a combined structural & FEA STEP model

3.1. Overview

This subsection introduces several key concepts used extensively in the subsequent sections.

In STEP high level items are represented as a **product**. By high level item we mean, *an Analysis, a CAD assembly, a CAD part, a manufactured part* etc. A product is a foundational concept that allow an item to be described, categorized, referenced, tracked, and versioned in ways that are familiar to modern day product data management users. Items that would not be considered a **product** could be a FEM element, a color definition, a property, a geometric shape etc.

```
ENTITY representation;
  name : label;
  items : SET[1:?] OF representation_item;
  context_of_items : representation_context;
DERIVE
  id : identifier := get_id_value (SELF);
  description : text := get_description_value (SELF);
WHERE
  WR1: SIZEOF (USEDIN (SELF, 'BASIC_ATTRIBUTE_SCHEMA.' + 'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;
  WR2: SIZEOF (USEDIN (SELF, 'BASIC_ATTRIBUTE_SCHEMA.' + 'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
END_ENTITY;


ENTITY surface_3d_element_representation
    SUBTYPE OF ( element_representation );
      model_ref           : fea_model_3d;
      element_descriptor  : surface_3d_element_descriptor;
      property            : surface_element_property;
      material            : element_material;
    UNIQUE
      ur1 : model_ref, SELF\representation.name;
    WHERE
      wr1:
      wr2:
      wr3:
      fu1:
END_ENTITY;
```

```
ENTITY element_representation
    SUPERTYPE OF (
        ONEOF (
            volume_3d_element_representation,
            axisymmetric_volume_2d_element_representation,
            plane_volume_2d_element_representation,
            surface_3d_element_representation,
            axisymmetric_surface_2d_element_representation,
            plane_surface_2d_element_representation,
            curve_3d_element_representation,
            axisymmetric_curve_2d_element_representation,
            plane_curve_2d_element_representation,
            point_element_representation,
            directionally_explicit_element_representation,
            explicit_element_representation,
            substructure_element_representation ) )
    SUBTYPE OF ( representation );
      node_list  : LIST [1 : ?] OF node_representation;
    WHERE
      wr1:
    END_ENTITY;
```

Fig. 4. Extract of the content in the AP209 document. (Some fields are left out for simplicity.)

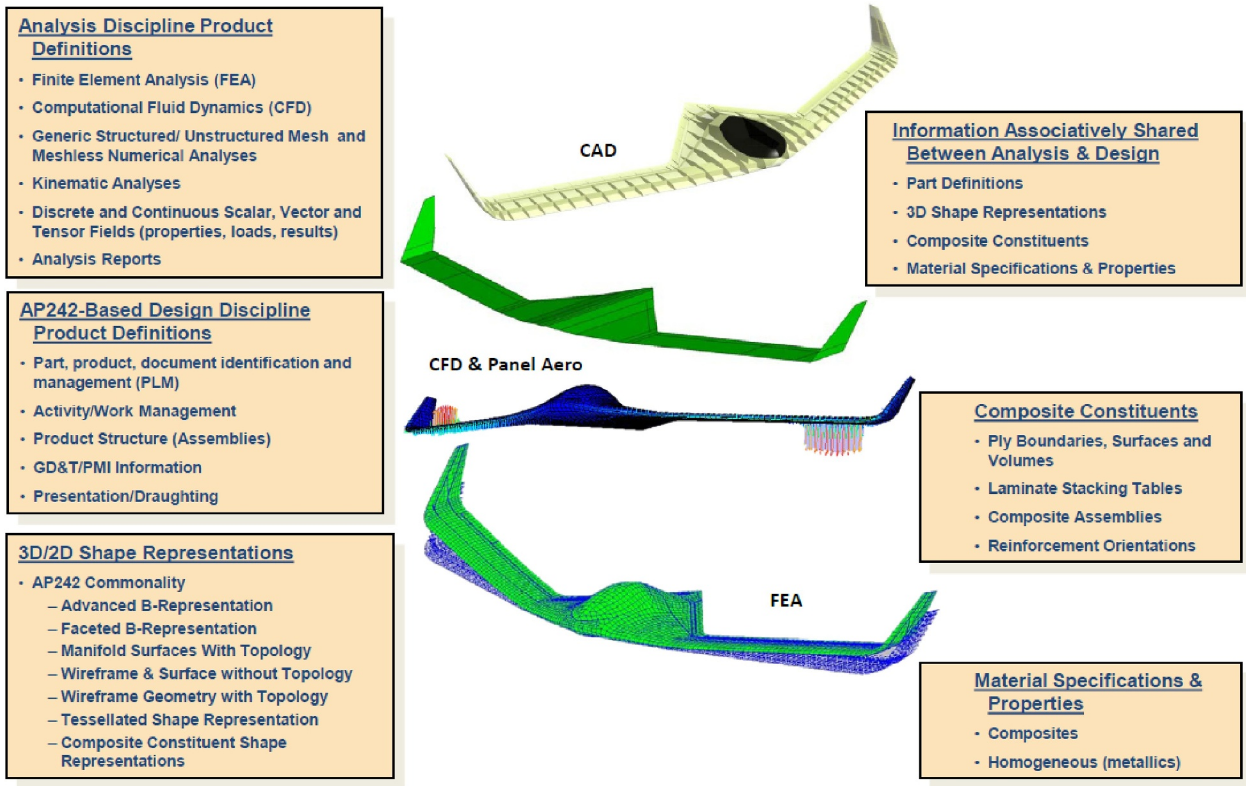

Fig. 5. Data which is supported by AP209 [17].

The **product** entity has certain mandatory attributes and related entities. For example, a **product** entity must have a version, a context and a category classifying the **product**. The detailed data entities that make up an instance of an analysis or CAD model, relate to a **product_definition** entity. In turn, this **product_definition** relate to a **product_definition_formation** which provides versioning for an instance of a **product**. These high level product entities also provide links to the **application_context** and **application_protocol_definition** that identify which STEP schema the data conforms to. Lastly, a variety of optional information about people, organizations, dates and times and other related metadata can be linked to these top level product entities.

This structure ensures that an application importing or accessing a **product** (such as an SDM tool) can understand what is being imported before handling the complete model. These high level entities serve to organize multiple STEP populations within the same system. Multiple STEP data-sets residing within a database removes the constraint that they be considered *files*. The constituents of each model or data set, are then identified by their relationship to the high level product entities. In this fashion, complex interrelated data-sets can be constructed which reduces data duplication.

An extract of a STEP P21 file (ASCII) showing some of these high level entities can be seen in Fig. 6. As shown, each instance of an entity has an identifier followed by the entity name. The attributes are enclosed by parentheses and comma separated. When an entity is an attribute of another entity, it is referenced by this identifier. Throughout the paper, graphical instantiation of this structure will be used (not to be confused with EXPRESS-G which is the standardized graphical representation of the EXPRESS language defined in Part 11). Instances are represented by boxes with the entity name in capital letters. Arrows show the referencing between instances. A string beside an arrow specifies the name of the attribute. In some cases STEP entity structures can be quite complex. If an entity box has its text in italic, it represents a simplification of a more complex structure, or a shortening of the entity name. Bold text beside an entity box is an additional description for the

```
(...)
#42= PRODUCT('1234','winglet analysis','',(#44));
#53= PRODUCT_DEFINITION_FORMATION('v.2','',#42);
#59= PRODUCT_DEFINITION('winglet analysis fine mesh',$,#53,#60);
#70= PRODUCT_RELATED_PRODUCT_CATEGORY('linear_static_analysis',$,(#42));
#44= PRODUCT_CONTEXT('design_context',#1,'design_context');
(...)
```

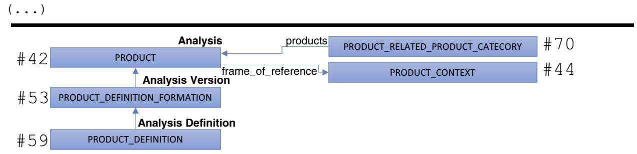

Fig. 6. Left: Extract of a STEP P21 file. Right: Graphical representation used in this paper.

reader to better relate the graphics to the context.

It is also important to understand that in addition to these high level entities, many of the low level entities such as nodes, elements and loads can hold addtional meta-data such as names, labels and descriptions. STEP post- and pre-processors can implement these, to describe intentions and comments regarding the creation, review and modification of the model.

3.2. *The analysis model*

The data structure of an Analysis STEP AP209 data set is well described in the Recommended Practices for AP209 [20].

A few details of the data structure will be discussed here, focusing on the parts that will have a relationship to the structural testing data.

In AP209 the analysis is represented by a **product** entity, which was described in the previous section. This analysis **product** has a version and a definition. The entity **product_definition_shape** represents the shape of the product used for the analysis. The **product_definition_shape** can include the idealization of the CAD model (abstraction), node sets, and more importantly, from the analysis perspective, the **fea_model_definition**. The **fea_model_definition** is the link to the

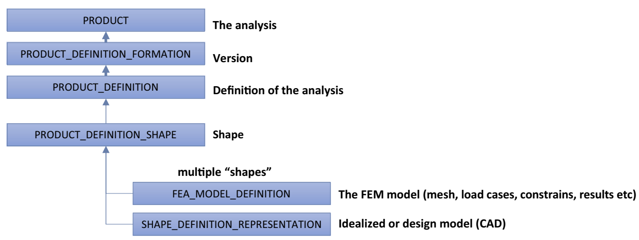

Fig. 7. High level entities in the Analysis Model.

nodes and elements making up the mesh **shape** and additional FEA related information. The whole analysis definition is then built up of entities linked to one another to give structure and meaning to the data.

Analysis load cases in AP209 are represented by **control_linear_static_analysis_step** entities that relates different **states**. Each **state** is a collector of loads, constraints or other nested **states**. Section 6.3 shows how the model relates the states to the actual test cases.

3.3. The structural test model

Fig. 7shows the high level structure of the FEA model. Fig. 8 introduces a similar structure representing the object that is being tested. The **product** in this case is the tested part which also has a version, definition and shape. The two versions are linked via relationship entities. The shape may be linked to the same Nominal Design data set, which is already related to the Analysis forming a consisten data set. The product being tested could be related to its own unique design version of needed.

Another **product** represents all the result data from all tests that relates to load cases in the Analysis Model. This product has a version as well, and multiple definitions with each representing the results from individual structural tests.

The sensors and the tests are also represented by STEP entities. The sensors are related to the tested part **product**, while the tests relate the sensors and the test results. Sensors and test representations are further discussed in Sections 4 and 5 respectively.

4. Sensors

There exists a wide variety of sensors such as strain gages, accelerometers, vibration sensors, displacements sensors and more. Many of these are assemblies of multiple sensors, for example a triaxial gage is just three sensors assembled together with specified angles between them.

To generically cover all types of sensors we represent each sensor as an assembly of multiple sensor components. Each sensor assembly and each component has its own **product** with a definition holding properties.

To avoid repetitive information, we introduce a **product** representing the type of sensors used. As an example, the specification of a tri-axial strain gage of a specific type, brand and model would be represented by one sensor type **product**. For each sensor of this type, mounted on the tested part, there exists a sensor assembly **product** having three individual sensor component **products**.

Each of the representations, sensor, sensor component and sensor type, are able to hold properties. Properties that are related to the sensor assembly:

1. **Position:** the position based on the coordinate system of the FE model
2. **Orientation:** the orientation of the sensor in the FE model
3. **Reference Element:** the element (or a set of elements) in the analysis model on which the sensor is placed
4. **Element Face ID:** an ID (or a set of IDs) representing the face of the elements on which the sensor is placed

Properties that are related to the sensor components:

1. **Direction:** the direction of the sensor component in the FE model
2. **ID:** An ID to number the sensor component

The definition of the complete set of properties for the sensor type is still ongoing. However suggested properties are:

1. **Sensor Type:** Strain gage / Accelerometer / Displacement Sensor
2. **Sensor Description:** Further description of the sensor type
3. **Manufacturer:** The name of the manufacturer
4. **Model name:** The model name of the sensor type
5. **Number of sensor components:** a number specifying the number of sensor components
6. **Angles:** For strain gages, a set of angles defining the angles between each sensor components

All these properties typically originate from different input sources, but are now contained within the same AP209 model and this facilitates the storing, organizing and sharing of the complete data set. Additional properties are planned to be added in future work to hold a comprehensive description of the sensors.

Properties that relates to the sensors, but are test case dependent are defined differently. For example filtering techniques performed on the data by the DAQ System (Data Acquisition system) are not necessarily the same for every usage of the sensor. These properties are related directly to the result data which we cover in the next section.

An example of how the sensor data structure can look in a STEP model is shown in Fig. 9. Note that the *reference element* property of the sensor assembly is a direct link to the actual element in the FE model, providing traceability between analysis and testing in the same model.

5. Structural tests

In STEP the generic entity **action** will be used to represent *the action of performing a structural test*. The items used in the test are assigned to this entity by a **applied_action_assignment**, which in turn assigns each item a role of *input* or *output* to the **action**. The *input* items to the test are the sensors and the tested part, while the *output* is the sensor result data for that particular test.

The **action_method** is the link to the description of how the test was performed. This could be in the form of a reference to a certain external document, or in a more structured form with STEP entities. The work

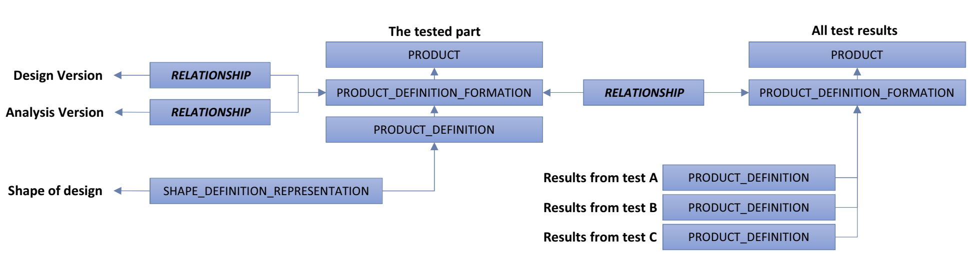

Fig. 8. Relations between FEA, Design, Tested Part and Test Results. In this case, there are three individual tests.

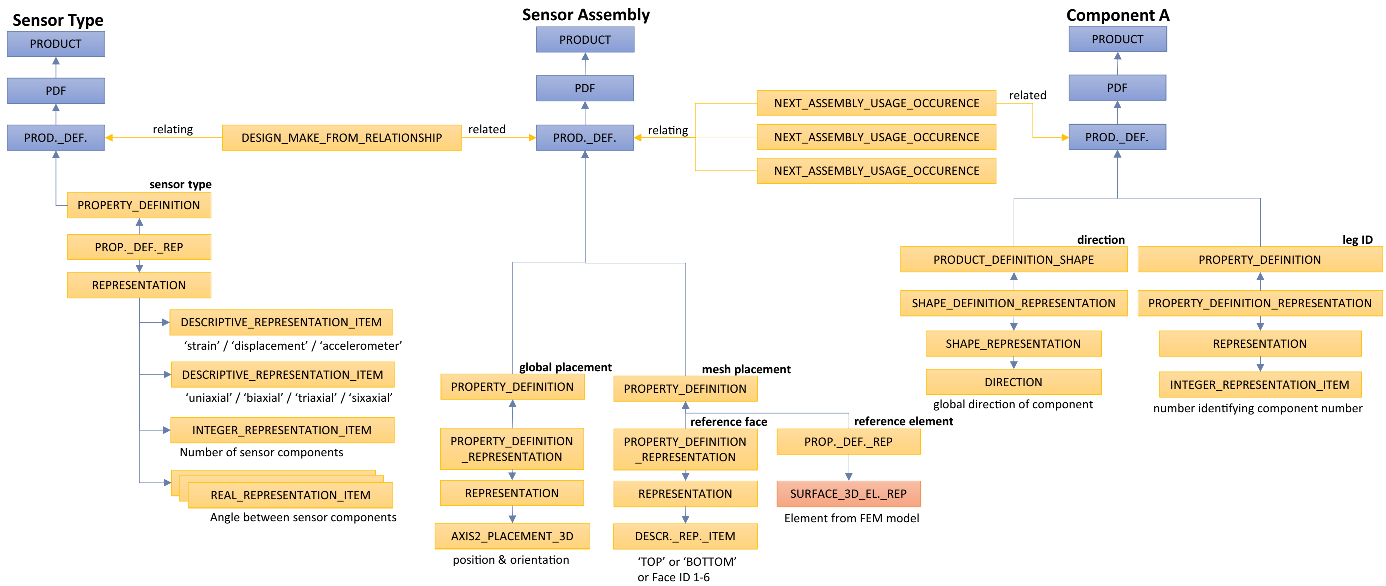

Fig. 9. Example of data structure for sensor with three sensor components (only one is shown).

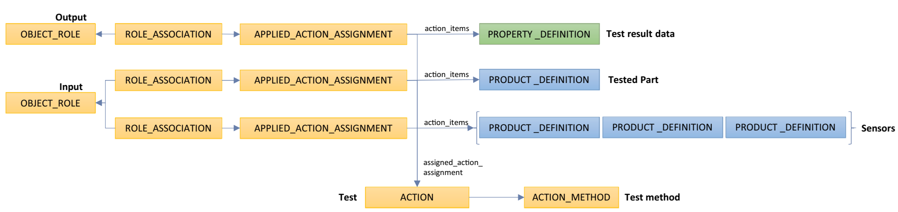

Fig. 10. Data structure for a test performed on a part with three sensors, resulting in a certain test result.

related to this is ongoing and is not presented in this paper.

6. Structural Test data

6.1. Structural test results

The original test result data coming from the test equipment software will typically be in the form of Excel files or other proprietary formats. The data can be extremely large, and it is generally expected that it has been filtered before being converted or imported to this STEP model.

The storing of test data in STEP is based on Part 50 *Mathematical constructs* [21]. The entity **listed_real_data** holds the values, but the complexity of this portion of the STEP standard requires multiple other entities to define what kind of data is held within it. The details of this data model are outside the scope of this paper, but readers are encouraged to review the Part 50 documentation. For simplicity we will define the entity ***data_array*** to represent an array of values. The information within this *entity* is an array of result data corresponding to the data output from *one* sensor component for *one* test case, the type of data (i.e. strain or displacement) and the size of the data. The**data_array** relates to a **property_definition** allowing us to use the result data as a property to other entities.

As seen in Fig. 11, relationships are used to group the **property_definition** and ***data_array*** results from each sensor component to **property_definitions** corresponding to the whole test case. The test case **property_definitions** reference the corresponding **product_definition** of the output data. These **property_definitions** are the same as those labeled as output of the **action** in Fig. 10.

Fig. 12is a combination of both Figs. 10 and 11 showing the overall relationship between the individual sensor results and the test actions.

6.2. Structural test result properties

In addition to the sensor properties presented in Section 4, we will now look at properties that are related to sensors, but that may vary for each test case. They are typically properties originating from the DAQ equipment and software used for retrieving test data.

This applies to properties such as:

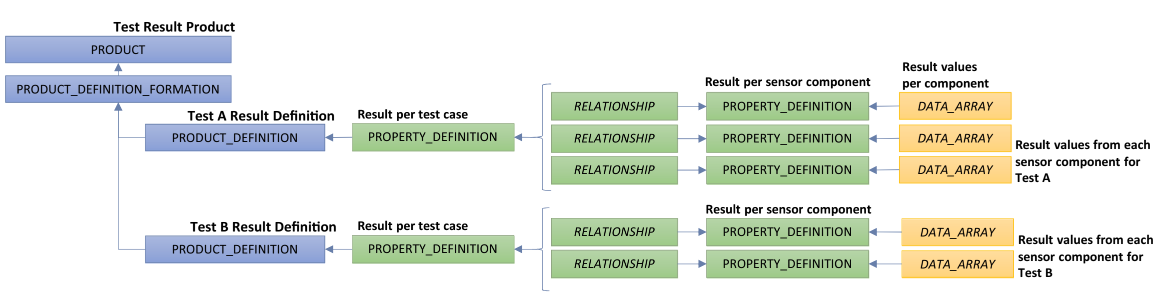

Fig. 11. Example of data structure showing the relation between the sensor result values and the output result data **product**.

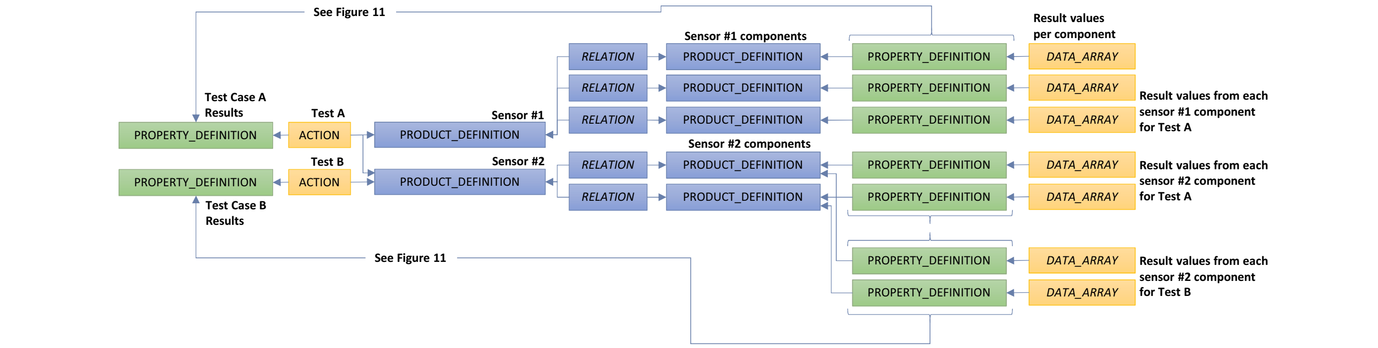

Fig. 12. Data structure showing relation between sensor results, sensors and tests. Here we have two tests and 2 sensors. One of the sensors (sensor 2) is used in both tests.

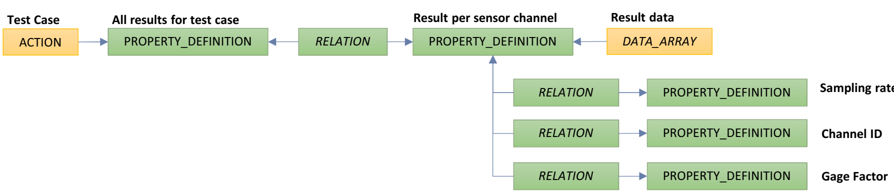

Fig. 13. The data values array references the result property of sensor a component, which is also referenced by properties that are unique for this sensor channel and test case.

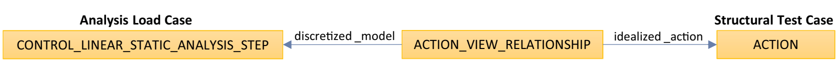

Fig. 14. Load cases from the Analysis data set are related to the Test Case of the Structural Test data set via the action_view_relationship.

1. The channel ID from the test Equipment
2. Filtering Techniques
3. Sample Rate
4. Scaling
5. Gage Factor

The use of relationships between **property_definition**s is again used to include these test case dependent properties. This is illustrated in Fig. 13.

6.3. *Structural test relation to analysis load case*

In Section 3.2 we presented how load cases are defined in an Analysis Model. The test cases are related to load cases via the **action_view_relationship** entity. This entity relates a *discretized model* (the analysis load case) to an *idealized action* (the test **action** that is being idealized) (Fig. 14).

7. Model development methodology

As AP209 is primarily meant to store and share CAD and simulation data, structural testing was not part of its original design. The first question when investigating the use of this standard for another domain such as structural testing, was if the standard itself required an extension: Does additional entities and types need to be added to the AP209 schema?

To answer the above, a careful examination of the AP209 schema was performed to get a detailed overview of which type of data the data model *can* represent. A good understanding of the whole schema was acquired during the development of a converter to translate FEM analyses in Nastran format to AP209.

The next step was to define which type of data from the structural testing domain needed to be included in the data model. These data types were then mapped to AP209 elements (entities, attributes, data types etc.). Careful attention was given to how to relate this domain to the analysis elements.

As noted previously, many of the STEP elements are generic, and can be used to represent a wide variety of data. However, the pre- and post-processors need to know how to interpret these generic constructs. An example is the entity **action**, a generic item, but with certain attributes to specify what the *action* represents (here, used to define the test case). This is where the *Application Reference Model* (as briefly discussed in 2.2) and documents such as Recommended Practices are required to specify the data model semantics.

The standard itself contains the formal description of every STEP element, while the Recommended Practices describe *how* it is intended to be used and implemented in applications. Such an implementors' guide is currently being developed for the testing domain to formally describe all the details presented here. In addition, to properly and formally introduce the results of this paper to the STEP standard, and make it available to the structural testing community, the AAMs and ARMs and their mapping to the AIM need to be developed. This would possibly also involve the introduction of new entity subtypes specifically for the domain of structural testing. The ARM shall include the concepts that are specific to the structural testing domain; they shall be mapped to the STEP resources as described in this paper. But for the purpose of this initial study, no extension of the standard is required.

After the mapping was defined from the test data to AP209 entities, another converter was created. This converter uses the results from structural tests in .csv format as well as files defining the sensors and test cases as input. The converter directly creates STEP data in an AP209 database (using Jotne's tools EDMS [23] and EDMopenSimDM [24]). The analysis related to the test case already resides within the database, allowing the converter to access it and create direct links between the new structural test data population and the analysis model.

The Simulation Data Management use case discussed in the introduction would utilize this converter function to construct a complete view of the product. A prototype is being performed to validate the usage of the model. An airplane winglet has been designed, simulated, manufactured and tested to imitate the different phases of product development. The data of each phase has been either exported or converted to STEP AP209 and imported to the EDMopenSimDM

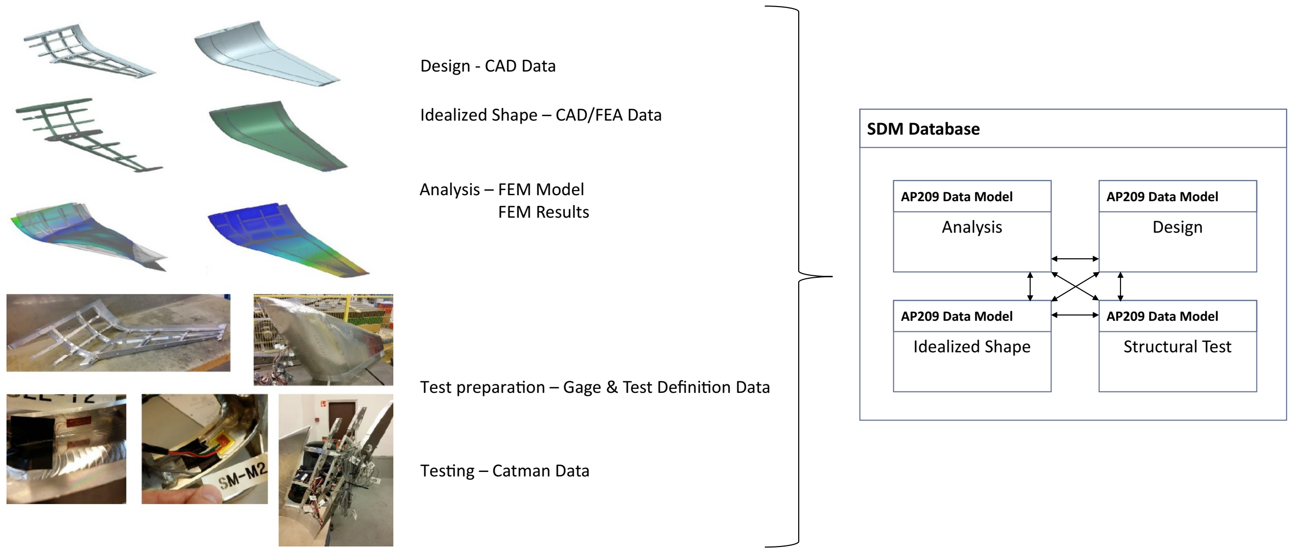

Fig. 15. Different AP209 data sets imported to the SDM tool.

application (Figure 15). This application uses AP209 as its database schema. The tool is being further developed to let the user access and manage the federated data.

8. Conclusion

We have now shown how the structural test related data can be represented in the AP209 data model, and how the relevant pieces of data can be connected to an engineering analysis and its results.

The purpose of SDM software is to manage and provide an overview of all the information related to simulations and give quick access to specific data. Having all the different aspects of the product in a consistent schema in a single database enables exactly this. If implemented properly, it enables the enterprise to utilize this data without having to open files in their original software.

Accessing information can easily be done by executing simple query functions on the single consistent and comprehensive data set. Examples of queries could be, retrieving the type of sensor used, the location of it on the mesh, getting the result data from a particular sensor for a particular test, and comparing it to the corresponding analysis results. Different views on the AP209 population can be implemented, such as an overview of all sensors that were used in a specific test, and their result values in both analysis and testing.

Besides representing contents data of analyses and structural tests, AP209 also provides the resources for data management. This includes defining who created a model, who accepted it, deadlines, tasks to be performed etc. These specifications can be directly linked to the entities that describes the analysis and structural test contents within the data sets.

The complete data set can then be exported to ASCII or binary STEP files that are compliant with the LOTAR (Long Term Archiving and Retrieval) specifications [22]. The resulting information can be shared with other systems conforming to the these standards, which enables project information to be archived or retrieved with all data still being traceable. To make the results of this study available to the structural testing community, AP209 should be updated and published as a new edition. The STEP resources seem to be sufficient to capture the information requirements discussed in this paper, but the AAM, the ARM and the mapping specification will need to be updated.

References

[1] 1994. ISO 10303-1:1994 Industrial automation systems and integration – Product data representation and exchange – Part 1: Overview and fundamental principles. Geneva (Switzerland): International Organization for Standardization (ISO).
[2] ISO 10303-209:2014 industrial automation systems and integration – product data representation and exchange – part 209: application protocol: multidisciplinary analysis and design. geneva (switzerland): international organization for standardization (ISO). 2014a.
[3] FAA-21-48 - using electronic modeling systems as primary type design data, u.s. department of transportation, federal aviation administration. 2010.
[4] ISO 10303-11:2004 industrial automation systems and integration – product data representation and exchange – part 11: description methods: the EXPRESS language reference manual. geneva (switzerland): international organization for standardization (ISO). 2004.
[5] ISO 10303-21:2016 industrial automation systems and integration – product data representation and exchange – part 21: implementation methods: clear text encoding of the exchange structure. geneva (switzerland): international organization for standardization (ISO). 2016.
[6] ISO 10303-22:1998 industrial automation systems and integration – product data representation and exchange – part 22: implementation methods: standard data access interface. geneva (switzerland): international organization for standardization (ISO). 1998.
[7] ISO 10303-23:2000 industrial automation systems and integration – product data representation and exchange – part 23: implementation methods: C++ language binding to the standard data access interface. geneva (switzerland): international organization for standardization (ISO). 2000a.
[8] Kramer T, Xu X. STEP In a nutshell, in: advanced design and manufacturing based on STEP. London: Springer - Verlag London Ltd.; 2009.
[9] Anonymous. STEP application handbook ISO 10303 version 3, SCRA, north charleston, south carolina, USA. 2006.
[10] ISO 10303-203:2011 industrial automation systems and integration – product data representation and exchange – part 203: application protocol: configuration controlled 3d designs of mechanical parts and assemblies. geneva (switzerland): international organization for standardization (ISO). 2011a.
[11] ISO 10303-2010:2001 industrial automation systems and integration – product data representation and exchange – part 214: application protocol: Core data for automotive mechanical design processes. geneva (switzerland): international organization for standardization (ISO). 2010.
[12] ISO 10303-43:2011 industrial automation systems and integration – product data representation and exchange – part 43: Integrated generic resource: representation structures. geneva (switzerland): international organization for standardization (ISO). 2011b.
[13] ISO 10303-42:2014 industrial automation systems and integration – product data representation and exchange – part 42: integrated generic resource: Geometric and topological representation. geneva (switzerland): international organization for standardization (ISO). 2014b.
[14] ISO 10303-104:2000 industrial automation systems and integration – product data representation and exchange – part 104: integrated application resource: Finite element analysis. geneva (switzerland): international organization for standardization (ISO). 2000b.
[15] ISO 10303-242:2014 industrial automation systems and integration – product data representation and exchange – part 242: application protocol: managed model-based 3d engineering. geneva (switzerland): international organization for standardization (ISO). 2014c.
[16] Hunten KA. CAD/FEA Integration with STEP AP209 technology and implementation. MSC aerospace users conference proceedings. 1997. http://web.mscsoftware.com/support/library/conf/auc97/p01297.pdf[accessed 18 january 2018]
[17] PDES Inc., ISO 10303-209 "Multidisciplinary Analysis and Design", http://www.ap209.org/introduction (accessed on 16 January 2018).
[18] Hunten KA, Feeney AB, Srinivasan V. Recent advances in sharing standardized STEP

composite structure design and manufacturing information. Comput.-Aided Des. 2013;45:1215–21.
[19] LOTAR (long term archiving and retrieval). LOTAR EAS: engineering analysis & simulation workgroup. http://www.lotar-international.org/lotar-workgroups/engineering-analysis-simulation/scope-activities.html(accessed on 19 January 2018).
[20] Hunten KA. Recommended practices for AP209ed2 10303-209:2014. CAx implementor forum. 2016.
[21] ISO 10303-50:2002 industrial automation systems and integration – product data representation and exchange – part 50: integrated generic resource: mathematical constructs. geneva (switzerland): international organization for standardization (ISO). 2002.
[22] NAS9300-001, long term archiving and retrieval of digital technical product documentation such as 3d, CAD and PDM data : part 101: structure, aerospace industries association of america inc.2017.
[23] EDMS (version 2.100.15) [STEP Data Manager software], Jotne EPM Technology AS.
[24] EDMopenSimDM (version 14.0) [Simulation Data Manager software], Jotne EPM Technology AS.

Remi Lanza completed his in M.Sc. in Mechanical Engineering in 2015 within the field of finite element analysis. Remi later acquired an interest for computer science, and joined Jotne EPM Technology in 2016 where he started his industrial Ph.D., which is a collaboration project between the Norwegian University of Science and Technology (NTNU) and Jotne. While pursuing his Ph.D., Remi has been exposed to several of the STEP ISO 10303 standards, especially STEP AP209. His initial research involves the usage of AP209 to facilitate data management and retention of simulation data and structural test data.

Jochen Haenisch leads the Aeronautics, Defence and Space business area in Jotne EPM Technology. He has contributed to and managed many implementations of the Jotne data interoperability tool EXPRESS Data Manager. These applied various STEP standards including ISO 10303-209 (Multidisciplinary analysis and design), ISO 10303-214 (automotive), ISO 10303-239 (product lifecycle support, PLCS) and ISO 15926 (oil&gas). In 1990 he entered into the ISO Subcommittee for Industrial Data, ISO/TC 184/SC 4, for many known as STEP. He regularly attends their plenary meetings as head of delegation for Norway. Currently he is deputy convenor of WG12, Common Resources.

Mr. Kjell Bengtsson, is a Vice President at Jotne, has a Mechanical Engineering background and a diploma in Marketing. He started out at Volvo Car and General Electric doing CAD/DB applications and later management positions, and is now VP at Jotne EPM Technology. Kjell has been exposed to STEP, PLCS and other related standards for the last 25 years and is actively involved in neutral database implementation projects in the most complex defense and aerospace sector projects. Kjell is a Member of the Board of PDES, Inc and supports other industry organizations like AIA/ASD, NIAG (NATO), FSI and more.

Prof. Rølvåg was born in Mo I Rana, 16/10-1963. Rølvåg holds a M.Sc. and a Ph.D. within finite element dynamics of elastic mechanisms and control from NTH. His publications are mainly within non-linear finite element dynamics and active damping of elastic mechanisms. He has been central in developing FEDEM, a finite element based modeling and simulation tool with multidisciplinary capabilities (see www.fedem.com). He has also established several engineering companies and optimized products for the automotive, offshore and aerospace industries. Prof. Rølvås research interests cover computer science applied for engineering applications focusing on simulation of behavior and strength of electromechanical products.