Doctoral theses at NTNU, 2020:416

Doctoral thesis

Remi Krister Sylvain Lanza

Improving and implementing the STEP ISO 10303 standard for design, analysis and structural test data correlation

NTNU
Norwegian University of Science and Technology
Thesis for the Degree of
Philosophiae Doctor
Faculty of Engineering
Department of Mechanical and Industrial
Engineering

NTNU
Norwegian University of
Science and Technology

Remi Krister Sylvain Lanza

Improving and implementing the STEP ISO 10303 standard for design, analysis and structural test data correlation

Thesis for the Degree of Philosophiae Doctor

Trondheim, December 2020

Norwegian University of Science and Technology
Faculty of Engineering
Department of Mechanical and Industrial Engineering

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the Degree of Philosophiae Doctor

Faculty of Engineering
Department of Mechanical and Industrial Engineering

ISBN 978-82-471-9399-0 (printed ver.)
ISBN 978-82-471-9676-2 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

Doctoral theses at NTNU, 2020:416

Printed by NTNU Grafisk senter

Table of Contents

List of Figures vi

List of Abbreviations viii

Preface 1

Acknowledgements 3

Abstract 5

1 Introduction 7

1.1 Industrial Context 7

1.2 Objective 9

1.3 Structure of Thesis 9

2 ISO 10303 STEP 11

2.1 The STEP Standard and Industrial Use 11

2.2 Architecture of the STEP Standard 12

2.3 ISO 10303 AP209 - Multidisciplinary Design and Analysis 21

3 Application development 27

3.1 STEP Converters 27

3.2 STEP Explorer 29

3.3 STEP in Simulation Data Management 32

4 Summary of Papers 37

4.1 Paper 1 - Relating Structural Test and FEA data with STEP AP209 37

4.2 Paper 2 - ISO 10303 AP209 - Why and how to embed nonlinear FEA . . 38

4.3 Paper 3 - Extending STEP AP209 for Nonlinear Finite Element Analysis . 39

4.4 Summary of STEP AP209 Extensions 40

5 Conclusion and Future Work 45

Appendices 53

A Main publications 55

A.1 Paper 1 - Relating Structural Test and FEA data with STEP AP209 55

A.2 Paper 2 - ISO 10303 AP209 - Why and how to embed nonlinear FEA . . 66

A.3 Paper 3 - Extending STEP AP209 for Nonlinear Finite Element Analysis . 98

B Secondary publication 123

B.1 Paper 4 - Open Simulation Data Management and Testing - The CRYSTAL Project 123

C Contact and Nonlinear Materials in STEP 143

C.1 Test Cases 143

C.2 Contact Interaction 148

C.3 Nonlinear Materials 152

D fea_interaction EXPRESS code 155

List of Figures

2.1 Engineering domains and overlap. 12

2.2 STEP Architectures and modules 13

2.3 STEP AP209 population 25

3.1 AP209 as a central format for converters 27

3.2 STEP Explorer instance box 30

3.3 STEP Explorer interface 31

3.4 STEP Explorer entity list 31

3.5 STEP Explorer instance search 31

3.6 STEP Explorer instance list 31

3.7 STEP Explorer instance diagrams 32

3.8 Winget use case 33

3.9 EDMopenSimDM interface 34

3.10 EDMopenSimDM product structure 34

3.11 FEM-Test Correlation tool interface 36

3.12 Sensors visualized on FEM mesh 36

C.1 Test case A geometry 144

C.2 Test case A mesh . . . 144
C.3 Test case A results . . . 145
C.4 Test case A displacements . . . 145
C.5 Test case B . . . 146
C.6 Test case B results . . . 147
C.7 Test case C . . . 147
C.8 Test case C, multilinear material model . . . 148
C.9 Test case C results . . . 148
C.10 Von Mises stress and plastic strain . . . 149
C.11 Element groups and faces . . . 149
C.12 New element group entities . . . 150
C.13 Instance diagram of element surface regions . . . 151
C.14 Hierachy of new FEA interaction entities . . . 151
C.15 Instance structure of linear material . . . 152
C.16 Instance structure of material with yield point . . . 153
C.17 Instance structure specifying a multilinear material model . . . 154

List of Abbreviations

AM	Application Module
AP	Application Protocol
API	Application Programming Interface
CAD	Computer Aided Design
CAE	Computer Aided Engineering
CAM	Computer Aided Manufacturing
CFD	Computational Fluid Dynamics
CNC	Computer Numerical Control
DAQ	Data Acquisition system
EDM	Express Data Manager
EDMS	Express Data Manager Supervisor
ESA	European Space Agency
FEA	Finite Element Analysis
FEM	Finite Element Method
HDF5	Hierarchical Data Format version 5
IAR	Integrated Application Resources
IGR	Integrated Generic Resources

ILS	Integrated Logistics Support
IoT	Internet of Things
ISO	International Organization for Standardization
NFR	The Research Council of Norway
PDM	Product Data Management
PLM	Product Lifecycle Management
SDAI	Standard Data Access Interface
SDM	Simulation Data Management
STEP	Standard for the Exchange of Product model data
XML	Extensible Markup Language

Preface

This study was realized as an industrial Ph.D. scheme; a collaboration between NTNU (Norwegian University of Science and Technology) and Jotne EPM Technology AS, and partly funded by NFR (The Research Council of Norway). Terje Rølvåg as the main supervisor from NTNU, and Jochen Haenisch as the main supervisor from Jotne. The work was performed over the course of 3.5 years; one year at the Department of Mechanical and Industrial Engineering at NTNU, and the remaining at Jotne's offices.

Three academic papers were written during the study; one published, one under review, and one submitted. These are included in an appendix of this document.

The thesis consists of an introduction, relevant background information for the attached papers, including details on how this work was used and implemented in Jotne's projects and software.

Acknowledgements

First of all I would like to give a huge thanks to Jotne, which gave me the opportunity for both a position in their company, and a Ph.D. education through NTNU. I also appreciate they let me be involved in projects involving interesting companies such as Lockheed Martin Aeronautics, ESA (European Space Agency), Boeing, and others. They gave me a chance to combine both my background education of Mechanical Engineering, and IT which was until that time only a hobby.

A very special thanks to my supervisor Jochen Haenisch, who has lead me through all my work over the last few years. His shared knowledge and advice has been crucial for completing this thesis, and I couldn't have done it without his support. His talent of performing very thoroughly and detailed checks of my papers and thesis has been very valuable!

A big thanks to my supervisor at NTNU, Terje Rølvåg, even though not being in the same location as me, Terje has always been easy to reach and quick to help. He has been very helpful in all the mechanical engineering side of this work, and has always had a good answer when I was stressed about which direction to go in this study.

From Jotne, I also want to thank Kjell Bengsston, aka. my boss. Kjell has also given me a lot of guidance, pushed me to go to conferences, and been very efficient to get people's attention towards my work. My co-worker, Olav Liestøl, deserves a special thanks for all his help and support when it comes to programming. The uncountable hours of debugging wouldn't be the same without his help!

I'd also like to thank everyone else at Jotne for helping me in different situations, and I look forward working with them for more years to come.

At the beginning of this study, a complicated mechanical structure had to be built. This was more complicated than expected, and I want to give a big thanks to the crew at the workshop, especially Børge and Carl-Magnus, for bringing it to completion.

Finally, and most importantly, my wife, Jahzel, to whom I got married in the middle of my Ph.D. studies, deserves all my gratitude for her encouragement and motivational support during my studies. My parents, Jean-Michel and Ingrid, and my sisters, Solveig and Karen, deserve of course a personal thanks, for their encouragement, and for making me think I have control over what I'm working on!

Abstract

The objective of this thesis was to improve the effectiveness of data management of physical test and simulation data, in the context of digital twins, PDM (Product Data Management)/PLM (Product Lifecycle Management) and SDM (Simulation Data Management).

Many different software applications, from different vendors are used for simulation, CAD (Computer Aided Design), PDM/PLM and other engineering activities. Most of them use different data storage formats and methods that are customized for the specific use of their application.

How can the lack of data interoperability among collaborating engineering applications be solved? Within and across companies sharing of data, for example, for creating and maintaining digital twins, becomes cumbersome when different parties use different applications. Files need to be converted, or tasks need to be re-done, potentially leading to loss of information. Managing project data becomes difficult, and with data originating from different sources, stored in different locations and companies, it is tough to keep track of what is where and in which version. This is especially important for digital twins, where different domain data need to be accessed by automatic processes.

This thesis addresses this problem specifically for the domains of FEM (Finite Element Method) and structural testing data. However, the fact that this is part of a larger context involving multiple other domains, is taken in consideration.

The STEP ISO 10303 standard is highly in focus throughout the study. This standard contains data models designed to cover as much as possible of the different engineering domains, across development life cycle stages.

The study shows how this standard can also be applied to represent and manage structural test data, including its relations to corresponding FEM analyses; this has never been done before. Implementations using and validating the new concepts were performed in converter applications and in a SDM tool.

To increase the FEM domain coverage of the standard, certain extensions of the standard are recommended after having been implemented and validated as part of this thesis. With these extensions the standard can be used for nonlinear FEM analysis, thus, allowing also such advanced analysis data to be shared among FEM solvers.

Chapter 1

Introduction

1.1 Industrial Context

Structural testing, analysis and their correlation has always been important when designing complex products and systems. These fields are tightly linked to the idea of a digital twin. The concept of digital twin, a virtual model of a product or process, has become increasingly important in many industries [1, 2]. A digital twin should collect all relevant information of a certain physical product or process, throughout its lifetime. Depending on the use-case, this would include data from many different domains. There are many technical aspects of a digital twin, such as, connectivity (IoT; *Internet of Things*), sensors, analytics, accessibility, data correlation, etc. This study focuses on the data representation aspect. Digital twin data need to be collected in a consistent product data model to enable the harvesting of the digital twin vision.

There exist multiple implementations of digital twins in different contexts, but as a relatively newly emerging topic, there are no standards for managing all the data that is involved. (The importance of standards, and the implications of the absence of standards, in the context of digital twins are discussed in [1, 3, 4, 5].)

In engineering industries, data management systems are used to keep a consistent and comprehensive overview of all data related to the development, operation, optimization loops and data tracing of products. These systems collect data and references to data, in a repository (in one location or distributed) to facilitate data management. Applications that handle this are known as PDM (Product Data Management) and PLM (Product Lifecycle Management) applications [6]; however, as they are not standard based, they do not give the industry full control over their own data.

Within engineering data management, there is also Simulation Data Management (SDM)

[7]. While PDM is data management on a higher level, SDM applications handle the organization of simulation data at a more detailed level [8]. These systems also lack standards compliance, and users are dependent on the specific tools supported by the SDM application they use.

In the context of digital twins, PDM, PLM and SDM, interoperability between data originating from different applications, from different domains, and across the product's entire lifecycle, is necessary to establish a consistent product data model.

Currently, data management applications, used by digital twins or not, handle interoperability by either (or both):

1. Holding references between related data on a file level, for example; a simulation file is related to a CAD file, or the documentation of a sensor is related to the results file of a physical test.

2. Holding references between related data on a data object level, for example; an object representing a sensor relates to its test result values, and a load case simulating the test.

The second option though, being more attractive, locks the user to the CAE (Computer Aided Engineering) applications provided by the vendor of their SDM system, and thus interoperability between applications is very limited.

The STEP ISO 10303 [9] standard was created as an interoperable (and common) data model across many engineering domains. It allows to relate engineering data cross-domain, and is thus very useful not only for domain specific applications, but also for PLM [10, 11, 12, 13, 14], SDM [15, 16], and digital twin models that span many domains.

The data scope of this thesis applies to the digital twins, for example, for predictive maintenance where measured data are compared to existing analysis and design data. The digital twin may derive, from analyzing the comprehensive and integrated data sets, the need to deviate from planned maintenance procedures. This thesis validates the completeness of STEP ISO 10303 AP209 [17] data sets for such queries. Other technical aspects of digital twin use cases, such as, data sample rate, data filtering and analytics, algorithms, frequency of analyses and comparisons, etc. are independent of this validation of the data availability aspect. The detailed recommendations of this thesis for AP209 are applicable to a few digital twin use cases; methodology and general conclusions concerning the use of STEP, however, should also be considered for a wider range of scenarios.

1.2 Objective

This thesis is a contribution to a critical review of the suitability of STEP ISO 10303 to capture all engineering information related to structural testing, analysis and their association. Within this wide field, the goal is to establish and validate methods to improve the effectiveness of simulation and test data management.

The focus of this thesis is mainly on the standard representation of:

- Structural test data, i.e. sensor information, sensor data, and test result data
- FEM (Finite Element Analysis) data
- The relation between the above two items
- All of the items above in the context of SDM applications

The objectives of this thesis in the context of data management, long term archiving, data interoperability and data traceability are as follows:

O.1 Validate that the ISO STEP 10303 standard may be used for storing, sharing, managing, and correlating design, simulation, and structural test data.

O.2 Identify the obstacles in using ISO STEP 10303 for data management and data exchange of FEM data.

These objectives are achieved by developing STEP based interoperability solutions among several commercial FEM and testing applications, and by integrating such data in a STEP compliant repository.

1.3 Structure of Thesis

Chapter 1 introduces the background and the goals of this study. Since the STEP ISO 10303 standard holds such central part of this thesis, a complete chapter, Chapter 2, is reserved for introducing and presenting its structure, architecture, and application methods. Chapter 3 presents some of the applications and projects which have been developed during the study or involved in the study. Summaries of the author's published and submitted academic articles, are discussed in Chapter 4. Finally we conclude the study and propose future work in Chapter 5.

The authors articles are included in Appendix A and B:

1. A.1 Relating Structural Test and FEA data with STEP AP209 [18]

- Published in *Advances in Engineering Software* (Main author)

2. A.2 ISO 10303 AP209 - Why and how to embed nonlinear FEA [19]
   - Under review with *Advances in Engineering Software* (Main author)
3. A.3 Extending STEP AP209 for Nonlinear Finite Element Analysis [20]
   - Under review with *Advances in Engineering Software* (Main author)
4. B.1 Open Simulation Data Management and Testing - The CRYSTAL Project [21]
   - Published in *NAFEMS World Congress 2017* (Co-author)

An additional Appendix C presents a study which was originally intended as an article, but due to limitations is here added as an appendix.

Chapter 2

ISO 10303 STEP

2.1 The STEP Standard and Industrial Use

ISO 10303, officially called *Industrial automation systems and integration - Product data representation and exchange*, and commonly known as STEP (*STandard for the Exchange of Product model data*), is an ISO standard defining data models for the representation of product, product development, and product usage information. The standard covers multiple engineering domains, including, but not limited to; PLM, PDM, CAD (Computer Aided Design), FEA (Finite Element Analysis), and CFD (Computational Fluid Dynamics).

Most applications use proprietary formats for their data storage. The problems with such formats are, 1) exchanging data between different systems is not always possible, and 2) systems may change their storage formats when introducing new versions.

To be freed from proprietary storage formats, the standard provides data models for all relevant engineering domains. The purpose of the standard is to enable a more seamless data integration for applications, both within the same, and across different domains. Throughout product development steps, multiple applications are used to perform activities both within the same, and across all steps. A lot of information may overlap across these activities, and without a common and central data model, consistency becomes difficult.

Although each domain covered by STEP have their own data models, as will be described in more details in section 2.2, all models share a common *sub*-data model, thus enabling interaoperability between them. Figure 2.1 shows the concept of overlapping domain data, were PLM is part of each domain.

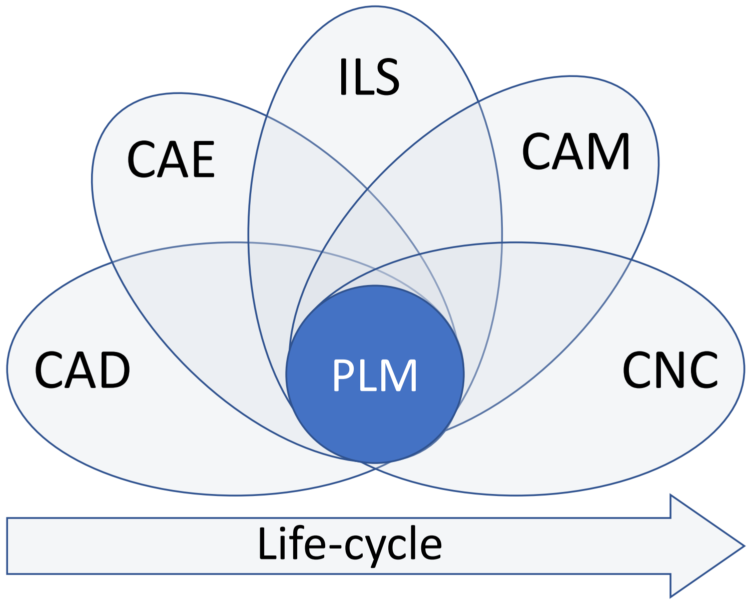


Figure 2.1: Multiple domains have certain overlap. A very central overlap is the PLM information.

The data models are written in the EXPRESS [22] language, discussed in 2.2.1, and can be mapped to any proprietary system that wishes to be STEP compliant.

2.2 Architecture of the STEP Standard

ISO 10303 is a collection of hundreds of documents each describing and defining different parts of the standard. The documents are divided in different categories. The main categories are the following;

- Description Methods
- Implementation Methods
- Integrated Generic Resources (IGR)
- Integrated Application Resources (IAR)
- Application Modules (AM)
- Application Protocols (AP)

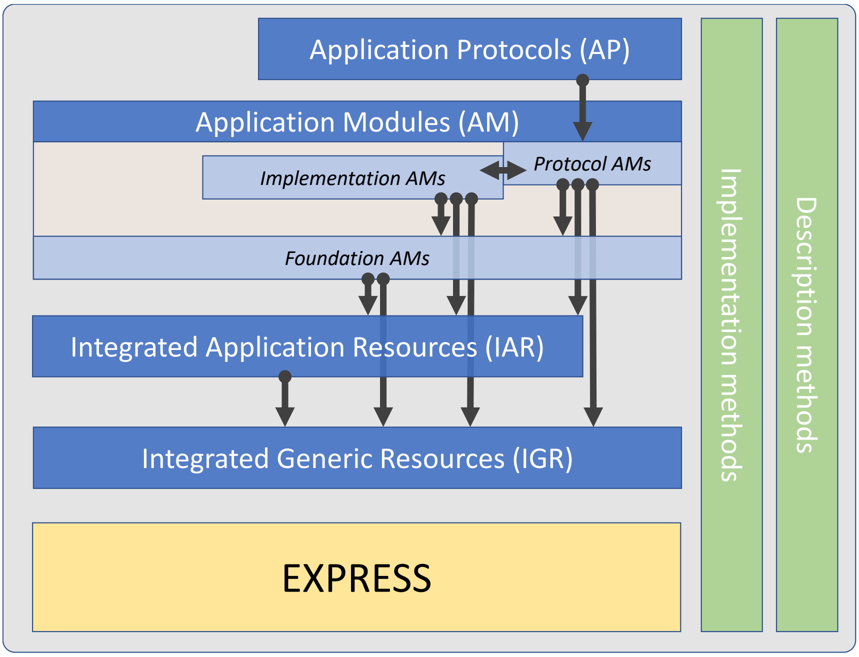


Figure 2.2: STEP architecture and modules. Each category boxes contains ISO documents. The arrows show between which categories there are allowable references.

2.2.1 Description and Implementation Methods

STEP EXPRESS

STEP represents the data models for these domains in the computer-interpretable language EXPRESS, which is itself documented by the standard.

The main component of EXPRESS data models are *entities*. Entities represent objects with properties and rules, and can be compared to classes in object-oriented languages. The properties, or attributes, may be classical data types such as; integers, reals and booleans, specialized data types defined by the data model, or other entities. As with classes, entities may inherit properties from other entities.

An example of an EXPRESS entity, **volume_3d_element_representation**, describing a FEA volume element, and its inherited parents, **representation** and **element_representation**, is shown in Listing 2.1. (In this document, EXPRESS entities are shown in **bold**.)

```

ENTITY representation
  SUPERTYPE OF ( left out for simplicity  ) ;
    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: left out for simplicity;
    WR2: left out for simplicity;
END_ENTITY;

ENTITY element_representation
  SUPERTYPE OF ( left out for simplicity )
  SUBTYPE OF (representation);
    node_list                   : LIST [1:?] OF node_representation;
  WHERE
    WR1: SIZEOF (QUERY(item <* node_list |
      'AP209_MULTIDISCIPLINARY_ANALYSIS_AND_DESIGN_MIM_LF.' +
      'GEOMETRIC_NODE' IN TYPEOF (item))) = 0;
END_ENTITY;

ENTITY volume_3d_element_representation
  SUBTYPE OF (element_representation);
    model_ref                   : fea_model_3d;
    element_descriptor          : volume_3d_element_descriptor;
    material                    : element_material;
  UNIQUE
    UR1: model_ref, SELF\representation.name;
  WHERE
    WR1: left out for simplicity;
    WR2: left out for simplicity;
    WR3: left out for simplicity;
    FU1:  required_3d_nodes (
      SELF\element_representation.node_list,
      element_descriptor.shape,
      element_descriptor\element_descriptor.topology_order);
END_ENTITY;
```

Listing 2.1: EXPRESS model of **representation**, **element_representation** and **volume_3d_element_representation**

The terms *SUBTYPE* and *SUPERTYPE* in Listing 2.1 refers to parent-child relationships, and are used to specify if an entity inherits from, or is the parent of a specific entity. In the snippet we also see how a volume element has attributes such as element material, element descriptor, FEA model, and list of nodes (inherited from its supertype **element_representation**). These attributes are other entities (or containers of entities) in the data model. The *WRx* and *FUx*, are so called *domain rules* or *WHERE rules*. These set restrictions on what is a valid occurrence of the entity. The specifications for such rules may be complex, and most are therefore left out from the example. *FU1*, in **volume_3d_**

element_representation, checks, with the use of the function *required_3d_nodes*, that the correct number of nodes are included in the *node_list* attribute. The required number of nodes is dependent on the shape and topological order of the element, and are therefore arguments to the function.

The rules, specified in EXPRESS, can be interpreted by applications and be used for automated validations of populations of the data model.

Storage forms of STEP

The data models describe the structure and semantics of the data. A population of this data can be stored in different ways:

- As an ASCII file following the STEP format defined in ISO 10303-21 [23].
- As a binary file using the HDF5 (Hierarchical Data Format Version 5) format [24]. ISO 10303-26 [25] defines the mapping of EXPRESS structures to HDF5 structures of STEP data.
- As an XML file following the specification in ISO 10303-28 [26].
- As a relational database by using the EXPRESS data model as a database schema.

Listing 2.2 shows a snippet of the ISO 10303-21 representation of a data population. It contains three FEA volume elements and some of its referenced instances. Each instance of an entity is identified by an identifier starting with # and is used when other instances reference it through attributes. For example `#1136` is an instance of the entity **volume_3d_element_representation**. Its first attribute, *name*, inherited from the entity **representation**, is defined as a *label* (which is defined as a string), is `'1'`. The subsequent attributes, inherited from **element_representation**, and defined in **volume_3d_element_representation**, are references to other instances. The attribute *element_descriptor*, references a **volume_3d_element_descriptor** with id `#1147`. This entity specifies the shape (hexahedron) and order (linear) of the element. This specification is used by the rule *FU1* under a validation of the model, to check that the required number of nodes is set.

```

#73= CARTESIAN_POINT('1',(0.,-4.,0.));
#75=  NODE('1',(#73),#28,#62);
#84=  NODE('2',(#82),#28,#62);
#104= NODE('7',(#102),#28,#62);

#62= FEA_MODEL_3D('Nastran job EAS test case ATS4m5',(#13),#28,
    'NASTRAN BDF Converter v1.0.1',('NASTRAN'),
```

```
    'AnalysisModelType');
#1130= MATERIAL_PROPERTY('1.2',$,#1109);
#1133= DESCRIPTION_ATTRIBUTE('TangentCTE',#1129);
#1134= ELEMENT_MATERIAL('1','Fea Material',(#1116,#1123,#1129));

#1145= ARBITRARY_VOLUME_3D_ELEMENT_COORDINATE_SYSTEM('',#13);
#1147= VOLUME_3D_ELEMENT_DESCRIPTOR(.LINEAR.,
    'LINEAR_HEXAHEDRON.CHEXA',
    (ENUMERATED_VOLUME_ELEMENT_PURPOSE(.STRESS_DISPLACEMENT.)),
    .HEXAHEDRON.);

#1136= VOLUME_3D_ELEMENT_REPRESENTATION('1',(#1145),#1137,
    (#75,#84,#104,#100,#140,#144,#164,#160),#62,#1147,#1134);
#1149= VOLUME_3D_ELEMENT_REPRESENTATION('2',(#1145),#1137,
    (#84,#88,#108,#104,#144,#148,#168,#164),#62,#1147,#1134);
#1152= VOLUME_3D_ELEMENT_REPRESENTATION('3',(#1145),#1137,
    (#88,#92,#112,#108,#148,#152,#172,#168),#62,#1147,#1134);
```

Listing 2.2: STEP file representation

Officially these files are called ISO 10303-21 files, but are in general referred to as *STEP files*. Another term is STEP P21 files, where P21 refers to the part of the ISO 10303 standard which specifies the format.

STEP data on this form are the most common in implementations and usage. Most CAD applications will offer the functionality of exporting and importing data in this form.

Listing 2.3 shows a snippet of the ISO 10303-28 (XML) representation of a single occurrence of an entity. This form (P28) is less popular than the P21 form. It is however used in some cases in the context of PLM where it may represent an assembly of a product, where each assembly component references P21 files containing the geometry of the parts.

```

<Volume_3d_element_representation id="i6758">
        <Name>1</Name>
        <Items exp:cType="set">
                <Arbitrary_volume_3d_element_coordinate_system
                 xs:nil="true" ref="i6765"/>
        </Items>
        <Context_of_items>
                <Geometric_representation_context-
                parametric_representation_context xs:nil="true"
                ref="i6759"/>
        </Context_of_items>
        <Node_list exp:cType="list">
                <Node xs:nil="true" ref="i6217"/>
                <Node xs:nil="true" ref="i6222"/>
                <Node xs:nil="true" ref="i6232"/>
                <Node xs:nil="true" ref="i6230"/>
```

```
                <Node xs:nil="true" ref="i6250"/>
                <Node xs:nil="true" ref="i6252"/>
                <Node xs:nil="true" ref="i6262"/>
                <Node xs:nil="true" ref="i6260"/>
        </Node_list>
        <Model_ref>
                <Fea_model_3d xs:nil="true" ref="i6209"/>
        </Model_ref>
        <Element_descriptor>
                <Volume_3d_element_descriptor xs:nil="true"
                 ref="i6766"/>
        </Element_descriptor>
        <Material>
                <Element_material xs:nil="true" ref="i6757"/>
        </Material>
</Volume_3d_element_representation>
```

Listing 2.3: XML file representation of **volume_3d_element_representation**

Implementing STEP

The standard also specifies how to create APIs interfacing the STEP data model for different programming languages;

- C++, as defined in ISO 10303-23 [27]
- C, as defined in ISO 10303-24 [28]
- Java, as defined in ISO 10303-27 [29]

The language bindings above are all dependent of ISO 10303-22 [30], which defines the *standard data access interface* (SDAI). The SDAI is independent of any programming language and defines an abstract API to manage data models and repositories, and access STEP data in a database. It provides mechanisms for STEP data access regardless of the underlying database format. Some SDAI implementations are discussed in [31, 32, 33].

For applications to work with STEP data, a language interface to the data model is needed. Since EXPRESS is computer readable, APIs may be generated by parsing the data models provided by STEP. The bindings listed above describe how this process can be done for the mentioned languages.

An example of generated C++ code, from Jotne's [34] application EDMS (*EXPRESS Data Manager TM* [35]), from an EXPRESS data model, is shown in listing 2.4. In this code extract, we see the class declaration of the generated class representing **volume_3d_element_representation**. It contains *get* and *put* functions for the different attributes defined in the EXPRESS specification. As the entity has a supertype, the generated class inherits from its parent class. The definition of the functions, not shown in the code listing, uses SDAI functions to access the data in the database implementation of the EXPRESS data model.

```

class volume_3d_element_representation
    :  public element_representation
{
protected:
   volume_3d_element_representation() {}
public:
   static const entityType eType =
        et_volume_3d_element_representation;
   List<node_representation*>*          get_node_list();
   void           put_node_list(List<node_representation*>* v);
   void           unset_node_list() { unsetAttribute(5); }
   bool           exists_node_list() { return isAttrSet(5); }
   SdaiAggr       get_node_list_aggrId();
   void           put_node_list_element(node_representation*);
   fea_model_3d* get_model_ref();
   InstanceId     get_model_ref_id(entityType *etp = NULL);
   void           put_model_ref(fea_model_3d* v);
   void           put_model_ref_id(InstanceId id);
   void           unset_model_ref() { unsetAttribute(6); }
   bool           exists_model_ref() { return isAttrSet(6); }
   volume_3d_element_descriptor* get_element_descriptor();
   InstanceId     get_element_descriptor_id(entityType *etp = NULL);
   void put_element_descriptor(volume_3d_element_descriptor* v);
   void put_element_descriptor_id(InstanceId id);
   void unset_element_descriptor() { unsetAttribute(7); }
   bool exists_element_descriptor() { return isAttrSet(7); }
   element_material* get_material();
   InstanceId     get_material_id(entityType *etp = NULL);
   void put_material(element_material* v);
   void put_material_id(InstanceId id);
   void unset_material() { unsetAttribute(8); }
   bool exists_material() { return isAttrSet(8); }
   void* operator new(size_t sz, Model *m) {
        return m->allocZeroFilled(sz);
    }
   volume_3d_element_representation(
        Model*     m,
        entityType et = et_volume_3d_element_representation) {
            dbInstance::init(m, et, this);
        }
   volume_3d_element_representation(
        Model*     m,
        InstanceId id,
```

```
        entityType et=et_volume_3d_element_representation) {
            dbInstance::init(m, et, this, id);
        }
};
```

Listing 2.4: Generated C++ class declaration for **volume_3d_element_representation**

2.2.2 Integrated Generic and Application Resources (IGR/IAR)

The *Integrated Generic* and *Integrated Application Resources* (IGR and IAR), are where the "core" of the STEP data models lies. The documents in these categories define the core of all the data model entities and types (in EXPRESS) in STEP. Each document handles different topics; some *generic*, and some *application* specific (as the category names suggested). For example, in the code in Listing 2.1, the representation of the **representation** entity, a generic entity which is inherited by a large amount of other entities, comes from an *Integrated Generic Resource* document. The **element_representation** and **volume_3d_element_representation**, which are *application* specific (FEM), comes from a document in the *Integrated Application Resources*.

Some examples of such documents are:

- *Integrated Generic Resources*:
  - ISO 10303-41 - Fundamentals of product description and support
  - ISO 10303-42 - Geometric and topological representation
  - ISO 10303-50 - Mathematical constructs
- *Integrated Application Resources*:
  - ISO 10303-101 - Draugthing
  - ISO 10303-104 - Finite element analysis
  - ISO 10303-110 - Computational fluid dynamics data

Even if entities are specified in each of the different documents, they can reference each other (for example by entities inheriting from entities in other documents), with the exception that data models in the IGRs, may not reference from the IARs.

2.2.3 Application Modules (AM)

The architecture of STEP is layered and modular; data models are divided in *modules* that can be reused by other modules depending on which *layer* they belong to. The lowest layer contains the data models in the IGRs (generic), above this layer are the IARs (application), and on top of these are the *Application Modules* (AM).

AMs can "collect" (by referencing) different parts of the IARs, IGRs and other AMs, thus modularizing data model content. This layering and modularizing, results in a tree structure of modules of AM, IAR, and IGR data models, where the IGRs are leaf nodes. There are multiple top nodes in this tree structure, these are all AMs, and more specifically, *Protocol* AMs.

As seen in figure 2.2, AMs are further divided in *Foundation*, *Implementation*, and *Protocol* categories. Where foundation AMs are the lowest layer, and provide data model content that is highly reusable, and implementation AMs are more specific to a certain usage. Protocol AMs are essentially the same as implementation AMs, however, as "top nodes", they define the content in *Application Protocol*s, where their name comes from.

2.2.4 Application Protocols (AP)

The highest level of the STEP architecture are the *Application Protocols*.

The Application Protocols are the complete data models defined by the aggregated content from a specific protocol AM. Each of them are tailored for a specific engineering application domain, and are the only semantically complete data models.

When an application, tool, or system wants to support STEP for a certain domain, it does so by implementing the data model specified in an AP. The AP can then be used to specify a database dictionary, API, or exchange format for the application. The developers implementing this support, need not to know the whole architecture of STEP (IGR, IAR, AM, etc) but only understand the content of the AP and its language EXPRESS and implementation methods. It is STEP experts, together with domain experts, that design the data models in each STEP module, with high focus on reusing existing modules, for a specific application.

Examples of APs are:

- AP203 Configuration controlled 3D design of mechanical parts and assemblies [36]
  - AP203 was the first application protocol and was intended for CAD data. It was, and is still implemented in many major CAD applications to exchange STEP files (P21) by import and export functionality. It supports 2D/3D geometry, assemblies, annotations, modeling history and much more.
- AP214 Core data for automotive mechanical design processes [37]
  - AP214 came later, also implemented as much as AP203, and supports everything in AP203, but adds support for representing configuration control, tolerance data, kinematics, and more.

- AP242 Managed Model Based 3D Engineering [38]
  - AP242 is the newest application protocol for CAD. It extends the content of AP203 and AP214 with PDM, tessellation (for visualization puroposes), composites representation, and more. This AP is implemented in the newest versions of major CAD tools.
- AP209 Multidisciplinary Analysis and Design
  - AP209 supports simulation information; mainly FEM, but also CFD. It also contains the whole content of AP242, and can therefore represent CAD, simulation, and relations between them.
- AP239 Product Life Cycle Support (PLCS) [39]
  - AP239 [40] is intended for PLM/PDM systems and support topics such as logistics, risk management, tasks, planing, etc.
- AP238 Integrated CNC Machining (STEP-NC) [41]
  - AP238 is first of all a replacement for G and M code for CNC manufacturing. It also supports CAD geometry representations and can relate this to the machining operations.

A lot of content in the above APs overlap, and because this is done by the APs sharing the same modules, interoperability between them is achieved.

2.3 ISO 10303 AP209 - Multidisciplinary Design and Analysis

2.3.1 AP209 Scope

AP209 is the core data model that is used for this study. As mentioned in the previous section, it supports CAD, FEA, CFD, as well as PLM and other CAE related information. All these domains are relevant in the context of digital twins. Some previous studies presenting the use of AP209 are presented in; [42, 43, 44, 45] related to exchange of analysis and composite data, [15, 46] for translation of analysis data, [47, 48] related to electromagnetism and thermal data exchange.

This part of the STEP standard can be used as a format for file exchange between different simulation solvers, and as database schema for SDM applications. It has however not gained the same support by software vendors as AP203, AP214, and AP242 has for the CAD domain.

The first edition of the standard was released in 2001, and was later updated to edition 2 in 2014. A "bug fix" of the current edition is ongoing, and will be released as edition 3

[49]. Jotne is also currently involved in the planing of an international project which will outline the content of a 4th edition. This will focus on nonlinear FEA support, and will take into account the recommendations outlined as a result of this thesis.

The scope of the standard includes, among many other items, the following;

- CAD data (geometry, product information, assemblies of parts)
- FEM load cases; including linear static and dynamic analysis
- FEM loads
- FEM boundary conditions
- FEM mesh (multiple element types, including generic element definition)
- FEM material properties
- Topological relations between FEM mesh and loads, and geometry
- Composite material including its FEM representation

2.3.2 High Level Entities

In STEP, every data representation can be traced back to a **product**, a **product_definition_formation**, and a **product_definition** entity of which they belong. Be it an analysis, a geometric shape, a sensor, or a material type. These entities, together with other related entities, are often referred to as *high level* entities. They hold product information, such as meta data and PLM data.

This section will not present all the details of PLM data representations, but introduce the basic entities that define a *product*.

Product descriptions in STEP, including the above entities, are defined in Parts 41 [50] and 44 [51] of the standard. These documents are focused on what defines a product in terms of its constituents. A product has an identification, categorizations, relations with other products, relation to a specific life cycle stage or discipline view, and may have multiple versions. With the use of those type of entities, this information can be represented semantically.

The formal definitions of the mentioned *product* entities are very generic. According to the documentations, the mentioned *product* entities, and their context and categorization entities, are defined as follows:

- **product**:

  - (...) a representation of a product or a type of product.
  - (...) depends on one or more instances of **product_context** specifying a frame of reference that determines the validity of the information held about the product or class of products.

- **product_definition_formation**:
  - (...) a collector of definitions of a product.

- **product_definition**:
  - (...) a representation of an aspect of a product, or of a class of products, for an identified life cycle stage.
  - (...) may represent particular products that are the members of an identified class of products.
  - (...) acts as an aggregator for information about the properties of products.

- **application_context**
  - (...) is the identification of an application protocol.
  - (...) represents various types of information that relate to product data and may affect the meaning and usage of that data.

- **product_context**
  - (...) is a type of **application_context_element** that represents life cycle independent information about a product. This information describes the discipline in which data about the product are created.

- **product_definition_context**
  - (...) is a type of **application_context_element** that represents information about the stage in the product life cycle for which a **product_definition** is created or used.

- **product_category**
  - (...) is a classification that applies to products.

In many implementations, for example in CAD tools, when a STEP model is exported, *product* related entities will not hold much more information than the product's name and ID, which very often are the same. For assemblies of parts, each part, and each assembly representation will have a **product**, **product_definition_formation**, and a **product_definition**. Also, when exporting from a CAD tool, context and categorization entities will usually hold default values or non-informative data. A CAD tool doesn't necessarily *care* of PLM information. STEP models in such context are more of a method for transfering design data between different applications. However, in a PLM/SDM environment these entities may be populated with information that gives more meaning. In such environment, relation entities may be used to relate different models together, and this is done with product level (high level) entities.

2.3.3 Low Level Entities

In STEP, by low level entities, we mean all the entities describing the details making up the complete data set. Figure 2.3 shows how some of the main content in a sample AP209 file is related. In blue are the "high level" entities representing the analysis as a *product*. The *product* has a shape, which is a FEM model, with a *structural response property* that relates to the details of the FEA model (in green). Further down, in yellow, are the load cases and their associated results. Only the top level entities of the results are shown in the figure for simplicity. Each load case has a tree-structure-like breakdown collecting all the loads and boundary conditions used in the load case. These are composed of *states* and *state_relationships*. In the figure, the breakdown of two load cases are shown. In pink, what relates to loads, and in cyan what relates to boundary conditions. In this specific example, the same boundary conditions are used in both load cases, while the loads have both independent loads and shared loads.

When using AP209 for structural testing, as presented in Paper 1, direct references between the instances representing structural test specific data (tests, test results, and sensors) and the FEA specific data discussed above, are possible. Such tight coupling is required for being able to correctly represent digital twin related data, where physical results, and simulation results, need to be strongly connected.

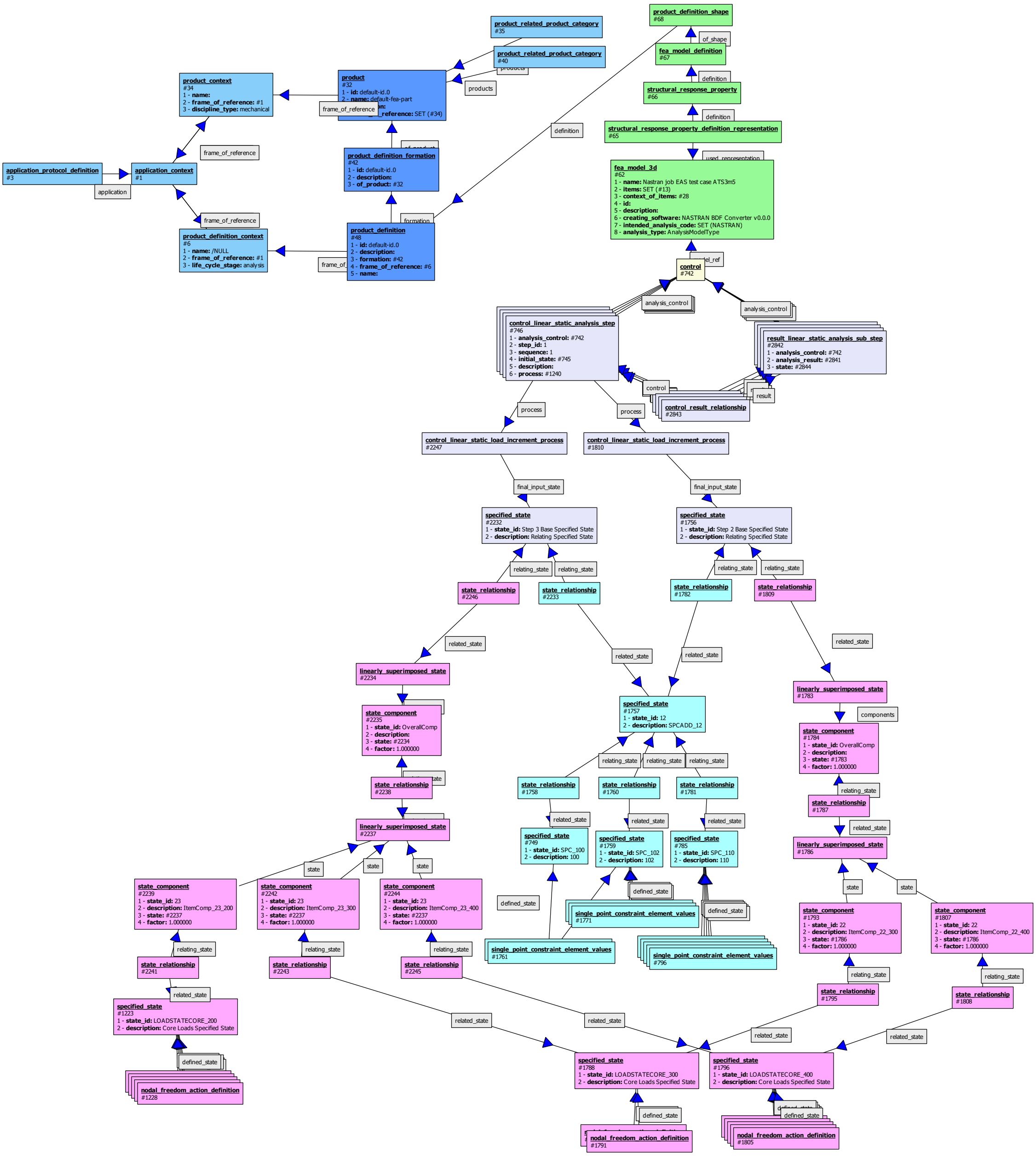


Figure 2.3: Detailed overview of a STEP AP209 population

Chapter 3

Application development

3.1 STEP Converters

To integrate FEM and structural test data in STEP compliant systems (in our case, a SDM system) we need to translate the data from their original source to STEP. A major task of this study was to develop such converters and embed them in a SDM application.

This thesis discusses STEP in the context of data management, and as a format for long term storage. However, converters, especially for FEM, are useful regardless of this context. Being able to quickly convert analysis files from one solver format to another, can save a lot of time for the engineering analyst. In our case, such converters are created by having AP209 as a central format, which every file translation converts to or from. Instead of having direct conversion functionality between each format, every format need only one to- and one from-conversion implementation (Figure 3.1).

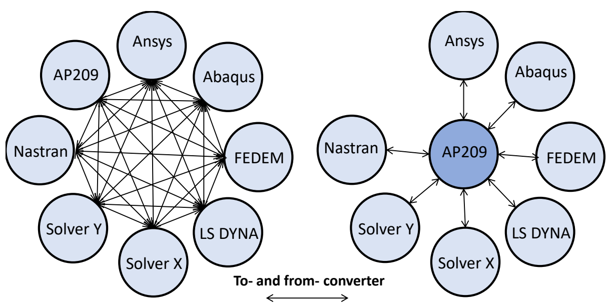


Figure 3.1: AP209 as a central format for converters

This section is dedicated to highlight some of the details and technical aspects involved in the converter development.

3.1.1 FEM Converter

The developed FEM converter works by taking in a source file, a source format, a target file, a target format, and some optional additional configuration parameters. The accepted formats are currently NASTRAN (bulk data format .nas/.bdf/.dat), ABAQUS (input file format, .inp), ANSYS (APDL file format), and AP209, with the following implemented conversions:

NASTRAN	$\Longleftrightarrow$	AP209
ABAQUS	$\Longrightarrow$	AP209
ANSYS	$\Longleftarrow$	AP209

If none of the target and source formats are AP209, the converter converts first to AP209, then to the target format (if that conversion is implemented).

The converter is a command line application created with C++. To work with AP209 data, it uses EDMS and a generated C++ AP209 interface as discussed in 2.2.1. When writing and reading AP209, the converter has direct access to any part of the model. For the native formats however, which are in ASCII, the data needs to be processed sequentially (line by line). Performance-wise, it would be beneficial to work with the solvers binary formats using an API and have direct access to the required parts of the model. Because these formats are proprietary and not open, with no available API, this was not done.

Different strategies may be implemented for converting sequential files. The current method used, is that the application first reads the complete file, store the information internally in memory as custom data structures, then converts the data to AP209.

The main drawback with this is that it sets a limit on the size of model to be converted, based on the available memory. However, such implementation is faster to develop, which has been a crucial benefit during this study.

Alternative implementations, that could be developed, are:

1. As the sequential model is read, convert data directly if possible. If not possible, store the data in memory until the missing information is read, then convert.

2. Same as above, but instead of temporarily storing in memory, store in a database.

3. Read the complete model first, and store everything in a database. This could be a STEP database with a custom AP schema. In that case, an API for the model could be generated and used for the conversion process.

3.1.2 Structural Test Converter

The structural test converter reads input files specifying sensor and test information, as well as result files holding sensor measurement data, and creates AP209 data in a STEP database model. An existing FEM analysis (in AP209) is also input to the converter, such that the created structural test data may relate to the relevant FEM data.

Paper 1 presents how the resulting model from this converter is structured.

The formats of the input files used for sensor and test definition are in ASCII, and based on a format provided by Lockheed Martin Aeronautics in connection with the CRYSTAL project which is discussed in 3.3.2. The files may be written manually, or when integrating the converter in an application, may be generated.

The converter was divided in three different modes:

- **Sensor definition mode:** Creates sensors as product data structures in a specified AP209 database model. Based on the data in the sensor input files, sensors are referencing finite elements on the input FEM mesh, and their orientation, position, type, identification, and other relevant information are added as properties.

  The sensors also reference the STEP representation of the physical product that the sensors are placed on.

- **Test definition mode:** Creates the representation of an executed or planned structural test in the same model.

  Based on the data in the test input files, each test gets linked to their corresponding load case that simulates them in a FEM model. They also reference the specific sensors that are used in each tests, and the physical product which the tests are applied on.

- **Test result mode:** Reads tests results from files generated from DAQ (Data acquisition) systems, such as CATMAN [52], and an input file specifying how each set of result data relates to a specific sensor. The results are collected in STEP data structures and related to their sensors and test cases.

3.2 STEP Explorer

3.2.1 Background

STEP is a complex data model covering a huge amount of concepts. In this study a lot of work was spent exploring, understanding, and extending the standard with new concepts, including generating STEP files with developed converters. This type of work involves a lot of debugging and inspection of generated STEP files, which could be done by reading STEP files in a text editor, or viewing its content in Jotne's database manager, EDMS. This proved to be time consuming for complex files, especially when attempting to explore deeply nested tree-like data structures. Because of this, the study devoted a certain amount time on developing an application, STEP Explorer, for doing such tasks more efficiently.

3.2.2 Overview

The main purpose of STEP Explorer is to give an easy way of exploring the content of STEP files and models. This is achieved by providing a 2D graphical view of STEP instances with their relations.

STEP Explorer (see Figure 3.3) requires that a STEP schema is specified, then a STEP file can be imported. The application initially gives an overview of the different entities (Figure 3.4) that exist in the STEP file. For each different entity, a tabular view can be displayed showing every instance of that type and its attribute values (Figure 3.6).

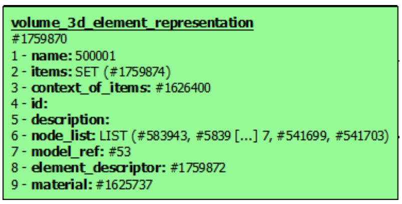


Figure 3.2: Example of instance box

By selecting instances in the tabular instance overview, or by searching for an entity name or instance ID (Figure 3.5), instances can be displayed in the graphical 2D view.
In the viewer, displayed instances are represented by boxes, containing their entity name, instance ID, and attribute values. When attributes are references to other instances, the referenced instance ID is displayed. An example of such box can be seen in Figure 3.2.

The displayed instance boxes have a context menu with the following main available functionalities:

- **Get all attributes:** every attribute of the selected instances which are instance references are displayed as additional boxes.

- **Get attribute:** gives the choice to select any of the attributes that are instance references and display it as an additional box.

- **Get references:** A query is done to find all instances that reference the selected instances. If this results in many results, they are displayed in a table form and a selection can be done of the desired instances, which are then displayed as additional boxes. If there are few results, additional instance boxes are directly created for each.

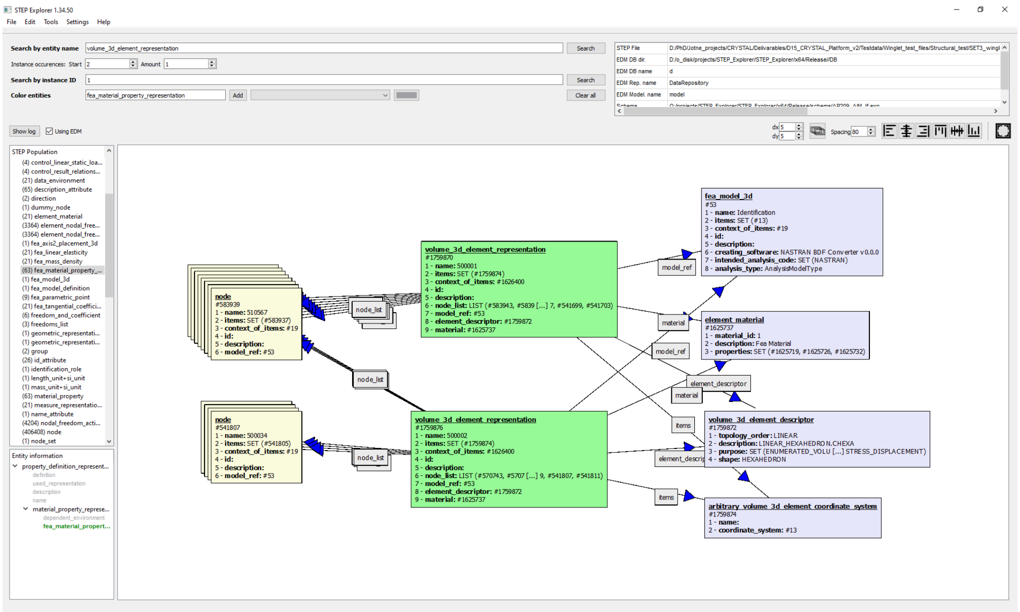


Figure 3.3: STEP Explorer interface

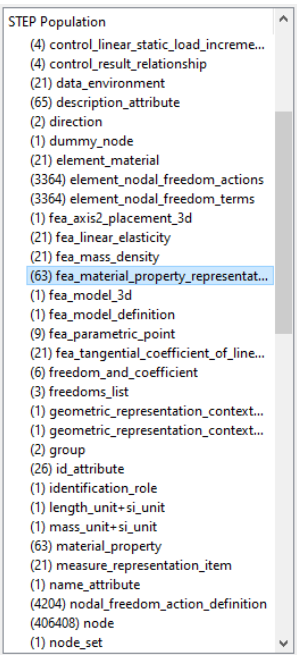


Figure 3.4: Entity list showing which entities exist in the STEP model.

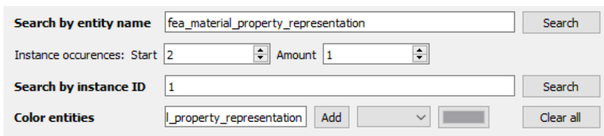


Figure 3.5: Search instance by entity name or instance ID

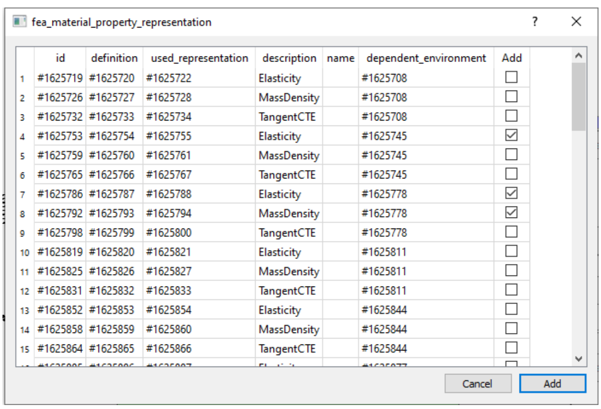


Figure 3.6: Instance list, showing all occurrences of an entity in the STEP model.

When an attribute or reference is displayed through one of the above functions, an arrow is displayed showing the relationship between them. This functionality makes it possible to create diagrams that show complex STEP data structures in a simplistic way. Instance boxes can be moved, aligned, centered, colored, and more, to allow the user to create detailed diagrams as seen in Figure 3.7. These may be exported to PDF or image files to be shared with others.

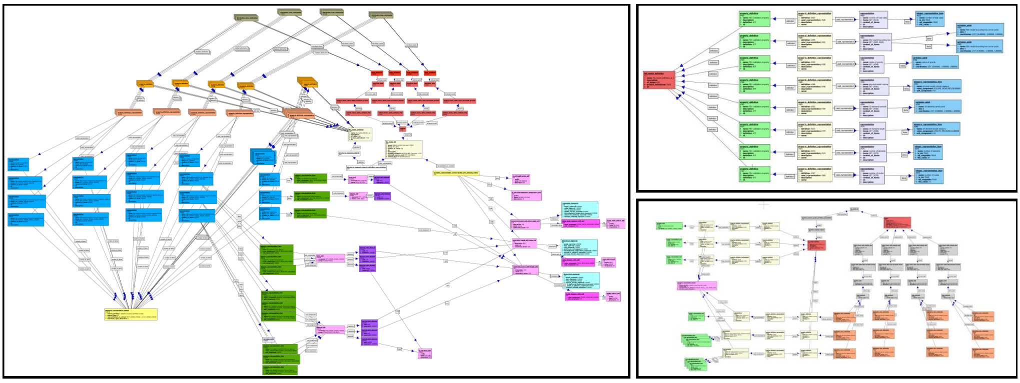


Figure 3.7: Example of generated diagrams

An additional functionality are scripts. A script, using a simple syntax, can be written to define small template-like instructions of which instances, and how instances, should be displayed in the viewer. These template scripts can be used on multiple STEP files to quickly generate standard diagrams that can be compared.

3.3 STEP in Simulation Data Management

As already mentioned, STEP, and more precisely STEP AP209, is a data model that can be used in a simulation data management system. The data model provides multiple PLM data structures for general data management, and for low level details of simulations.

In this study, the research has been heavily related to the use of, and the extension of Jotne's SDM application; EDMopenSimDM™ [53]. This has been used to validate the use of AP209 for managing design, testing, and simulation, as stated in objective **O.1**.

As part of this validation, a use case was introduced, involving design, FEA, manufacturing, testing, and data management of an aircraft winglet, Figure 3.8. The design of the generic winglet was provided by Lockheed Martin Aeronautics, it was simulated with NX Nastran and manufactured and tested at NTNU. EDMopenSimDM with the extensions provided by this study and the CRYSTAL project discussed in 3.3.2, and the converters discussed in 3.1, was used to manage all data.

The winglet use-case was designed to involve the data relevant for a digital twin. Though it didn't take into account any real-time connection between the physical and virtual twin, it satisfies the coverage of the data domains of the thesis' objective. A real life complex industrial digital twin, will in most cases involve some method of feedback between an analysis and sensor measurements on a product in operation. The data transfer mechanism for such a system, can be implemented in a multitude of ways, and is out of scope of this study. In our use-case, the winglet, rather than being in operation, is tested in a lab and the data is collected and imported to the SDM application, to be integrated with the virtual digital twin data; the design and FEM analysis.

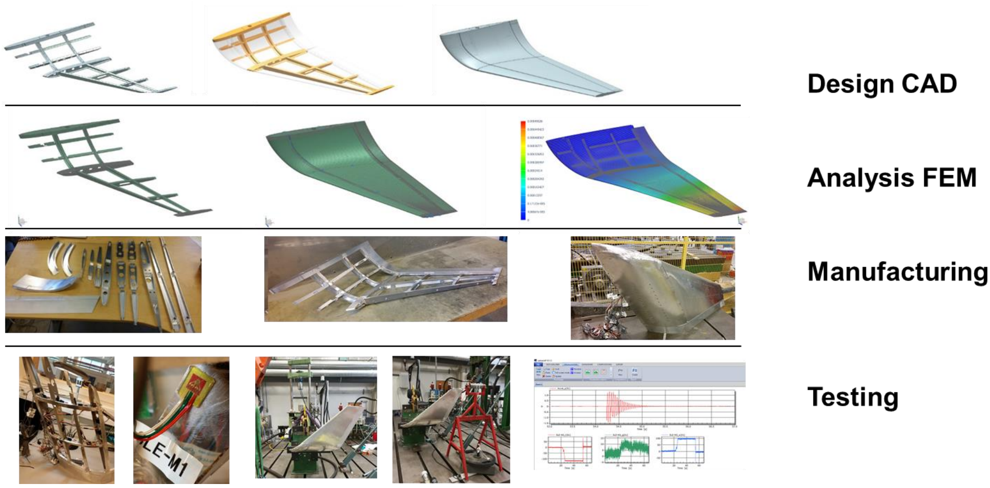


Figure 3.8: Winglet use case

3.3.1 EDMopenSimDM

EDMopenSimDM$^{\text{TM}}$ addresses simulation data management and engineering data archival and retention. The tool is composed of a client and a server application, which enables for collaboration between people and teams within the same, or across different companies.

The main concept of EDMopenSimDM is that it is based on the AP209 data model, which is used as an underlying database dictionary.

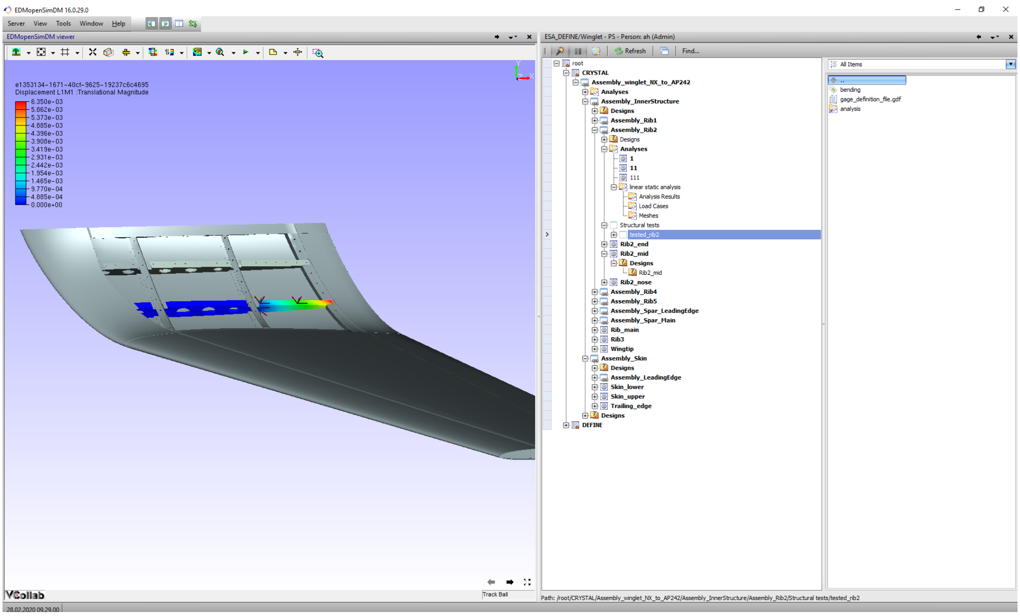


Figure 3.9: EDMopenSimDM interface

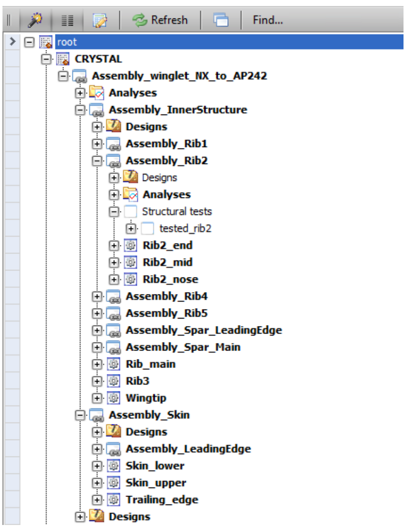


Figure 3.10: EDMopenSimDM product structure

All management concepts are stored as AP209 data structures, which includes; people and organization with access restrictions, tasks and tasks methods, approvals, versioning, etc. The application accepts files in any formats, but has special processing methods for STEP files.

If a STEP file containing a CAD assembly is imported, the assembly structure may be used to define a product structure in a project. Product structures are presented as seen in Figure 3.10. In this particular example the product breakdown structure is a result of importing the CAD assembly of the winglet exported from NX [54].

EDMopenSimDM has integrated the concept of *federated model*. A federated model implies that every model (STEP models) imported is treated as sub-models of the overall federated model of a project. Relations between sub-models are managed through a special *link* model (using the STEP AP209 schema). Each model, may be of different domains, including; CAD, FEA, and structural testing. Section 2.3.2 presented high level entities (**product**, **product_definition**, etc), and how they hold product management information. Relations between sub-models are first of all done on these entities. In addition, for specific cases, low level relations are also created. This includes for example the representation of a test case in a structural test model, and a load case in a FEA model, or a sensor representation and a FEA element. These types of relations are done with the converter presented in 3.1.2. In theory, more detailed relations could be done, such as relations between FEA nodes, elements, loads, and their related CAD geometric constituents such as surfaces and edges. This is however not implemented as it would require integrated CAD and FEA kernels.

Having all this data and information in the SDM application, in one data model, facilitates the integration of data processing tools that requires cross-domain information. An example of such a tool, which further validates objective **O.1**, is the *FEM-Test Correlation tool* (Figure 3.11). This application was developed during this study, and embedded into the SDM application. It accesses the information of a selected structural test and related FEM model, and allows the user to select sensors and test cases. These are related to test result values and FEM results. For each sensor, their test measurements and results from analysis can be plotted individually or overlapped, enabling them to be compared and checked for correlation. Additional information for each sensor is also available, such as orientation, position, type, etc.

Another example of service developed, which takes the advantage of tightly coupled FEM and sensor data, is the visualization of sensors on its related FEM mesh (Figure 5).

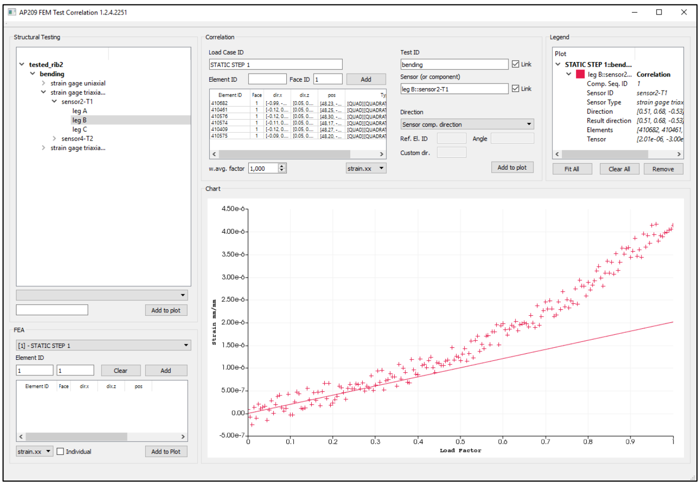


Figure 3.11: FEM-Test Correlation tool interface

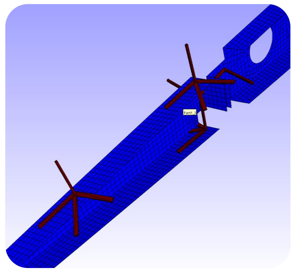


Figure 3.12: Sensors visualized on FEM mesh

3.3.2 CRYSTAL Project

The study of this thesis started while Jotne had a project named CRYSTAL with Lockheed Martin Aeronautics Company. Part of the objectives of this thesis were also requirements in the scope of CRYSTAL. The project's goal was to explore and implement a system for a central storage of design, simulation, and structural test data. This system was EDMopenSimDM, and many of the features presented in 3.3.1, were developed during the project using results from this thesis, especially Paper 1. Paper 4 presents the vision and scope of CRYSTAL.

3.3.3 DEFINE Project

The DEFINE project is an ongoing ESA (European Space Agency) project done together with Jotne. Its main objective is to increase the integration of 3D digital models in order to improve the efficiency and effectiveness of the assembly, integration, and test procedures and documentation, fully integrated into the overall space system lifecycle. It is similar to the CRYSTAL project described in section 3.3.2, however, in addition to FEA and structural test data, it introduces additional domains, including; thermo and thermo-elastic analysis, optical raytracing analysis, cloud points data, testing data from thermal scan, vibration testing, load cells, and others.

Results from this thesis will be applied in this context to further improve the functionality and domain coverage of EDMopenSimDM.

Chapter 4

Summary of Papers

Paper 1 together with the implementations in SDM (section 3.3), is the main contribution to objective **O.1**. This paper highlights the details around utilizing AP209 for structural testing.

Paper 2, contributing to **O.2**, identifies some of the obstacles for AP209 to gain more acceptance from the industry. The main concern is the lack of support of nonlinear FEM, which is the main topic addressed in this paper.

Paper 3 contributes to both **O.1** and **O.2**. The main concern from Paper 2 is addressed here, by showing how AP209 can be enhanced to cover nonlinear FEM. Nonlinear FEM is required for both digital twin data, which can involve complex dynamic simulations, and for general FEM data management that might involve complex analyses.

An abstract and summary of each paper are presented in the next sections.

4.1 Paper 1 - Relating Structural Test and FEA data with STEP AP209

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. The proposed method takes advantage of data structures and elements defined in STEP AP209 Edition 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.

This paper focuses on **O.1** by investigating how to handle structural test data with the STEP standard. It presents the technical aspect of how this was done, and answers the question; how can STEP AP209, which is intended for multidisciplinary design and analysis, cover structural testing?

The reason for using AP209 for this domain, is because not only should the standard represent the structural test data, but it should also represent the *relations* to data from FEM, as well as other domains. AP209 already covers generic data representations such as PLM constructs, mathematical concepts, general engineering information, generic properties, and more. These representations can all be related to FEA specific data representations. The paper presents how these generic representations can be used to represent test data and relate those to analysis data.

The study focused on how to handle the following information:

- how to represent sensors with focus on strain gauges.
- how to represent sensor components (for example in biaxial/triaxial gauges), and how to relate these to parent sensor assemblies, and to test results in a specific test.
- how sensors relate to a position and elements in the FEM model.
- how sensors relate to a specific test representation and a physically tested product.
- how the test cases relate to load cases in the FEM model.
- how to handle sensor specific properties, such as gauge factor, sensor type and model, and DAQ (Data Aquisition) connection IDs.

The converter for structural test data, discussed in section 3.1.2, was developped during the study of this paper, first as a stand alone application, then integrated in EDMopenSimDM.

4.2 Paper 2 - ISO 10303 AP209 - Why and how to embed nonlinear FEA

Abstract

ISO 10303 STEP AP209 edition 2 is a data model standard intended for data exchange and storage of simulation information. The standard has a wide coverage of FEA (Finite Element Analysis) information, but is missing certain features such as nonlinear FEA. This paper gives an introduction to the STEP AP209 standard and presents projects in which AP209 has been implemented.

The study then identifies requirements that should be supported by a standard FEA data model, but are not fully covered by AP209. Each requirements are discussed in the context of how they are supported by existing major solver applications. Without giving detailed solutions for how these should be implemented in AP209, starting points for further research is suggested.

For AP209 to be more widely used it needs to gain interest from more engineering application vendors, especially FEM solver vendors. This study presents STEP and AP209, and discusses projects and use cases where it has been implemented and used. It further tries to identify why it has not gained as much traction as the STEP application protocols for CAD (AP203, AP214, and AP242). One of the main issues identified is the lack of support for certain important FEA concepts. How major FEA solvers support these missing FEA concepts, and how these should be implemented in AP209, is investigated. No implementations or detailed solutions are discussed, but further work is suggested. Paper 3 and Annex C continue this study in more details.

4.3 Paper 3 - Extending STEP AP209 for Nonlinear Finite Element Analysis

Abstract

ISO 10303 STEP AP209 is a standard for exchanging and storing simulation information along side related PLM (Product Lifecyle Management), CAD (Computer Aided Design), and other CAE (Computer Aided Engineering) data. The AP209 standard, despite being well documented and covering a wide range of engineering information, has not been widely implemented by FEA (Finite Element Analysis) solver or SDM (Simulation Data Management) applications. This is assumed to mainly be due to AP209 not yet supporting nonlinear FEA.

*The following study takes basis in the findings of **Paper 3**, where improvements of the AP209 standard were suggested. Some of these suggestions, related to nonlinear FEA, are here further investigated and implemented, and proposed for further standardization.*

Analysis test cases using these new features are created, and converters between different FEA formats are developed.

The test cases are nonlinear, static and dynamic, with different defined time step control parameters and loading conditions. The FEA data converters translates data between AP209 and the solver specific formats. The complete data information from the analyses are preserved during the conversion, and the generated analyses are solved. To confirm that no information was lost during the process, simulation results are investigated and compared.

Paper 3 continues from the findings in Paper 2. The study shows how AP209 can be extended to support multiple types of analysis, and not only linear FEA. This is done by extending the entity hierarchy of the analysis type entities, and introducing the representation of FEA solver parameters to the standard. In addition, as it is often needed in nonlinear analyses, a recommended representation of varying load (by time or space) is described.

The improvements to the standard are implemented in a custom AP209 schema, which is implemented in the FEA converter discussed in 3.1.1. With this converter, different test cases created in Abaqus are converted to Ansys and Nastran formats via a translation to AP209. These tests are solved in the respective solvers, and to confirm that AP209 kept all the necessary information during the conversions, and the results are compared.

4.4 Summary of STEP AP209 Extensions

From Paper 1

The following summarizes the STEP usage recommendations from Paper 1 related to how to represent structural test data and its relation to FEM.

Below, the term *product structure* is referring to an instance structure of a set of **product**, **product_definition_formation**, and **product_definition** instances.

- Representation of a physically tested part
  - With relation to:
    - Design model
    - Analysis
    - Physical tests
- Representation of a sensor type
  - Represented by a *product structure*
  - With sensor type specific properties and metadata such as; manufacturer, model name, description, type identifier, angles for multi-axial strain gauges, etc.
- Representation of a sensor
  - Represented by a *product structure*
  - Assembly of multiple sensor components (for example different measurement directions in a multi-axial strain gauge)

  - With sensor specific properties and metadata such as; ID, name, geometrical location, mesh location (elements and nodes)

- Representation of a sensor component
  - Represented by a *product structure*
  - With sensor component specific properties and metadata such as; measurement direction
  - With relations to:
    - Measurement results
    - Test cases

- Representation of a physical test case
  - With relations to:
    - The physically tested part
    - Measurement results from sensor components
    - Sensors used in the test
    - Load case in analysis

- Representation of results of specific test
  - With properties and metadata such as; DAQ specific properties, sampling rate, gauge factor, channel ID, etc.
  - With relations to:
    - Sensor component
    - Test case

- Representation of collection of physical test results
  - Represented by a *product structure*
  - With relations to all tests performed on a physically tested part and the analyses simulating those tests

From Paper 3

The following summarizes the AP209 extensions and recommendations from Paper 3:

- Analysis types
  - Extensions of **control_analysis_step**, **result_analysis_step** and **control_process**

    - New entities for each type of analysis
    - Hierarchically structured

- Analysis parameters
  - Extension of **property_definition** and **property_definition_representation**
    - With entities **fea_parameter_property** and **fea_parameter_property_definition_representation**
    - Used to relate FEA parameters to specific load case or complete analysis
  - Added new type ***control_or_control_process***
    - Used by **fea_parameter_property** and **fea_parameter_property_definition_representation** to be able to either relate to a specific load case or a complete analysis
  - Recommendation of standardized FEA parameters to be used by **fea_parameter_property** and **fea_parameter_property_definition_representation** for common FEA properties, structured by categories
- Varying loads
  - Representation of loads varying by time, load factor, coordinates, or other variables

From Appendix C

- Mesh regions
  - Extension of entity **element_group**
    - Extended with element type specific groups that can hold an element aspect (face ID and edge ID)
    - Multiple groups can be related to form regions defined by sub-regions with different element aspects
- Mesh interactions (contact and glue)
  - Extension of entity **state_definition**
    - Extended with entities for specific mesh regions, and interaction types such as contact and glue
    - Relates to interaction specific properties represented by the entities **property_definition** and **property_definition_representation**
- Nonlinear materials

- Representation of elastic perfectly plastic material model defined by a yield point
- Represenstation of multilinear plasticity material model defined by a set of stress and plastic strain values

Chapter 5

Conclusion and Future Work

The STEP standard covers a large scope of industrial data. However, it is only in the CAD domain that it has been widely accepted and implemented. Today thousands of STEP CAD files are produced and archived daily world-wide [55], it has proven to simplify CAD file exchange between different systems. A similar wide implementation in other domains would greatly simplify data management and archiving in engineering industry.

With the main goal of improving the effectiveness of data management of physical test and simulation data, in the context of digital twins, PDM, PLM and SDM, this study has investigated, proposed recommendations, and applied the use of STEP for structural testing and FEM.

For the structural testing domain we showed how test data can be stored without making changes to the standard, but by defining how to structure the data in a semantical correct way. It was illustrated how to implement these recommendations in a SDM application, which can then manage both structural test and FEA data in the same format. The application, which also stores design data in STEP, is able to relate models and information together, ensuring interoperability and traceability.

The need for nonlinear FEM to be supported by AP209 was identified, and after making recommended extensions to the standard, implementations were performed to show their applicability.

While this thesis recommends many additions to the standard and makes STEP even more applicable for use in digital twins, for these recommendations to become available for industry, they need to be officially endorsed by the ISO community and published in official ISO standards and recommended practices. Furthermore, the domains of structural testing, and nonlinear FEM are still larger than what was covered in this study. This study covers the basics, but future work would be required, to apply these results in a wider set of use cases including additional types of sensors and more complex simulations.

Bibliography

[1] David Jones, Chris Snider, Aydin Nassehi, Jason Yon, and Ben Hicks. Characterising the digital twin: A systematic literature review. *CIRP Journal of Manufacturing Science and Technology*, 2020.

[2] F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee. Digital twin in industry: State-of-the-art. *IEEE Transactions on Industrial Informatics*, 15(4):2405–2415, April 2019.

[3] Thomas H.-J. Uhlemann, Christian Lehmann, and Rolf Steinhilper. The digital twin: Realizing the cyber-physical production system for industry 4.0. *Procedia CIRP*, 61:335 – 340, 2017. The 24th CIRP Conference on Life Cycle Engineering.

[4] Qinglin Qi, Fei Tao, Tianliang Hu, Nabil Anwer, Ang Liu, Yongli Wei, Lihui Wang, and A.Y.C. Nee. Enabling technologies and tools for digital twin. *Journal of Manufacturing Systems*, 2019.

[5] Wim Gielingh. An assessment of the current state of product data technologies. *Computer-Aided Design*, 40(7):750 – 759, 2008. Current State and Future of Product Data Technologies (PDT).

[6] Merja Peltokoski, Mika Lohtander, and Juha Varis. The role of product data management (pdm) in engineering design and the key differences between pdm and product lifecycle management (plm). 04 2014.

[7] NAFEMS Simulation Data Management Working Group. *What is simulation data management?* NAFEMS, 2014.

[8] Andrea Buda, Petri Makkonen, Ronan Derroisne, and Vincent Cheutet. Pdm suitability study for cae data management. 07 2011.

[9] ISO 10303-1:1994. Industrial automation systems and integration – Product data representation and exchange – Part 1: Overview and fundamental principles. Standard, International Organization for Standardization, Geneva (Switzerland), 1994.

[10] P. Brun, F. Fernández González, M. Lehne, A. McClelland, and H. Nowacki. A neutral product database for large multifunctional systems. In R. Vio and W. Van Puymbroeck, editors, *Computer Integrated Manufacturing*, pages 87–97, London, 1991. Springer London.

[11] Soonhung Han, Young Choi, Sangbong Yoo, and Namkyu Park. Collaborative engineering design based on an intelligent step database. *Concurrent Engineering: R&A*, 10:239–249, 09 2002.

[12] Jeongsam Yang, Soonhung Han, Matthias Grau, and Duhwan Mun. Openpdm-based product data exchange among heterogeneous pdm systems in a distributed environment. *The International Journal of Advanced Manufacturing Technology*, 40(9):1033–1043, February 2009.

[13] Han M. Shih. Migrating product structure bill of materials excel files to step pdm implementation. *International Journal of Information Management*, 34(4):489 – 516, 2014.

[14] D. Iliescu, I. Ciocan, and I. Mateias. Assisted management of product data: A pdm application proposal. In *2014 18th International Conference on System Theory, Control and Computing (ICSTCC)*, pages 128–133, Oct 2014.

[15] Sébastien Charles and Benoit Eynard. Integration of cad and fea data in a pdm environment: Specification of a step simulation data management schema. *17th IMACS World Congress*, 02 2005.

[16] Guillaume Ducellier, Sébastien Charles, Benoit Eynard, and Emmanuel Caillaud. Traceability of simulation data in a plm environment: Proposition of a step-based system that support parameter integration. *9th International Design Conference, DESIGN 2006*, 01 2006.

[17] ISO 10303-209:2014. Industrial automation systems and integration – Product data representation and exchange – Part 209: Application protocol: Multidisciplinary analysis and design. Standard, International Organization for Standardization, Geneva (Switzerland), 2014.

[18] R. Lanza, J. Haenisch, K. Bengtsson, and T. Rlvåg. Relating structural test and fea data with step ap209. *Advances in Engineering Software*, 127:96 – 105, 2019.

[19] R. Lanza, J. Haenisch, K. Bengtsson, and T. Rlvåg. ISO 10303 AP209 - Why and how to embed nonlinear FEA. Under review, 2020.

[20] R. Lanza, J. Haenisch, K. Bengtsson, and T. Rlvåg. Extending STEP AP209 for Nonlinear Finite Element Analysis. Under review, 2020.

[21] J. Haenish, K. Bengtsson, R. Lanza, and O. Liestl. Open Simulation Data Management and Testing, The Crystal Project. *NAFEMS World Congress 2017*, page 161, 2017.

[22] ISO 10303-11:2004. Industrial automation systems and integration – Product data representation and exchange – Part 11: Description methods: The EXPRESS language reference manual. Standard, International Organization for Standardization, Geneva (Switzerland), 2004.

[23] 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. Standard, International Organization for Standardization, Geneva (Switzerland), 2016.

[24] The HDF Group. https://www.hdfgroup.org/solutions/hdf5/. Accessed on 29 March 2020.

[25] ISO 10303-26:2011. Industrial automation systems and integration – Product data representation and exchange – Part 26: Implementation methods: Binary representation of EXPRESS-driven data. Standard, International Organization for Standardization, Geneva (Switzerland), 2011.

[26] ISO 10303-28:2007. Industrial automation systems and integration – Product data representation and exchange – Part 28: Implementation methods: XML representations of EXPRESS schemas and data, using XML schemas. Technical report, International Organization for Standardization, 2007.

[27] 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. Technical report, International Organization for Standardization, 2000.

[28] ISO 10303-24:2001. Industrial automation systems and integration – Product data representation and exchange – Part 24: Implementation methods: C language binding of standard data access interface. Technical report, International Organization for Standardization, 2001.

[29] ISO 10303-27:2000. Industrial automation systems and integration – Product data representation and exchange – Part 27: Implementation methods: Java TM programming language binding to the standard data access interface with Internet/Intranet extensions. Technical report, International Organization for Standardization, 2000.

[30] ISO 10303-22:1998. Industrial automation systems and integration – Product data representation and exchange – Part 22: Implementation methods: Standard data access interface. Standard, International Organization for Standardization, Geneva (Switzerland), 1998.

[31] Richard M. Botting and Anthony N. Godwin. Analysis of the step standard data access interface using formal methods. *Computer Standards & Interfaces*, 17(5):437 – 455, 1995. Formal Description Techniques.

[32] A. Goh, S.C. Hui, and B. Song. An integrated environment for product development using step/express. *Computers in Industry*, 31(3):305 – 313, 1996. Product and process data modelling.

[33] S Ma, Y Maréchal, and J.-L Coulomb. Methodology for an implementation of the step standard: a java prototype. *Advances in Engineering Software*, 32(1):15 – 19, 2001.

[34] Jotne EPM Technology AS. http://www.jotneit.no. Accessed on 14 February 2020.

[35] EXPRESS Data Manager™ Software Development Kits. http://www.jotneit.no/products/express-data-manager-edm, 2019. Accessed on 14 February 2020.

[36] 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. Standard, International Organization for Standardization, Geneva (Switzerland), 2011.

[37] 10303-214:2010. Industrial automation systems and integration – Product data representation and exchange – Part 214: Application protocol: Core data for automotive mechanical design processes. Standard, International Organization for Standardization, Geneva (Switzerland), 2010.

[38] ISO 10303-242:2014. Industrial automation systems and integration – Product data representation and exchange – Part 242: Application protocol: Managed model-based 3D engineering. Standard, International Organization for Standardization, Geneva (Switzerland), 2014.

[39] ISO 10303-239:2012. Industrial automation systems and integration – Product data representation and exchange – Part 239: Application protocol: Product life cycle support. Technical report, International Organization for Standardization, 2012.

[40] PDES, Inc, AFNeT. ISO 10303 (STEP) AP 239 edition 3 Application Protocol For Product Life Cycle Support (PLCS). Technical report, 2015.

[41] ISO 10303-238:2012. Industrial automation systems and integration – Product data representation and exchange – Part 238: Application protocol: Product life cycle support. Standard, International Organization for Standardization, Geneva (Switzerland), 2014.

[42] Keith A. Hunten. Cad/fea integration with step ap209 technology and implementation. *MSC Aerospace Users Conference Proceedings*, 1997.

[43] N. Pitre K.A. Hunten, J.W. Klintworth and T.E. Mack. New standards based data exchange bridge for design (cad), analysis (cae) and manufacturing (cam) of composite structures. In *MSC 1999 Aerospace Users Conferance Proceedings*, 1999.

[44] Edward L. Stanton, Takman Mack, Hiren D. Patel, and Jamie Klintworth. Composite beam models using iso step ap209. 2003.

[45] Keith A. Hunten, Allison Barnard Feeney, and Vijay Srinivasan. Recent advances in sharing standardized step composite structure design and manufacturing information. *Computer-Aided Design*, 45(10):1215 – 1221, 2013.

[46] Peter Bartholomew and Christian Paleczny. Standardization of the finite element analysis data-exchange in aeronautics concurrent engineering. *Journal of Computing and Information Science in Engineering - JCISE*, 5, 03 2005.

[47] Eric Lebègue, Georg Siebes, and Charles Stroom. Thermal analysis data exchange between esa and nasa with step. 07 1999.

[48] Ma Singva. Step standard's evaluation for modeling in electromagnetism. *COMPEL - The international journal for computation and mathematics in electrical and electronic engineering*, 18(3):311–323, March 2020.

[49] ISO/AWI 10303-209 Industrial automation systems and integration — Product data representation and exchange — Part 209: Application protocol: Multidisciplinary analysis and design - edition 3. https://www.iso.org/standard/75060.html, 2019. Accessed on 02 March 2020.

[50] ISO 10303-41:2000. Industrial automation systems and integration – Product data representation and exchange – Part 41: Integrated generic resource: Fundamentals

of product description and support. Standard, International Organization for Standardization, Geneva (Switzerland), 2000.

[51] ISO 10303-44:2019. Industrial automation systems and integration – Product data representation and exchange – Part 44: Integrated generic resource: Product structure configuration. Technical report, International Organization for Standardization, 2019.

[52] HBM. CATMAN DAQ Software. https://www.hbm.com/en/2290/catman-data-acquisition-software/. Accessed on 2March 2020.

[53] EDMopenSimDM (version 12.0). http://www.jotneit.no/edmopensimdm, 2019. Accessed on 14 February 2020.

[54] NX (version 11.0). https://www.plm.automation.siemens.com/global/en/products/nx/, 2016.

[55] C. Jackson and D. Prawel. The 2013 State of 3D Collaboration and Interoperability Report (page 17). https://www.plm.automation.siemens.com/en_us/Images/Lifecycle-Insights-2013-Collaboration-Interoperability_tcm1023-210162.pdf, 2013. Accessed on 14 February 2020.

[56] A.W.A. Konter. *Advances Finite Element Contact Benchmarks*. NAFEMS, 2006.

[57] NAFEMS. *Introduction to Non-Linear Finite Element Analysis*. NAFEMS, 2000.

Appendices

Appendix A

Main publications

A.1 Paper 1 - Relating Structural Test and FEA data with STEP AP209

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

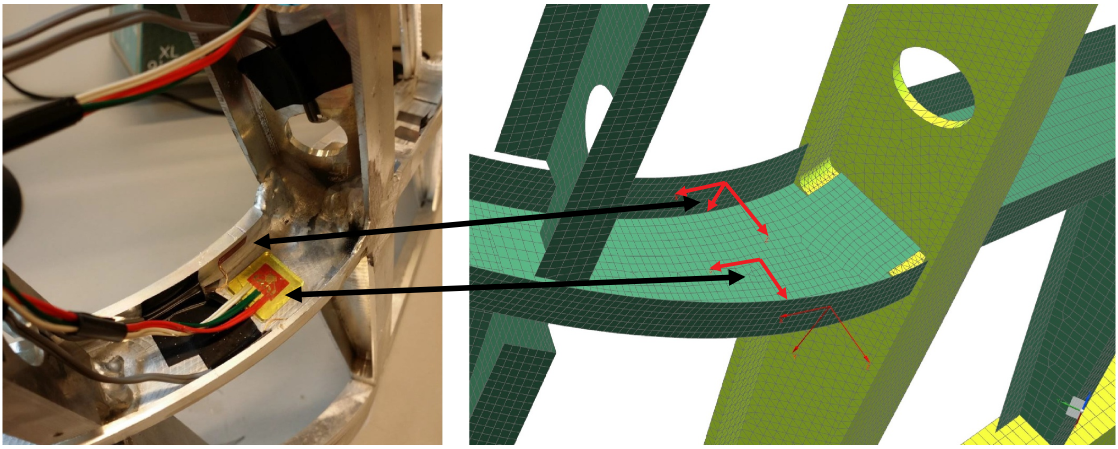


Fig. 1. Mapping sensor locations and orientations to FEM model.

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 life-span 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

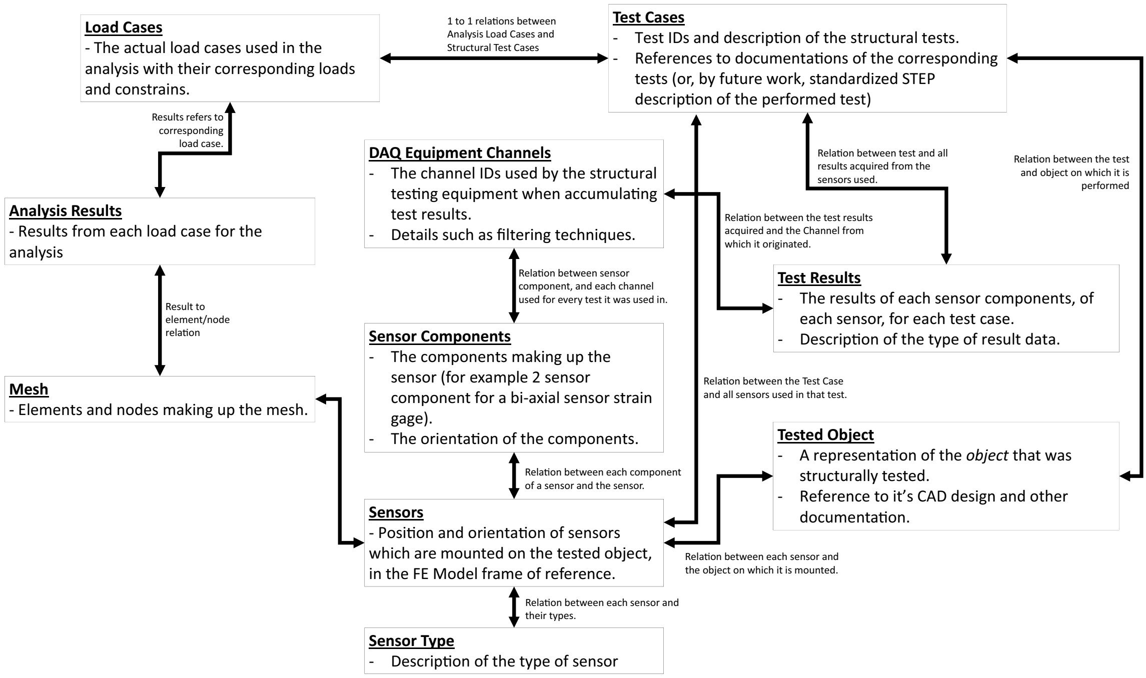


Fig. 2. Overview of main data represented in an AP209 model containing Analysis and Structural Test data, and how they relate.

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.

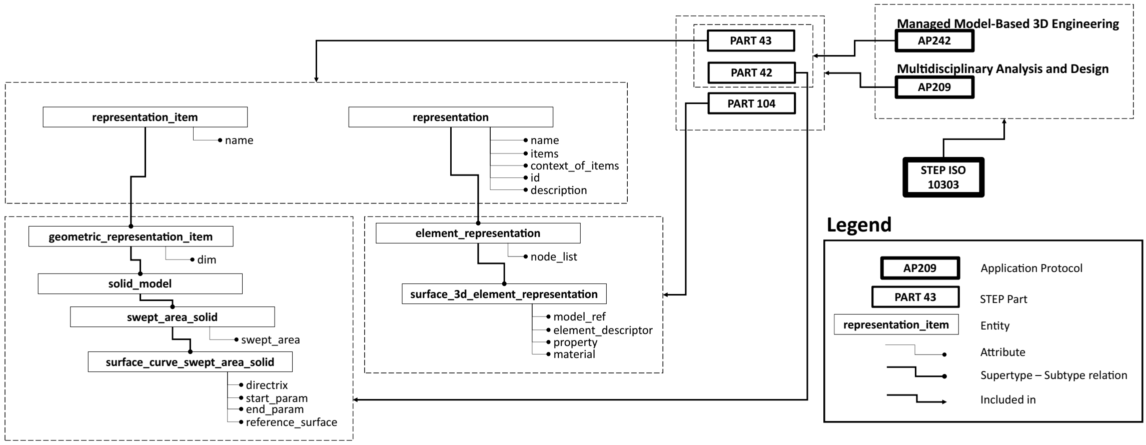


Fig. 3. Example of how entities are included in Parts which again are included in Application Protocols.

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.)

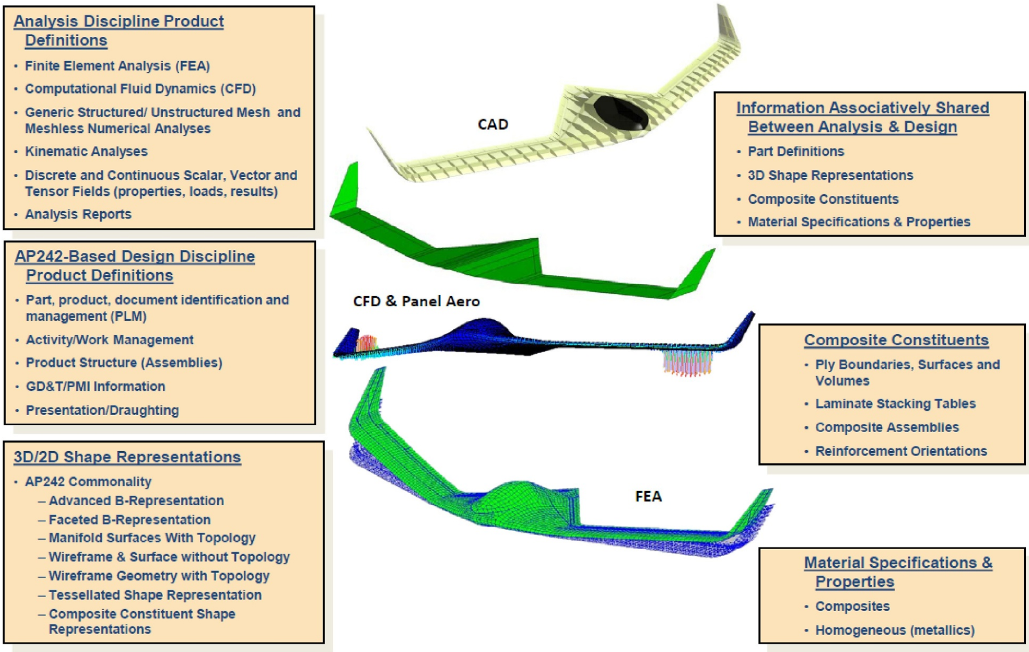


Fig. 5. Data which is supported by AP209 [17].

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');
(...)
```

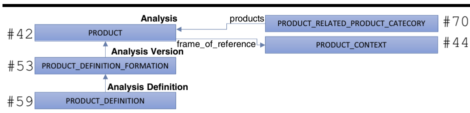


Fig. 6. Left: Extract of a STEP P21 file. Right: Graphical representation used in this paper.

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

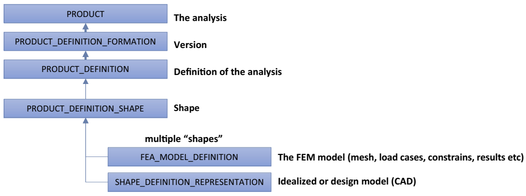


Fig. 7. High level entities in the Analysis Model.

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

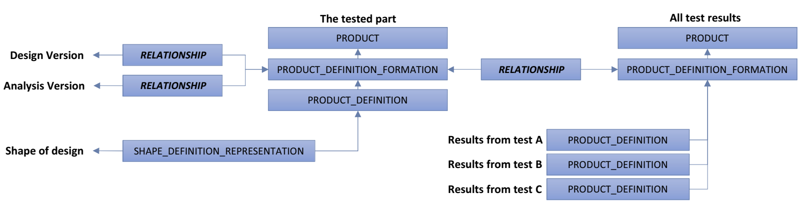


Fig. 8. Relations between FEA, Design, Tested Part and Test Results. In this case, there are three individual tests.

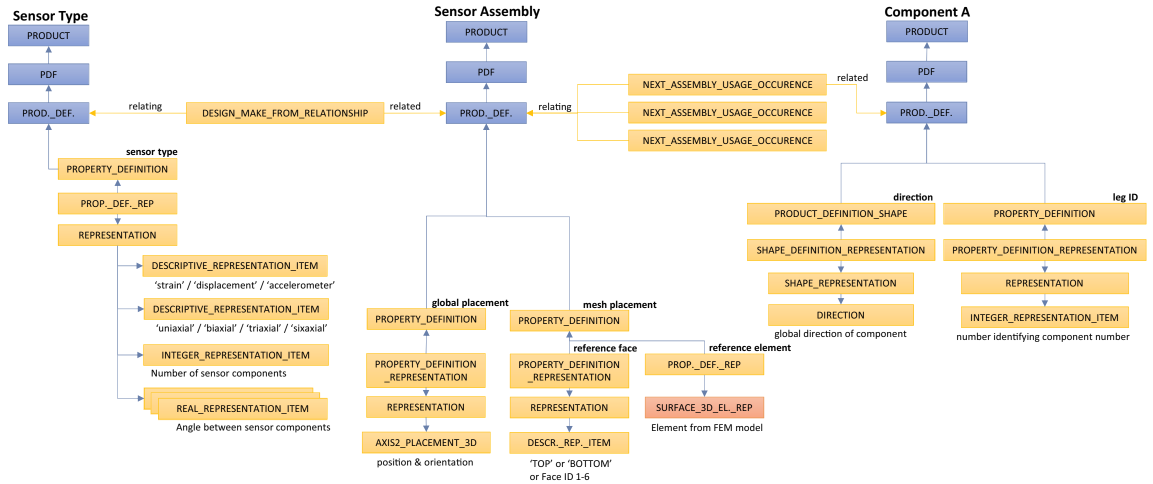


Fig. 9. Example of data structure for sensor with three sensor components (only one is shown).

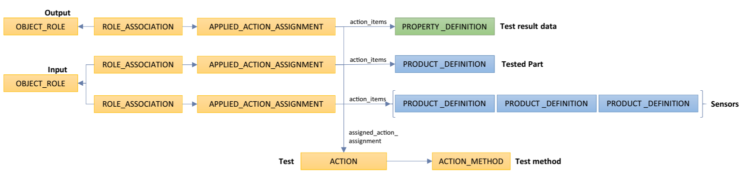


Fig. 10. Data structure for a test performed on a part with three sensors, resulting in a certain test result.

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:

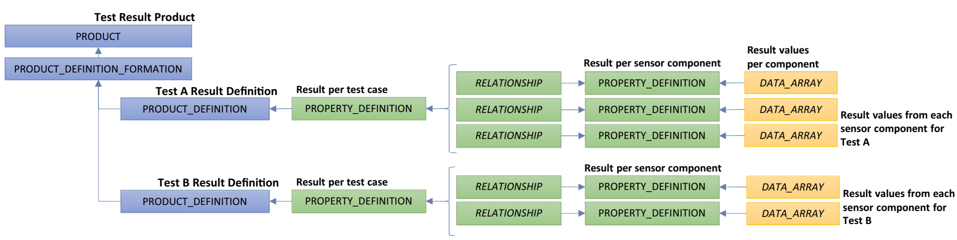


Fig. 11. Example of data structure showing the relation between the sensor result values and the output result data **product**.

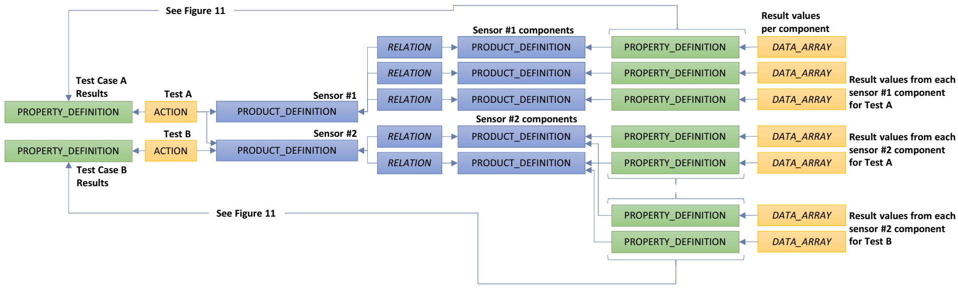


Fig. 12. Data structure showing relation between sensor results, sensors and tests. Here we have two tests and 2 sensors. One of the sensors (sensor 2) is used in both tests.

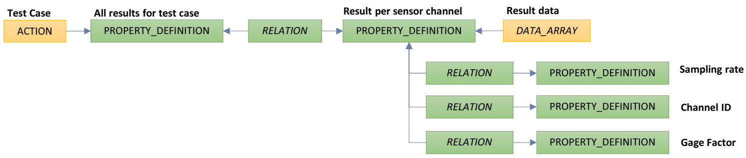


Fig. 13. The data values array references the result property of sensor a component, which is also referenced by properties that are unique for this sensor channel and test case.

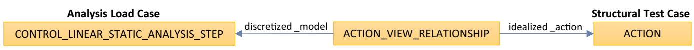


Fig. 14. Load cases from the Analysis data set are related to the Test Case of the Structural Test data set via the action_-view_relationship.

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

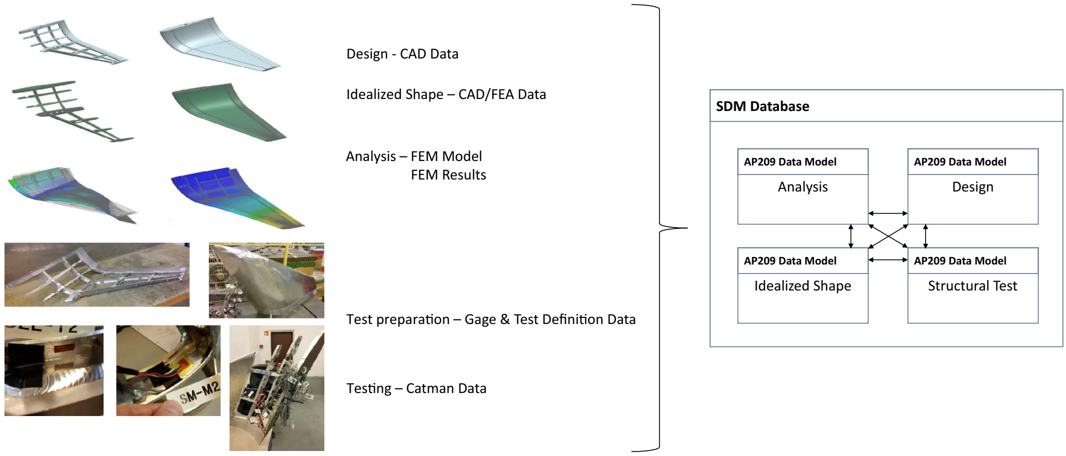


Fig. 15. Different AP209 data sets imported to the SDM tool.

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: engingeering 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.

A.2 Paper 2 - ISO 10303 AP209 - Why and how to embed nonlinear FEA

Contents

Vitae 2

Abstract 3

1 Introduction 3

2 The STEP ISO 10303 Standard 5
2.1 How STEP enables interoperability of engineering data 5
2.2 Analysis of the industrial relevance of AP209 8

3 State of the art of existing AP209 implementations 10
3.1 Early implementations 10
3.2 CAx-IF and LOTAR 10
3.3 TERRIFIC 11
3.4 Cloudflow 11
3.5 VELaSSCo 11
3.6 CAxMan 11
3.7 CRYSTAL 12
3.8 Arrowhead Tools 12

4 Recommended FEA extensions to AP209 12
4.1 FEA requirements for AP209 12
4.2 Recommended extensions 15

5 Conclusion and future work 17

Appendix A Solver analysis parameters 25

Appendix B Contact parameters 31

Remi K.S. Lanza
Jotne EPM Technology AS & Norwegian University of Science and Technology

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 PhD, which is a collaboration project between the Norwegian University of Science and Technology (NTNU) and Jotne. While pursuing his PhD, 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.

Terje Rølvåg
Norwegian University of Science and Technology

Prof. Rølvåg was born in Mo I Rana, 16/10-1963. Rølvåg holds a MSc. 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.

Jochen Haenisch
Jotne EPM Technology AS

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^{TM}. 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.

Kjell A. Bengtsson
Jotne EPM Technology AS

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.

ISO 10303 AP209 - Why and how to embed nonlinear FEA

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

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

Abstract

ISO 10303 STEP AP209 edition 2 [1, 2] is a data model standard intended for data exchange and storage of simulation information. The standard has a wide coverage of FEA (Finite Element Analysis) information, but is missing certain features such as nonlinear FEA. This paper gives an introduction to the STEP AP209 standard and presents projects in which AP209 has been implemented.

The study then identifies requirements that should be supported by a standard FEA data model, but are not fully covered by AP209. Each requirements are discussed in the context of how they are supported by existing major solver applications. Without giving detailed solutions for how these should be implemented in AP209, starting points for further research is suggested.

Keywords: STEP ISO 10303, FEM Analysis, Nonlinear FEA, Data Exchange, Simulation Data Management

1. Introduction

Nowadays all aspects of life involve information that is stored digitally as data. In the industry or privately, data is stored as files on hard-drives either locally, on servers, or dispersed in cloud systems. The average person may be interested in *where* their data is stored, but not necessarily *how* it is stored. We rely on the availability of applications to be able to open our files and to interpret their formats to view or edit the information. The reason we can use different software from different providers for these tasks on the same files, are the defined file formats. File format definitions are either open, that is, publicly available, or proprietary. Anyone may create applications to access the content of files in open formats; the details of proprietary formats are only known to a few and are kept confidential for business reasons. From a user's point of view, open formats are more attractive as they usually give a wider selection of applications to choose from and, thus, more user control over the data.

For instance; image files can be stored in formats well defined by standards such as JPEG[3], PNG[4], BMP[5] etc., and are therefore understood by many

*Corresponding author. Tel.: +47 452 04 992
Email addresses: `remi.lanza@jotne.com` (R.Lanza), `jochen.haenisch@jotne.com` (J.Haenisch), `terje.rolvag@ntnu.no` (T.Rølvåg)

applications. The same applies to music, video and text documents. In industrial domains, we can find open formats for 2D and 3D models such as, DXF[6], OBJ[7], STL[8], X3D[9] etc.

The more complicated and rich the data is, the more advanced becomes the data format.

Engineers depend on tools for many different and advanced domains, such as, CAD (Computer Aided Design), FEA (Finite Element Method), CFD (Computational Fluid Dynamics), photometric simulation, control engineering, electronic circuit design, etc. For each domain, the engineer may choose from multiple tools from different providers. Companies will usually select the application that best supports their work-flows.

Some of these application may only support proprietary formats, some may use standard formats, and others may support both. However, the standard format will usually only cover a subset of the application information scope.

Depending on the selected applications, exchanging engineering data across different parties within the same engineering domain or among different domains often leads to unnecessary additional work. When cooperation between different engineering teams requires information to be transferred between two non-interoperable applications, conversions or redundant input of data are necessary.

In the context of CAD data, many CAD software vendors have implemented standard formats for import and export. However, CAD applications may offer data types and user operations that only exist within that tool. These special features may lead to limitations on how big parts of the application data model can be shared by a standard data model. Anyway, a standard that covers the majority of the data would still greatly simplify CAD file exchanges.

Multiple open formats support CAD data exchange. Most of these are limited to a certain subset of geometric definitions. [10] and [11] summarizes and compares some of the most broadly adopted 3D model formats. The most widely implemented and used non-proprietary exchange format for CAD [12], is ISO 10303 [1] (commonly known as STEP). It includes a wide variety of geometric and topological definitions and links those to PDM (Product Data Management) information, other engineering domains and product lifetime data in general.

The benefit for CAD users, from vendors implementing such a standard, is that it enables them to share models across multiple CAD tools. Standard formats such as STEP are also backwards compatible with newer versions. This is not always the case for the proprietary formats, which may modify their format with new releases, unabling the opening of files from previous versions.

Despite having been a crucial part of product development for many decades, FEA applications rarely offer standard exchange formats.

Data is exchanged between different solvers, but often only mesh data is well implemented in export and import. Analysis information, such as load case definitions, loads, boundary conditions and additional analysis specific data, often need to be exchanged manually or through custom routines. Some solvers will accept NASTRAN and Abaqus input file formats to import and export such analysis information, in addition to mesh data, but often with limited scope.

A widely implemented FEA standard will give the same benefit the CAD domain already has; to allow engineers to share between, and work across, FEA solvers from different vendors. Equivalently significant, is the ability it gives to archive FEA information to be retrieved in the future, regardless of the originating application releasing new versions.

The mentioned STEP standard does have considerable support for FEA through one of its Application Protocols known as AP209 [2].

Other attempts for defining a standard format for simulation data are FEMML [13, 14, 15], SysML [16] and PAM [17].

The purpose of this study is to give an overview of STEP AP209 and its capabilities, as well as identifying some missing domain coverage of FEA. Focus is given to nonlinear FEA, and without going in details, initial suggestions are given for how potential extensions to the standard could be done.

The paper is organized as follows; section 2 gives an overview of the STEP standard and AP209. Section 3 presents completed and ongoing use cases and projects where AP209 has been applied. In section 4 concepts that AP209 does not cover are identified and discussed with respect to how they are supported in major FEA solvers. Finally section 5 concludes the study and suggests future work.

2. The STEP ISO 10303 Standard

2.1. How STEP enables interoperability of engineering data

As discussed in section 1, standard formats simplify reuse of data by providing portability that enables data file exchange and database sharing. The goal of STEP is to offer to the public a consistent suite of data definitions for all major engineering domains. Being consistent, means that interoperability is not only possible within the same domain, but also between overlapping domains. For example, the subsets of STEP that are known as AP203[18], AP214[19] and AP242[20] cover among others, CAD and PDM data. Part of the definitions used in these subsets, such as the definitions of geometric surfaces, are also relevant in the FEA domain. AP209, a superset of AP242, has, in addition to all the content of AP242, support for FEM and other simulation types. PDM information, which links data of all domains into consistent product descriptions, is part of all subsets of STEP, that is, Application Protocols.

The structure of the STEP standard is relatively complex; a short introduction is therefore included here. Other suggested resources to get a better understanding are; *STEP in a Nutshell* [21], chapter 2 of *Relating structural test and FEA data with STEP AP209* [22] and *STEP Application Handbook* [23].

The standard is managed by ISO [24] as the ISO 10303 series and is divided in a set of several hundreds documents. The documents are organized into categories, such as; *Description methods*, *Implementation methods*, *Integrated application* and *Integrated generic resources* (IR), *Application modules* (AM) and *Application protocols* (AP). Figure 1 shows this classification.

The main concept of the standard is that it defines a data model; this consists of a set of *entity* data types and other supporting data types. An entity is essentially the same as a class in an Object-Oriented language; it can inherit from other entities and hold attributes. Each attribute is of either an entity data type or any other data type. An example of an entity is **surface_3d_element_representation**, which defines a FEM 3D surface element by referencing nodes. It inherits from the entity **element_representation** and adds surface specific attributes, such as, element properties and material. The EXPRESS data model for these entities are seen in Listing 1. The details of data

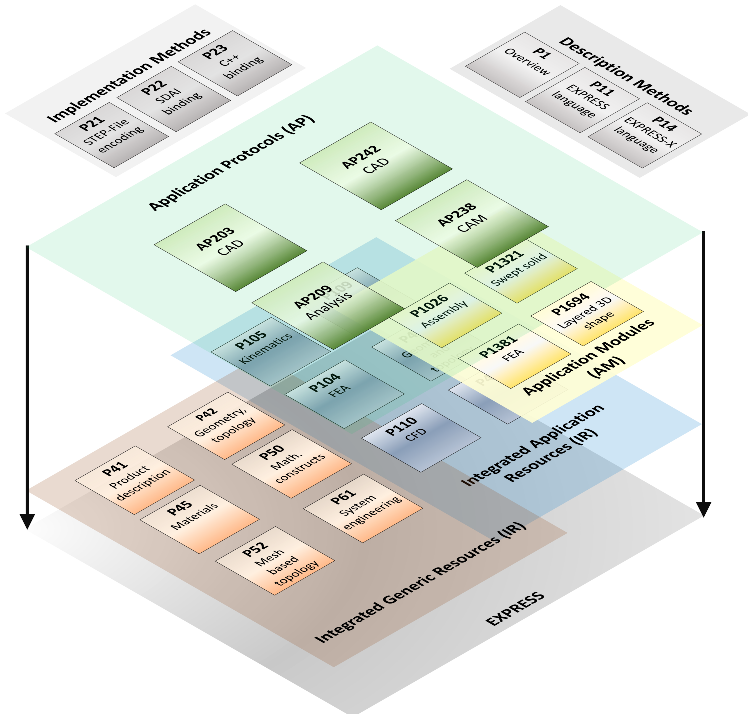


Figure 1: Overview of STEP ISO-10303 documents. Each layer shows examples of documents. Documents in a layer may reference documents in lower layers.

model constraints in so called "where"-rules (WR) and functions (FU) are left out for simplicity.

```
ENTITY element_representation
    SUPERTYPE OF (ONEOF(left out for simplicity))
    SUBTYPE OF (representation);
        node_list : LIST OF node_representation;
WHERE
    WR1: (left out for simplicity)
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, name;
    WHERE
        WR1: (left out for simplicity);
        WR2: (left out for simplicity);
```

```
        WR3: (left out for simplicity);
        FU1: (left out for simplicity);
END_ENTITY;
```

Listing 1: EXPRESS model of **element_representation** and **surface_3d_element_representation**

The *generic* and *application* IR documents are the foundation of the STEP product model and hold also the formal definitions of the entities and data types. An important part of the STEP standard is that these definitions are written in a formal lexical and graphical data modeling language, *EXPRESS*, which is itself defined by the standard in the *Description methods* documents as ISO 10303-11[25]. Definitions written in EXPRESS, are computer readable (as well as human readable), and may be processed by software applications. The AM documents reference and may add semantics to the content from the IR documents. These are used by APs that are built by the modular approach (AP239[26], AP242) vice the monolithic approach (AP209, AP235[27], AP238[28]). For a monolithic AP, the information model is specified as an integral part of the AP document. For a modular AP, the information model is specified by reference to a top level AP module that references a tree structure of AMs.

An application that supports STEP does this by reference to a specific Application Protocol (AP). For example, most CAD software supports one or multiple of the APs; AP203, AP214 and AP242. Each AP is described in its own document in which it defines a data schema. The AP reference content from the generic IR, application IR and AM documents. An AP, thus, groups, specializes, and adds to content from the STEP resources for a specific engineering domain and/or product life-cycle stage.

From a developer's point of view, if an application is to support a certain AP, the AP schema is the core specification from which the service is developed. As the schema is written in EXPRESS, APIs and frameworks may be generated by the developer, or already existing third-party applications may be resused.

The process of creating interfaces, is also, to some extent, standardized by STEP. The standard specifies a generic interface (SDAI; Standard Data Acess Interface) to access STEP data stored in a database systems that uses EXPRESS schemas as basis for their database dictionaries. For certain languages (C, C++, Java), the standard also specifies how to generate an interface layer on top of the SDAI interface, specifically for allowing applications to work with STEP databases. This greatly simplifies the implementation of the standard. Some implementations and analyses of STEP interfaces are presented in [29], [30], and [31].

PDM and SDM applications, which may cover multiple engineering domains, can implement multiple APs. An overview and discussion on the STEP standard in the context of PDM is presented in [32]. Multiple implementations of PDM/-PLM (Product Lifecycle Management) systems using STEP as a database backbone and for data exchange are outlined in [33, 34, 35, 36, 37]. A SDM/PDM implementations using an extended AP209 schema is presented in [38, 39].

As all APs are based on the same low level details of product structure, properties, units, etc., the application may create direct references between models of the different APs. For example, a SDM application may accept both CAD and FEA STEP models and hold relations between them, such as, an applied FEA force on a CAD edge or a set of finite elements on a geometric surface.

Applications with such functionality currently only exist using proprietary formats. The data may be stored and exchanged as an ASCII STEP file [40] or as a binary database [41] based on the schema of the AP.

The most common APs used in the industry are AP203, AP214 and AP242, whereas AP203 and AP214 are by now deprecated, and are replaced and upward compatible with AP242.

These are supported by most CAD applications and by some PDM/PLM applications.

2.2. Analysis of the industrial relevance of AP209

The Application Protocol AP209 (ISO 10303-209) has the tittle *Multidisciplinary Analysis and Design*. The newest version of AP209 is called AP209 edition 2 (or AP209e2), and AP209 edition 3 is currently being developed. In this paper, AP209e2 will be referenced as just AP209. The purpose of this part of the STEP standard is to serve as:

1. A file format to share data between simulation solvers.
2. A database schema for PDM and SDM applications to integrate, share, and archive simulation data, independent of any proprietary format.

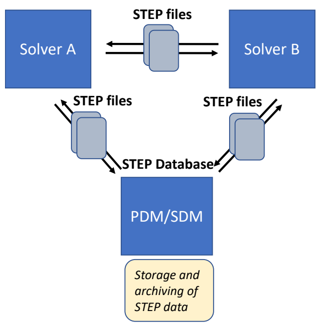


Figure 2: AP209 for file exchange and database integration and storage.

The AP209 standard has not yet been widely implemented by FEA tool vendors, but several trial implementations by different vendors and organizations exists. Some of these implementations are described in section 3.

One of the major benefits of a universal data exchange format for FEA (and other domains) is the reduction in number of converters required for an application. As shown in Figure 3 the number of converters required for an engineering process expands more than linear when the number of involved applications with proprietary formats grows. Without a central format, the number of two-way converters for a single application is calculated by: $\sum_{n=1}^{N_f-1} n$ where N_f is the number of formats. With a central format, the number of converters for all involved applications is equal to the number of applications.

There are many possible reasons for why AP209 or other FEA standard formats have not been widely implemented, some of which are:

1. Vendors want to keep their customers
    - Naturally, vendors want their customers to use their software as much and as long as possible. Rather than focusing on interoperability with

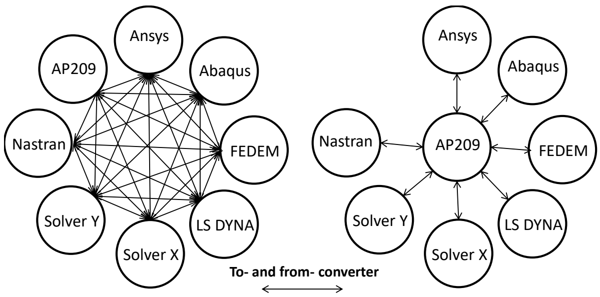


Figure 3: AP209 as a central format.

other systems, the focus is on interoperability within the vendor's suite of applications.

2. A data exchange standard is not interesting for a vendor before it is a user requirement.
   - As long as a data exchange standard is not widely used or is an outspoken user requirement, it is difficult to justify spending resources on implementing it. It is easy to see here the danger of a deadlock situation; a standard will not be widely used before it is widely implemented. This may be resolved by powerful user organizations, such as government bodies or industry associations who demand such solutions.
3. STEP ISO 10303 is a complicated standard
   - STEP covers many large and complex engineering domains, and even understanding just a subset of the standard, still requires knowledge of its overall intention and structure.
4. FEA is a complicated domain
   - FEA is a very large discipline, and not every aspect of it is implemented in the exact same manner. A FEA solver calculates the simulation results by solving a huge number of equations. Different mathematical optimization methods may be used for this, and these may vary across solvers. Conceivably they should give the same results, but small discrepancies will occur. This is especially true for nonlinear solvers, where the algorithms are significantly more complex than for linear ones.
     Many solvers will also have functions for automatic time stepping, where solution time steps are decided by the solver based on certain criteria. These decisions and criteria will also depend on the particular solver implementation.
     Other examples are how the solver decides whether the solution has converged, how many iterations are used in a solution step, how element contact algorithms are implemented, and many more. All of these may have parameters for fine tuning the methods. In some solvers these may be fixed, in others they may be user specified, which across the solvers may have varying default values.

5. STEP AP209 has a limited scope
    - AP209, at its current state, is designed to cover linear static and linear modal analysis, while also being capable to be extended to cover nonlinear analysis.
      As stated in [42] the scope of AP209 "*...will address 60 to 90 percent of the analysis needs of an enterprise.*" and "*roughly 90 percent of the nonlinear problem is addressed at the present time*".
      The majority of all FEM analyses done are linear static. According to [43] this could be as much as 90%. However, for AP209 to gain more interest from FEA solvers, it is highly important to cover the remaining 10%.

Other causes, for lack of support of digital formats in general for PDF (Product Data Technologies), are outlined in [44]. In this paper some of the above items are addressed with focus on the last one, that is; How can AP209 be extended to also cover nonlinear finite element analysis?

3. State of the art of existing AP209 implementations

This section presents some use-cases where AP209 has been tested and implemented.

3.1. Early implementations

Some of the first implementations of AP209 are presented in [42], [45], [46]. These were centered on exchanging analysis, and especially composites information between different applications (including MSC/PATRAN), by exchanging the data via AP209 files. The studies are also summarized in [47].

Additionally, [48] presents the development of a translator between AP209 and the FEA solver SAMCEF for a usecase in the EU project *ENHANCE* [49].

3.2. CAx-IF and LOTAR

AP209 and other STEP parts, are documented in ISO documents as described in section 2.1. These documents provide the very formal definitions of the STEP data model. They do not, however, describe in details, implementation methods and use cases for the standards. CAx-IF [50] (Computer Aided X Implementor Forum and recently renamed to MBx-IF; *Model-based X Interoperability Forum*), is a joint effort between PDES, Inc., ProSTEP iViP and AFNeT, with the objective of accelerating CAD, CAE (Computer Aided Engineering) and other industrial data translators and ensure their compliance with the respective standards. CAx-IF performs test rounds where different software vendor participants develop translators between the standards and their own applications. Translations from and to the standard across the participants tools are then tested and verified by CAx-IF workgroups. By doing this, conformance with the standard is ensured, the applicability of the standard is verified, and potential improvements are suggested. Based on experience from these test rounds, Recommended Practices are documented and published. These documents, in contrast to the official ISO documents, focus rather on the implementations of the standard in translators. They are generally generic as to be applicable to any systems, and are essential for developers that are responsible of implementing support of the standard in their tools.

LOTAR [51] (Long Term Archiving and Retrieval) is an international organization, which aims to develop, test, publish and maintain standards for long-term archiving. LOTAR EAS [52] (Engineering Analysis and Simulation) workgroup joined CAx-IF in 2017 to handle the test rounds for AP209 FEA converters. A test round starts with an original set of files in the NASTRAN format, which are converted by all parties to AP209 and then shared to be imported and checked by the other parties. Statistics are calculated and checked by the EAS working group to verify the validity of the converters. From these results Recommended Practices and handbooks for AP209 are written and updated. Currently the following are published; Recommended Practices for AP209ed2 [53], STEP AP209 ed2 Linear Static Structural FEA Handbook Volume 1 [54] and Volume 2 [55].

3.3. TERRIFIC

Terrific [56] was an EU funded RD project (2011 - 2014) with the goal of improving the interoperability among applications for design, analysis and optimization of products. The focus was on further developing and applying isogeometric analysis (IGA), that is an innovative approach to close the gap between the 3D product representations in design and analysis [57]. A finite element mesh is not any more created from scratch on an idealized shape; instead, the NURBS (non-uniform rational B-splines) [58] of the original CAD design shape are reused as analysis mesh by just changing their parameterization. In Terrific, the process of updating AP209 and AP242 for IGA was started ([59, 60]). Particularly, locally refined B-splines were introduced to enable adequate re-parameterization of CAD-shapes for the purpose of engineering analysis.

3.4. Cloudflow

Another EU funded RD project related to AP209 was Cloudflow [61] (2013 - 2017), which aimed at smoother manufacturing processes by improved interaoperability of engineering applications within a cloud computing framework [62] for European manufacturing enterprises. CAD, CAM (Computer Aided Manufacturing), CFD and PLM were all part of cloud workflows using STEP-standards to more easily connect. In the CFD implementation, AP209 was used for managing simulation data on the cloud.

3.5. VELaSSCo

VELaSSCo [63], an EU funded RD project (2014 - 2016), aimed to provide new visualization methods of large-scale simulations. The project developed the VELaSSCo platform for accessing, visualizing, and querying distributed simulation information stored across multiple servers [64]. In the project, AP209 was validated and Discrete Element Method (DEM) extensions were proposed. AP209 was used for storing simulation data.

3.6. CAxMan

CAxMan [65], also an EU funded RD project (2015 - 2018) involving cloud systems, had the purpose of delivering Cloud based toolboxes and workflows to optimize design, simulation, and process planning for additive manufacturing. The goal was to be able to reduce material usage in additive manufacturing by simulating against both structural and thermal constraints and by providing automated feedback to the original design [66]. The various simulations and their links to the original design shape were facilitated by AP209.

3.7. CRYSTAL

More recently, Jotne EPM [67] and Lockheed Martin [68] has through the project CRYSTAL [69], developed AP209 converters for both Abaqus and NASTRAN formats. During this same project extensive support for AP209 was implemented in the SDM application EDMopenSimDM [70]. This allowed to relate CAD, FEA and PLM information in the same system stored in the same format; AP209. CRYSTAL took this one step further by also using AP209 for representing structural testing. This further allowed sensors and test results to relate to corresponding FEM analyses information. Having this data, from these different domains, in the same repository, and in the same format, enabled a more efficient way of retrieving, querying, and processing the engineering data. The study behind this integration is presented in [22].

3.8. Arrowhead Tools

Arrowhead Tools [71], is yet another ongoing EU project. Its goal is to reduce engineering costs by 40-60% for automation and digitalization solutions, by developing an open-source platform for design and run-time engineering of IoT (Internet of Things) and System of Systems [72, 73]. In the Arrowhead Tools project, AP209 will be used to represent and exchange simulation, sensor, and IoT information.

4. Recommended FEA extensions to AP209

The AP209 standard covers many of the data concepts needed for FEM analysis. Still, for a standard to be widely accepted, "many" may not be enough.

This section goes through the different aspects of FEA that would be expected in a standard format, which are either missing in the AP209 standard, or exist, but their use have never been implemented or documented in documents such as AP209 Recommended Practices or AP209 handbooks.

Section 4.1 describe certain requirements for AP209 as a FEA standard, and discuss how these are implemented in some of the major solvers. Without going in details, section 4.2 presents suggestions for how these requirements could be implemented or addressed in AP209.

The choice of solvers investigated was based on their market share and availability. The chosen ones were:

1. Abaqus 6.14 [74]
2. NX Nastran 11.0 [75]
3. Ansys MAPDL 19.0 [76]

These were all mentioned as leading vendors in [77] together with MSC Nastran [78].

4.1. FEA requirements for AP209

4.1.1. Analysis type categorizations

Table 1 shows an overview of how solver categorizes their supported analyses. In the table the term analysis *categories* is used as opposed to analysis *types*. The category names are based on what the solvers provide as analysis setups or solutions, which may involve multiple analysis types available for their load cases.

NX Nastran 11.0	Abaqus CAE 6.14	Ansys MPADL 19.0
SOL101 - Linear Statics	Static, General	Static Structural
SOL103 - Real Eigen Values	Static, Riks	Transient Structural
SOL103 - Response Dynamics	Dynamic Implicit	Rigid Dynamics
SOL105 - Linear Buckling	Dynamic Explicit	Harmonic Response
SOL106 - Nonlinear Statics	Buckle	Modal
SOL107 - Direct Complex Eigenvalues	Frequency	Explicit Dynamics
SOL108 - Direct Frequency Response	Static, Linear Perturbation	
SOL109 - Direct Transient Response	Steady-state dynamics, Direct	
SOL110 - Modal Complex Eigenvalues	Substructure generation	
SOL111 - Modal Frequency Response		
SOL112 - Modal Transient Response		
SOL129 - Nonlinear Transient Response		
SOL601(106) - Advanced Nonlinear Statics		
SOL601(129) - Advanced Nonlinear Transient		
SOL701 - Explicit Advanced Nonlinear Analysis		

Table 1: Categorization of analyses provided by different solvers.

It is important to note that in most solvers, a set of load cases on a finite element model, that relate, or are sequential, is often referred to as a *solution*. In most cases it is the solution that is initialized as a certain analysis type or category. The load cases that take part of this solution are then generally restricted to be only of one or a few specific analysis types, depending on the chosen solution. The specific limitations varies across the different solvers.

The exact type of analysis also depends on the analysis parameters that the user select. For example; Ansys provides the solution called "Static structural", but will then provide a choice to set it as linear or nonlinear, which are, in the context of this study, two different analysis types.

Depending on the analysis categories in table 1, a solver will decide which routines or algorithms to use in the analysis process and will require user input for certain parameters. The amount and type of user modifiable parameters varies for each solver.

4.1.2. Analysis parameters

In linear FEA, there are very few parameters that affect how the analysis is performed. Most solvers will solve a linear analysis using similar algorithms and give similar results. However, for nonlinear analyses, analysis parameters are very important. By analyses parameters, we mean settings the user may set that affect how the analysis is performed. This can be parameters such as the solver's; time step sizes, number of increment, maximum iterations, line search settings, type of convergence criteria, etc.

For nonlinear analyses, solvers always have different settings that may be set to specify how the model is solved. Some modifiable settings are common across most solvers, while others are specific to the individual solvers.

In Appendix A, tables A.2, A.3, A.4, A.5 and A.6, shows the most common analysis parameters for each analysis category and for the selected solvers for the FEA concepts of increment, arc-length, iteration, convergence and line-search parameters, respectively.

4.1.3. Variable depending loads and constraints

A common way of defining loads or constraints, especially in dynamic analyses, is to have a load or constraint magnitude that depends on time. It is also common, in both dynamic and static analyses that a load is defined as a field and depends on variables such as model coordinates. Typical examples are loads and constraints that are scaled throughout the analysis based on either a time dependent function or tabulated values, or a load depending on space dimension.

NX Nastran, Ansys and Abaqus all allow loads and constraints to be defined from a table or function with variables such as time, coordinates, temperature etc. In NX Nastran and Abaqus this is done by defining a load, such as nodal loads or element pressure, and then applying a tabular or functional amplitude to it. In Ansys, you may not amplify an existing load, but load values may be defined by a table.

4.1.4. Nonlinear material properties

There exists a wide range of different nonlinear material models. Every nonlinear capable FE solver offers the use of a subset of these. Some of the most general material models are; perfectly plastic, bi-linear and multi-linear plasticity material models. One thing to note however, is that each of these may be defined differently, for example via stress and strain values, or multiple E-modulus values. When defined by stress and strain pairs, these may be input as either true or engineering stress/strain values, depending on the solver.

NX Nastran, Ansys and Abaqus, each covers the material models mentioned above, as well as many other specific material models which will not be described in details.

4.1.5. Element contact

Element contact is when two element regions come into contact, and the solver uses algorithms to prevent the regions to overlap. Instead of overlapping, collision is simulated by calculating the appropriate deformations on the regions.

In solvers, contact is usually defined by first defining one or more regions, then defining interaction properties between or within the regions.

In NX Nastran regions are defined by selecting the nodes of faces on volume elements, or element sides on surface elements. In Abaqus, a surface on volume elements is defined by selecting the face IDs. For surface elements it is similar to Nastran. In Ansys however, contact regions are always defined by selecting nodes. The program then generates special contact elements based on the elements attached to those nodes.

In all three solvers, contact interaction properties may be defined and related to single regions or pairs of regions. A list of available contact parameters offered by these solvers are listed in Appendix B in table B.7. The listed parameters are the most common ones which may be found across the different solvers. There are, however, multiple more, which are very specific to each solver and their implemented algorithms.

4.1.6. Element gluing

By element gluing we mean two or more node or element regions that are defined to not separate by not allowing any deformation between them. The term *glue* is used in NX Nastran, while in Abaqus the equivalent is referred to as *tie*, and in Ansys, as *bonded*.

The solvers might implement this gluing differently, but essentially, for the user it is very similar to defining element contact as mentioned in section 4.1.5.

4.1.7. Superelements

Superelement (also known as substructure) reduction is a technique where parts of the FEM are divided in element groups; superelements. On each superelement, exterior nodes are defined, which can be used to connect to other superelements or normal elements. The model of the superelements are mathematically reduced such that their structural behaviour may be defined by only the degrees of freedom of the exterior nodes. This can greatly reduce computation time for large FE models.

NX Nastran, Abaqus and Ansys, all support the concept of superelements.

4.1.8. CAD-FEM relations

Generally, CAD/FEA applications allows for a mesh to be defined on a CAD model. Mesh regions can be created on CAD lines, surfaces and volumes. If the CAD shape is modified, the related mesh can then be regenerated. Similarly, loads, boundary conditions, contact regions, and other analysis definition, can be defined on the CAD geometry. The application will then automatically determine which nodes or elements these analysis definitions will be applied on.

The major FEA solvers have very good solutions for this type of FEA/-CAD associations. However, this information is only stored in the application's proprietary formats. The information in the files of these formats are not accessible outside the application, and are often only applicable for the specific version used. The FEA input files of the application contains only the FEA information, meaning that all CAD/FEA information is lost if the user wish to use another application.

The AP209 format support the representation of the CAD and FEA information, as well as their relation, such as the topological relation of geometric shapes in the CAD model, and elements or loads in the FEA model.

To be able to exchange such information between different systems, is very useful for engineers. The AP209 data model does support this sort of representations, but this capability has not been adopted or implemented by FEA/CAD applications.

4.2. Recommended extensions

4.2.1. Analysis type categorizations

In an AP209 model, the type of analysis type used is specified at the load case level. The type of entity used to define a load case, defines the type of analysis for that specific load case. Currently only linear static and modal analyses are supported in AP209.

The entity **control** represents the collection of load cases in the analysis; the solution. Load cases are represented by the entity **control_analysis_step** which has subtypes specific for linear static and linear frequency analysis. New subtypes could be added to this entity for each type of analysis to be supported. Ideally this could be organized as a hierarchy of sub-entities, such that these are organized based on being for example, static or dynamic, and linear or nonlinear. Special solutions, such as buckling analysis should also be considered.

4.2.2. Analysis parameters

AP209 does not have any specific entities for analysis parameters, and there are no documentations which describes how this should be represented.

There are a huge number of different existing parameters, and their availability vary with each solver and analysis type. Because of this, it is suggested that a generic method is used to represent each parameters. A generic method would mean an entity holding a parameter name, representing the actual parameter, and its value. The parameter names could be defined in a Recommended Practices, defining its meaning and appropriate use. The actual entities representing these parameters, should then reference a **control_analysis_step** (load case), or **control** (complete analysis) entity, if applicable for the whole analysis.

4.2.3. Variable depending loads and constraints

The existing entities for applying FE loads in AP209, does not have any options for representing a load value that varies. However, Part 50 [79] defines mathematical constructs such as function and tables, and Part 107 [80] defines how to represent relations between content in Part 50 and Part 104 [81]. Part 104 is the STEP part which defines most FEM specific data types, including applied loads. All of these are part of the 209 application protocol.

Using the content in those parts of the standard, and describing their implementation methods in a recommended practices, could enable AP209 to cover the above cases.

4.2.4. Nonlinear material properties

AP209 has specific entities only for defining linear material properties, such as E-modulus, poisson ratio, mass density, shell bending stiffness and more. Although, there are no specific entities for specifying nonlinear material properties, there are generic entities for material properties. These generic entities may be used to hold any type of values together with a material property name. Again, updated Recommended Practices could specify how to use these generic entities to represent nonlinear material properties.

4.2.5. Element contact

AP209 can collect elements in groups, but not which of the element faces belong to it. Meaning that you can't define element surface regions. There are also no specific entities for describing contact properties.

A possible simple extension, could be to create a new entity, inheriting from the the entity **element_group** and introducing an attribute that references the type ***element_aspect***. ***element_aspect*** is a STEP *SELECT type*, which can represent types such as ***volume_3d_face***, ***surface_3d_face***, etc. This way, surfaces could be defined using element groups. Additionally, for surfaces composed of sets of element faces with different IDs, AP209 already has the capabilities to relate multiple entity groups.

For defining the actual contact within or between the region(s) another new entity might be required. In AP209 the entity **state_definition** is a supertype of everything that is load or boundary condition related, or that somewhat defines the *state* of the FEA model. The most appropriate way to add contact definitions would be to extend the **state_definition** with new subtypes. It could be considered to add different entity types for specific cases, such as surface to surface contact and surface self contact. Another consideration, which hasn't been mentioned, are edge contacts, specially for 2D mesh models.

Parameters defining the properties and configuration of the contact could follow a similar generic approach as was discussed with analysis parameters in 4.2.2.

4.2.6. Element gluing

In the context of AP209, glueing should be implemented similarly to contact. The element region implementation could be used for both contact and glued regions. Specific entities for defining the actual glued connection between the regions, should also be done by extending the **state_definition** entity.

4.2.7. Superelements

AP209 contains the concept of *element substructures*, but this is not well documented and has not been implemented in previous AP209 studies. The entity **substructure_element_representation**, is a subtype of **element_representation**. This entity can collect multiple elements to define a superelement.

Recommended Practices need to be updated or created to describe the use of it.

4.2.8. CAD-FEM relations

The AP209 format support the representation of the CAD and FEA information, as well as their relation, such as the topological relation of geometric shapes in the CAD model, and elements or loads in the FEA model.

To be able to exchange such information between different systems, is very useful for engineers. The AP209 data model does support this sort of representations, but this capability has not been adopted or implemented by FEA/CAD applications.

5. Conclusion and future work

The point of having a standard data model for a domain such as FEA is to be able to store and exchange data regardless of its original format. An ISO standard model is maintained and ensured to be backwards compatible. The model is also *open*, meaning it is available to anyone who wish to adopt and implement it.

Such a model solves the problem of having to perform duplicate work when migrating or exchanging data from one system to another. It also prevents problems such as files being incompatible with newer versions of applications.

In addition to these mentioned benefits, data represented in STEP from any domain, may be related to other domains through it's PLM support.

In the CAD domain, STEP has shown, to a certain degree, to solve these problem and is widely used to move data between different systems.

As have been mentioned, the standard seem to lack support from FEA solver vendors. The main reason for this, is assumed to be lack of information scope for certain analysis types. To reconcile this, AP209 should extend its domain to be compatible with the type of advanced analysis that are available in existing solvers. The standard already has all the major generic building blocks (entities and data types) for many of the missing items, allowing it to easily extend its scope.

Future work is highly suggested to address the topics mentioned in section 4, and to define how these improvements should be implemented in the standard. This should be further pushed to the ISO STEP 10303 committee and documented in associated Recommended Practices.

References

[1] ISO 10303-1:1994. Industrial automation systems and integration – Product data representation and exchange – Part 1: Overview and fundamental principles. Standard, International Organization for Standardization, Geneva (Switzerland), 1994.

[2] ISO 10303-209:2014. Industrial automation systems and integration – Product data representation and exchange – Part 209: Application protocol: Multidisciplinary analysis and design. Standard, International Organization for Standardization, Geneva (Switzerland), 2014.

[3] ISO/IEC 10918-1:1994. Information technology – Digital compression and coding of continuous-tone still images: Requirements and guidelines. Standard, International Organization for Standardization, Geneva (Switzerland), 1994.

[4] ISO/IEC 10918-1:2004. Information technology – Computer graphics and image processing – Portable Network Graphics (PNG): Functional specification. Standard, International Organization for Standardization, Geneva (Switzerland), 2004.

[5] Microsoft Corporation. GDI documentation, Bitmap Storage. https://docs.microsoft.com/en-us/windows/win32/gdi/bitmap-storage?redirectedfrom=MSDN, 2018. Accessed on 14 February 2020.

[6] Autodesk Inc. Autodesk AUTOCAD 2020, About the DXF Format (DXF). http://help.autodesk.com/view/ACD/2020/ENU/?guid=GUID-235B22E0-A567-4CF6-92D3-38A2306D73F3, 2019. Accessed on 14 February 2020.

[7] Wavefront Technologies. Advanced Visualizer manual, Appendix B1. http://www.cs.utah.edu/~boulos/cs3505/obj_spec.pdf, 1992. Accessed on 14 February 2020.

[8] Inc. 3D Sytems. *Stereolithography Interface Specification.* Valencia, California, 1989.

[9] ISO/IEC 19775-1:2013. Information technology – Computer graphics, image processing and environmental data representation – Extensible 3D (X3D) – Part 1: Architecture and base components. Standard, International Organization for Standardization, Geneva (Switzerland), 2013.

[10] Alain Pfouga and Josip Stjepandić. Leveraging 3d geometric knowledge in the product lifecycle based on industrial standards. *Journal of Computational Design and Engineering*, 5(1):54 – 67, 2018.

[11] Arnulf Fröhlich. White paper: 3d formats in the field of engineering - a comparison. Technical report, PROSTEP AG, 2013.

[12] C. Jackson and D. Prawel. The 2013 State of 3D Collaboration and Interoperability Report (page 17). https://www.plm.automation.siemens.com/en_us/Images/Lifecycle-Insights-2013-Collaboration-Interoperability_tcm1023-210162.pdf, 2013. Accessed on 14 February 2020.

[13] John Michopoulos, R Badaliance, T Chwastyk, L Gause, and P Mast. Femml for data exchange between fea codes. 01 2001.

[14] *Development of the Finite Element Modeling Markup Language*, volume Volume 1: 22nd Computers and Information in Engineering Conference of *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 09 2002.

[15] *Advances of the Finite Element Modeling Markup Language*, volume Computers and Information in Engineering of *ASME International Mechanical Engineering Congress and Exposition*, 11 2005.

[16] Jerome Szarazi, Axel Reichwein, and Conrad Bock. Integrating finite element analysis with systems engineering models | nist. 2017.

[17] Byoung-Keon Park and Jay J. Kim. A sharable format for multidisciplinary finite element analysis data. *Computer-Aided Design*, 44(7):626 – 636, 2012.

[18] 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. Standard, International Organization for Standardization, Geneva (Switzerland), 2011.

[19] 10303-214:2010. Industrial automation systems and integration – Product data representation and exchange – Part 214: Application protocol: Core data for automotive mechanical design processes. Standard, International Organization for Standardization, Geneva (Switzerland), 2010.

[20] ISO 10303-242:2014. Industrial automation systems and integration – Product data representation and exchange – Part 242: Application protocol: Managed model-based 3D engineering. Standard, International Organization for Standardization, Geneva (Switzerland), 2014.

[21] T. Kramer and X. Xu. *STEP In a nutshell, in: Advanced design and manufacturing based on STEP*. Springer - Verlag London Ltd., 2009.

[22] R. Lanza, J. Haenisch, K. Bengtsson, and T. Rølvåg. Relating structural test and fea data with step ap209. *Advances in Engineering Software*, 127:96 – 105, 2019.

[23] Step application handbook iso 10303 version 3. Technical report, SCRA, North Charleston, South Carolina, USA, 2006.

[24] International Organization for Standardization (ISO). http://www.iso.org. Accessed on 14 February 2020.

[25] ISO 10303-11:2004. Industrial automation systems and integration – Product data representation and exchange – Part 11: Description methods: The EXPRESS language reference manual. Standard, International Organization for Standardization, Geneva (Switzerland), 2004.

[26] ISO 10303-239:2007. Industrial automation systems and integration – Product data representation and exchange – Part 238: Application protocol: Application interpreted model for computerized numerical controllers. Standard, International Organization for Standardization, Geneva (Switzerland), 2007.

[27] ISO 10303-235:2014. Industrial automation systems and integration – Product data representation and exchange – Part 235: Application protocol: Engineering properties and materials information. Standard, International Organization for Standardization, Geneva (Switzerland), 2014.

[28] ISO 10303-238:2012. Industrial automation systems and integration – Product data representation and exchange – Part 238: Application protocol: Product life cycle support. Standard, International Organization for Standardization, Geneva (Switzerland), 2014.

[29] Richard M. Botting and Anthony N. Godwin. Analysis of the step standard data access interface using formal methods. *Computer Standards & Interfaces*, 17(5):437 – 455, 1995. Formal Description Techniques.

[30] A. Goh, S.C. Hui, and B. Song. An integrated environment for product development using step/express. *Computers in Industry*, 31(3):305 – 313, 1996. Product and process data modelling.

[31] S Ma, Y Maréchal, and J.-L Coulomb. Methodology for an implementation of the step standard: a java prototype. *Advances in Engineering Software*, 32(1):15 – 19, 2001.

[32] Chandresh Mehta, Lalit Patil, and Debasish Dutta. *STEP in the Context of PLM*, pages 383–397. Springer London, London, 2009.

[33] P. Brun, F. Fernández González, M. Lehne, A. McClelland, and H. Nowacki. A neutral product database for large multifunctional systems. In R. Vio and W. Van Puymbroeck, editors, *Computer Integrated Manufacturing*, pages 87–97, London, 1991. Springer London.

[34] Soonhung Han, Young Choi, Sangbong Yoo, and Namkyu Park. Collaborative engineering design based on an intelligent step database. *Concurrent Engineering: R&A*, 10:239–249, 09 2002.

[35] Jeongsam Yang, Soonhung Han, Matthias Grau, and Duhwan Mun. Openpdm-based product data exchange among heterogeneous pdm systems in a distributed environment. *The International Journal of Advanced Manufacturing Technology*, 40(9):1033–1043, February 2009.

[36] Han M. Shih. Migrating product structure bill of materials excel files to step pdm implementation. *International Journal of Information Management*, 34(4):489 – 516, 2014.

[37] D. Iliescu, I. Ciocan, and I. Mateias. Assisted management of product data: A pdm application proposal. In *2014 18th International Conference on System Theory, Control and Computing (ICSTCC)*, pages 128–133, Oct 2014.

[38] Sébastien Charles and Benoit Eynard. Integration of cad and fea data in a pdm environment: Specification of a step simulation data management schema. *17th IMACS World Congress*, 02 2005.

[39] Guillaume Ducellier, Sébastien Charles, Benoit Eynard, and Emmanuel Caillaud. Traceability of simulation data in a plm environment: Proposition of a step-based system that support parameter integration. *9th International Design Conference, DESIGN 2006*, 01 2006.

[40] 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. Standard, International Organization for Standardization, Geneva (Switzerland), 2016.

[41] ISO 10303-22:1998. Industrial automation systems and integration – Product data representation and exchange – Part 22: Implementation methods: Standard data access interface. Standard, International Organization for Standardization, Geneva (Switzerland), 1998.

[42] Keith A. Hunten. Cad/fea integration with step ap209 technology and implementation. *MSC Aerospace Users Conference Proceedings*, 1997.

[43] Adams Vince. *A Designer's Guide to Simulation with Finite Element Analysis*. NAFEMS, 2008.

[44] Wim Gielingh. An assessment of the current state of product data technologies. *Computer-Aided Design*, 40(7):750 – 759, 2008. Current State and Future of Product Data Technologies (PDT).

[45] N. Pitre K.A. Hunten, J.W. Klintworth and T.E. Mack. New standards based data exchange bridge for design (cad), analysis (cae) and manufacturing (cam) of composite structures. In *MSC 1999 Aerospace Users Conferance Proceedings*, 1999.

[46] Edward L. Stanton, Takman Mack, Hiren D. Patel, and Jamie Klintworth. Composite beam models using iso step ap209. 2003.

[47] Keith A. Hunten, Allison Barnard Feeney, and Vijay Srinivasan. Recent advances in sharing standardized step composite structure design and manufacturing information. *Computer-Aided Design*, 45(10):1215 – 1221, 2013.

[48] Peter Bartholomew and Christian Paleczny. Standardization of the finite element analysis data-exchange in aeronautics concurrent engineering. *Journal of Computing and Information Science in Engineering - JCISE*, 5, 03 2005.

[49] The European Commission. Enhanced aeronautical concurrent engineering, ENHANCE. https://cordis.europa.eu/project/id/BRPR987001, 1999. Accessed on 24 February 2020.

[50] CAx Implementor FOrum (CAx-IF). https://www.cax-if.org. Accessed on 14 February 2020.

[51] NAS9300-001. Long Term Archiving and Retrieval of digital technical product documentation such as 3D, CAD and PDM data : part 101: Structure. Standard, Aerospace Industries Association of America Inc., 2017.

[52] LOTAR (Long Term Archiving and Retrieval). LOTAR EAS: Engingeering Analysis & Simulation Workgroup. http://www.lotar-international.org/lotar-workgroups/engineering-analysis-simulation/scope-activities.html. Accessed on 14 February 2020.

[53] Keith A. Hunten. *Recommended Practices for AP209ed2 10303-209:2014*. CAx Implementor Forum, 2016.

[54] Chris Johnson, Jochen Boy, Jean-Marc Crepel, Phil Roché, Joseph G. Draper, and Albert Lévy. *STEP AP209ed2 Linear Static Structural FEA Handbook - Volume 1: FEA Input for LOTAR EAS Pilot Study #1*. CAE Implementor Forum and Long Term Archiving and Retrieval (LOTAR), 2019.

[55] Chris Johnson, Jochen Boy, Jean-Marc Crepel, Phil Roché, Joseph G. Draper, and Albert Lévy. *STEP AP209ed2 Linear Static Structural FEA Handbook - Volume 2: FEA Steps, Loads and Boundary Conditions for LOTAR EAS Pilot Study #2*. CAE Implementor Forum and Long Term Archiving and Retrieval (LOTAR), 2019.

[56] The European Commission. Towards Enhanced Integration of Design and Production in the Factory of the Future through Isogeometric Technologies, TERRIFIC. https://cordis.europa.eu/project/id/284981, 2011. Accessed on 14 February 2020.

[57] Tor Dokken, Vibeke Skytt, Jochen Haenisch, and Kjell Bengtsson. Isogeometric representation and analysis: Bridging the gap between cad and analysis. 01 2009.

[58] Tor Dokken, Tom Lyche, and Kjell Pettersen. Polynomial splines over locally refined box-partitions. *Computer Aided Geometric Design*, 30:331â356, 03 2013.

[59] Tor Dokken and Oliver Barrowclough. Standardization and innovative applications essential for deployment of iga in industry. In *IGA 2018: Integrating Design and Analysis*, 2018.

[60] Tor Dokken, Vibeke Skytt, and Oliver Barrowclough. Trivariate spline representations for computer aided design and additive manufacturing. *Computers & Mathematics with Applications*, 78(7):2168 – 2182, 2019. Simulation for Additive Manufacturing.

[61] The European Commission. Computational Cloud Services and Workflows for Agile Engineering, CloudFlow. https://cordis.europa.eu/project/id/609100, 2013. Accessed on 14 February 2020.

[62] Havard Heitlo Holm, Jon Hjelmervik, and Volkan Gezer. Cloudflow - an infrastructure for engineering workflows in the cloud. 10 2016.

[63] The European Commission. Visulization for Extremely Large-scale Scientific Computing, VELaSSCo. https://cordis.europa.eu/project/id/619439, 2014. Accessed on 14 February 2020.

[64] Benoit Lange and Toan Nguyen. A hadoop use case for engineering data. In Yuhua Luo, editor, *Cooperative Design, Visualization, and Engineering*, pages 134–141, Cham, 2015. Springer International Publishing.

[65] The European Commission. Computer Aided Technologies for Additive Manufacturing, CAxMan. https://cordis.europa.eu/project/id/680448, 2015. Accessed on 14 February 2020.

[66] Lorenzo Tamellini, Michele Chiumenti, Christian Altenhofen, Marco Attene, Oliver Barrowclough, Marco Livesu, Federico Marini, Massimiliano Martinelli, and Vibeke Skytt. Parametric shape optimization for combined additive-subtractive manufacturing. *JOM*, 72(1):448–457, January 2020.

[67] Jotne EPM Technology AS. http://www.jotneit.no. Accessed on 14 February 2020.

[68] Lockheed Martin Corporation. https://www.lockheedmartin.com/. Accessed on 14 February 2020.

[69] J. Haenish, K. Bengtsson, R. Lanza, and O. Liestøl. Open Simulation Data Management and Testing, The Crystal Project. *NAFEMS World Congress 2017*, page 161, 2017.

[70] EDMopenSimDM (version 12.0). http://www.jotneit.no/edmopensimdm, 2019.

[71] The European Commission. Arrowhead Tools for Engineering of Digitalisation Solutions. https://cordis.europa.eu/project/id/826452, 2019. Accessed on 14 February 2020.

[72] Pal Varga, Fredrik Blomstedt, Luis Ferreira, Jens Eliasson, Mats Johansson, Jerker Delsing, and Iker MartÃnez de Soria. Making system of systems interoperable â the core components of the arrowhead framework. *Journal of Network and Computer Applications*, 81, 08 2016.

[73] J. Delsing. *IoT Automation: Arrowhead Framework*. CRC Press, 2017.

[74] Abaqus (version 6.14). https://www.3ds.com/products-services/simulia/products/abaqus/abaqusstandard/, 2014.

[75] NX (version 11.0). https://www.plm.automation.siemens.com/global/en/products/nx/, 2016.

[76] Ansys Mechanical (version 19.0). https://www.ansys.com/products/structures/ansys-mechanical-enterprise, 2018.

[77] TechNavio. Global Finite Element Analysis Market 2016-2020, 2016.

[78] MSC-NASTRAN. https://www.mscsoftware.com/product/msc-nastran.

[79] ISO 10303-50:2002. Industrial automation systems and integration – Product data representation and exchange – Part 50: Integrated generic resource: Mathematical constructs. Standard, International Organization for Standardization, Geneva (Switzerland), 2002.

[80] ISO 10303-107:2006. Industrial automation systems and integration – Product data representation and exchange – Part 107: Integrated application resource – Finite element analysis definition relationships. Standard, International Organization for Standardization, Geneva (Switzerland), 2006.

[81] ISO 10303-104:2000. Industrial automation systems and integration – Product data representation and exchange – Part 104: Integrated application resource: Finite element analysis. Standard, International Organization for Standardization, Geneva (Switzerland), 2000.

Appendix A. Solver analysis parameters

	NX			ABAQUS			ANSYS	
	SOL106	**SOL129**	**SOL601**	**Static, general**	**Static, Riks**	**Dynamic Implicit**	**Static**	**Transient**
Step time period				STATIC		DYNAMIC	TIME	TIME
Initial number of increments		TSTEPNL - NDT					NSUBST - NSBSTP	NSUBST - NSBSTP
Fixed number of increments	NLPARM - INC		TSTEP - Ni				NSUBST - NSBSTP	NSUBST - NSBSTP
Max number of increments	NLPARM - INC		TSTEP - Ni	STEP - INC	STEP - INC	STEP - INC	NSUBST - NSBMX	NSUBST - NSBMX
Min number of increments							NSUBST - NSBMN	NSUBST - NSBMN
Initial increment size		TSTEPNL - DT	TSTEP - DTi	STATIC		DYNAMIC	DELTIM - DTIME	DELTIM - DTIME
Fixed increment size		TSTEPNL - DT (ADJUST = 0)	TSTEP - DTi	STATIC		DYNAMIC	DELTIM - DTIME	DELTIM - DTIME
Max increment size			NXSTRAT - ATSMXDT	STATIC		DYNAMIC	DELTIM - DTMAX	DELTIM - DTMAX
Min increment size			NXSTRAT - ATSSUBD	STATIC		DYNAMIC	DELTIM - DTMIN	DELTIM - DTMIN
Tabular number of time steps			TSTEP - Ni				*	*
Tabular fixed step size			TSTEP - DTi				*	*
Tabular output Nth increment			TSTEP - NOi					
Max step size adj. Ratio		TSTEPNL - MAXR						
Min step size adj. Ratio		TSTEPNL - MAXR						
Step size adj. every n-th step		TSTEPNL - ADJUST						
Step size adj. function		TSTEPNL - MSTEP/RB						
Output Nth increment		TSTEPNL - NO	TSTEP - NOi					
Output all or last increment	NLPARM - INTOUT							

Table A.2: Increment parameters available per analysis type and solver.

	NX			ABAQUS			ANSYS	
	SOL106	**SOL129**	**SOL601**	**Static, general**	**Static, Riks**	**Dynamic Implicit**	**Static**	**Transient**
Min. arc length adj. ratio	NLPCI - MINALR							
Max. arc length adj. ratio	NLPCI - MAXALR							
Min. arc length adj. ratio to initial	NLPARM - MAXR						ARCLEN - MAXARC	ARCLEN - MAXARC
Max. arc length adj. ratio to initial	NLPARM - MAXR						ARCLEN - MINARC	ARCLEN - MINARC
Min. arc length increment					STATIC RIKS			
Max. arc length increment					STATIC RIKS			
Fixed arc length increment					STATIC RIKS			
Initial arc length increment	NLPARM - NINC (*)				STATIC RIKS		TIME + NSUBST - NSBSTP(*)	TIME + NSUBST - NSBSTP(*)
Max increment number	NLPCI - MXINC		NXSTRAT - LDCSUBD		STATIC RIKS			
Arc length scale factor	NLPCI - SCALE				STATIC RIKS			
Arc length mehtod type	NLPCI - METHOD							
Start displ. at node			NXSTRAT - LDCDISP					
Max. displ. at node			NXSTRAT - LDCDMAX		STATIC RIKS		ARCTRM - VAL	ARCTRM - VAL
Max. LPF					STATIC RIKS			

Table A.3: Arc-length parameters available per analysis type and solver.

	NX			ABAQUS			ANSYS	
	SOL106	**SOL129**	**SOL601**	**Static, general**	**Static, Riks**	**Dynamic Implicit**	**Static**	**Transient**
Max. iteration per increment	NLPARM - MAXITER	TSTEPNL - MAXITER	NXSTRAT - MAXITE				NEQIT - NEQIT	NEQIT - NEQIT
Advanced iteration parameters				CONTROLS - PARAMETERS = TIME INCR.	CONTROLS - PARAMETERS = TIME INCR.	CONTROLS - PARAMETERS = TIME INCR.		
Bisection controls	NLPARM - MAXBIS	TSTEPNL - MAXBIS						
Update stiffness matrix on first iter.	NLPARM - METHOD							
Update stiffness matrix at Nth iter.	NLPARM - METHOD/K-STEP			SOLUTION TECHNIQUE		SOLUTION TECHNIQUE		
Newton raphson (full)	NLPARM - METHOD/K-STEP	NLSTEP - KUPDATE		SOLUTION TECHNIQUE		SOLUTION TECHNIQUE	NROPT - FULL	NROPT - FULL
Modified newton raphson	NLPARM - METHOD/K-STEP						NROPT - MODI	NROPT - MODI
Quasi newston				SOLUTION TECHNIQUE		SOLUTION TECHNIQUE		

Table A.4: Iteration parameters available per analysis type and solver.

	NX			ABAQUS			ANSYS	
	SOL106	**SOL129**	**SOL601**	**Static, general**	**Static, Riks**	**Dynamic Implicit**	**Static**	**Transient**
Displacement (incl. rotation)	NLPARM - CONV	NLSTEP - CONV	NXSTRAT - CONCRI					
Displacement				CONTROLS - FIELD = DISPLACE-MENT	CONTROLS - FIELD = DISPLACE-MENT	CONTROLS - FIELD = DISPLACE-MENT	CVNTOL - Lab = U	CVNTOL - Lab = U
Rotation				CONTROLS - FIELD = ROTATION	CONTROLS - FIELD = ROTATION	CONTROLS - FIELD = ROTATION	CVNTOL - Lab = R	CVNTOL - Lab = R
Force (incl. moment)	NLPARM - CONV	NLSTEP - CONV	NXSTRAT - CONCRI					
Force							CVNTOL - Lab = F	CVNTOL - Lab = F
Moment							CVNTOL - Lab = M	CVNTOL - Lab = M
Work	NLPARM - CONV	NLSTEP - CONV	NXSTRAT - CONCRI					
Displacement tolerance (incl. rot)			NXSTRAT - DTOL					
Displacement tolerance	NLPARM - EPSU	NLSTEP - EPSU		CONTROLS - FIELD = DISPLACE-MENT	CONTROLS - FIELD = DISPLACE-MENT	CONTROLS - FIELD = DISPLACE-MENT	CVNTOL - TOLER	CVNTOL - TOLER
Rotation tolerance				CONTROLS - FIELD = ROTATION	CONTROLS - FIELD = ROTATION	CONTROLS - FIELD = ROTATION	CVNTOL - TOLER	CVNTOL - TOLER
Force (incl. moment) tolerance	NLPARM - EPSP	NLSTEP - EPSP	NXSTRAT - RTOL					
Force toletance							CVNTOL - TOLER	CVNTOL - TOLER
Moment tolerance							CVNTOL - TOLER	CVNTOL - TOLER
Work tolerance	NLPARM - EPSW	NLSTEP - EPSW	NXSTRAT - ETOL					

Table A.5: Convergence criteria parameters available per analysis type and solver.

	NX			ABAQUS			ANSYS	
	SOL106	**SOL129**	**SOL601**	**Static, general**	**Static, Riks**	**Dynamic Implicit**	**Static**	**Transient**
	SOL106	SOL129	SOL601	Static, general	Static, Riks	Dynamic Implicit	Static	Transient
Line search on/off	NLPARM - MAXLS	TSTEPNL - MAXLS	NXSTRAT - LSEARCH	CONTROLS - PARAMETERS = LINE SEARCH	CONTROLS - PARAMETERS = LINE SEARCH	CONTROLS - PARAMETERS = LINE SEARCH	LNSRCH	LNSRCH
Max. line search iterations	NLPARM - MAXLS	TSTEPNL - MAXLS		CONTROLS - PARAMETERS = LINE SEARCH	CONTROLS - PARAMETERS = LINE SEARCH	CONTROLS - PARAMETERS = LINE SEARCH		
Max. correction factor				CONTROLS - PARAMETERS = LINE SEARCH	CONTROLS - PARAMETERS = LINE SEARCH	CONTROLS - PARAMETERS = LINE SEARCH		
Min. correction factor				CONTROLS - PARAMETERS = LINE SEARCH	CONTROLS - PARAMETERS = LINE SEARCH	CONTROLS - PARAMETERS = LINE SEARCH		
Residual reduction factor terminate				CONTROLS - PARAMETERS = LINE SEARCH	CONTROLS - PARAMETERS = LINE SEARCH	CONTROLS - PARAMETERS = LINE SEARCH		
Ratio new to old cor. fac. terminate				CONTROLS - PARAMETERS = LINE SEARCH	CONTROLS - PARAMETERS = LINE SEARCH	CONTROLS - PARAMETERS = LINE SEARCH		
Relative energy error tolerance	NLPARM - LSTOL							
Error tolerance for W criterion	NLPARM - EPSW	TSTEPNL - EPSW						
Lower bound for Line Search			NXSTRAT - SLOWER					
Upper bound for Line Search			NXSTRAT - UPPER					
Line search convergence tolerance			NXSTRAT - STOL					

Table A.6: Line-search parameters available per analysis type and solver.

Appendix B. Contact parameters

	NX	ABAQUS	ANSYS
static friction coeffition	BCTSET - FRICi	FRICTION	MP - MU
min. search distance	BCTSET - MINDi		
max. search distance	BCTSET - MAXDi	SURF. INTERACTION - TRACK. THICKNESS	CONTAx REAL=6 (PINB)
handling of initial penetration	BCTPARA - INIPENE	SURF. BEHAVIOUR - PRESSURE-OVERCLOSURE	CONTAx KEYOPT(9), REAL=7 (PMAX), REAL=8 (PMIN)
contact algorithm: Constraint function	BCTPARA - TYPE		
contact algorithm: Segment method	BCTPARA - TYPE		
contact algorithm: Rigid target	BCTPARA - TYPE		
contact algorithm: Direct		SURF. BEHAVIOUR - DIRECT	
contact algorithm: Penalty		SURF. BEHAVIOUR - PENALY	CONTAx KEYOPT(2) = 1 / 3
contact algorithm: Augmented lagrange		SURF. BEHAVIOUR - AUGMENTED LAG.	CONTAx KEYOPT(2) = 0
contact algorithm: Multipoint constraint			CONTAx KEYOPT(2) = 2
contact algorithm: Lagrange			CONTAx KEYOPT(2) = 4
contact algorithm: Lagrange / penalty			CONTAx KEYOPT(2) = 3
double sided contact	BCTPARA - NSIDE	SURFACE (not specifying face/edge)	
contact surface thickness	BCTPARA - OFFTYPE / OFFSET	SURF. INTERACTION	CONTAx REAL=10 (CNOF)

Table B.7: Contact parameters available per analysis solver.

A.3 Paper 3 - Extending STEP AP209 for Nonlinear Finite Element Analysis

Contents

Vitae 2

Abstract 3

1 Introduction 3

2 The STEP ISO 10303 Standard 5
2.1 STEP 5
2.2 AP209 6

3 FEM Test cases and converter development 7
3.1 FEM Test cases 7
3.2 Converters 9

4 Proposed AP209 extension 9
4.1 Analysis types 9
4.2 Analysis parameters 10
4.2.1 STEP representation of analysis parameters 10
4.2.2 Proposed analysis parameters 14
4.3 Time varying loads 17

5 Test case results 18
5.1 Test Case A 18
5.2 Test Case B 19
5.3 Test Case C 20

6 Conclusion and future work 20

Remi K.S. Lanza
Jotne EPM Technology AS & Norwegian University of Science and Technology

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 PhD, which is a collaboration project between the Norwegian University of Science and Technology (NTNU) and Jotne. While pursuing his PhD, 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.

Terje Rølvåg
Norwegian University of Science and Technology

Prof. Rølvåg was born in Mo I Rana, 16/10-1963. Rølvåg holds a MSc. 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.

Jochen Haenisch
Jotne EPM Technology AS

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 Managerâ¢. 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.

Kjell A. Bengtsson
Jotne EPM Technology AS

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.

Extending ISO 10303-209 for Nonlinear Finite Element Analysis

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

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

Abstract

ISO 10303-209 [1, 2] is a standard for exchanging and storing simulation information along side related PLM (Product Lifecyle Management), CAD (Computer Aided Design), and other CAE (Computer Aided Engineering) data. The AP209 standard, despite being well documented and covering a wide range of engineering information, has not been widely implemented by FEA (Finite Element Analysis) solver or SDM (Simulation Data Management) applications. This is assumed to mainly be due to AP209 not yet supporting nonlinear FEA.

The following study takes basis in the findings of [3], where improvements of the AP209 standard were suggested. Some of these suggestions, related to nonlinear FEA, are here further investigated and implemented, and proposed for further standardization.

Analysis test cases using these new features are created, and converters between different FEA formats are developed.

The test cases are nonlinear, static and dynamic, with different defined time step control parameters, and loading conditions. The FEA data converters translates data between AP209 and the solver specific formats. The complete data information from the analyses are preserved during the conversion, and the generated analyses are solved. To confirm that no information was lost during the process, simulation results are investigated and compared.

Keywords: STEP ISO 10303, FEM Analysis, Nonlinear FEM, Data Exchange, Simulation Data Management

1. Introduction

There exist a wide range of different FEA (Finite Element Analysis) solvers. All of which have different strengths, by specializing in specific aspects of FEA. Some are part of larger application packages, and connect the FEA solver to CAD (Computer Aided Design) systems.

FEA solvers are commonly composed of a preprocessor, postprocessor and a solver. The preprocessor and postprocessor are generally available through a

*Corresponding author. Tel.: +47 452 04 992
Email addresses: remi.lanza@jotne.com (R.Lanza), jochen.haenisch@jotne.com (J.Haenisch), terje.rolvag@ntnu.no (T.Rølvåg)

user interface. The preprocessor is what the user use to create the mesh and define the analysis. If the system integrates a CAD application, the preprocessor's operations may be done directly on CAD shapes. The engineering work done while defining the FE model and analysis is typically stored in files using proprietary formats. When the model created in the preprocessor is complete, and ready to be solved, all the necessary information to solve the analysis is written to what is often called *input* files. These files, generally written in ASCII, are sent to the solver which computes the requested results. Results will often also be written in two formats. One using a proprietary format in binary, and one in ASCII. The binary files are sent to the postprocessor, which is where the user may interact with the results.

The ASCII input and result files, uses solver specific formats that are in most cases documented by the solver's provider. This allows engineers to manually create or edit the input files that are sent to the solver, without using the preprocessor. Custom scripts or applications can also be created to write these files for business specific cases.

Based on business needs and engineering tasks, the analysis engineers need to choose their FEA solvers. However, there might be many cases where multiple solvers are used, either within or across teams and companies. There can also be cases where engineers wish to migrate from one system to another. The difficulty with this, is that the analysis files of the different solvers, even if they are solving the same problems, do not share the same formats (they do not *talk* to each other).

As mentioned, the solvers have two types of files, proprietary and input files. The proprietary may hold more information than the input files. This can for example be how a pressure load is applied on a CAD surface or a specific section of the mesh, or a boundary condition applied to a CAD model edge. In the input files this information is translated to a set of nodal or elemental loads and nodal boundary conditions. The information of the connection between the analysis and the CAD model is lost. Other lost information can be related to the mesh generation. In the proprietary format, there might be stored how the mesh was generated based on element sizes over a surface, while in the input file only the generated mesh is stored. Obviously this is because the input file contains only the information necessary by the solver to run the requested analysis, and the proprietary format file is meant as a means to save the current work.

The Nastran format of input files is one of the most widely used format for analysis import in solvers. However, non-Nastran based solvers that support import of Nastran files, might only be able to import the mesh information and a sub-set of the available model information in these files.

There is a clear lack of support for data exchange of FE data across the different FEA solvers.

In the CAD domain, this problem has been addressed by most CAD software vendors adopting standard data formats. Multiple standards have been defined over the years, but most recently and most dominant, is the ISO 10303 STEP [1] standard. This format allows to exchange a wide range of different geometric representation, as well as associated PLM (Product Lifecycle Management) data.

CAD and PLM is just subset of what STEP can support. FEA is another domain implemented in the standard. The part of the standard that handles FEA is called ISO 10303-209 *Multidisciplinary Analysis and Design* (AP209)

[2]. Currently this standard has not widely been implemented by solvers, and has primarily support of linear static analysis, which was the focus during its creation. This study aims to propose certain extensions to the standard to allow it to cover additional aspects of FEA, with special focus on nonlinear FEA.

A preliminary study on this topic has been done in [3], and in this study we implement some of its proposed improvements.

The paper is organized as follows; section 2 gives a short introduction to the STEP standard and AP209. Section 3 presents the test cases and discusses the converter development, whereas section 4 goes into details of the recommended AP209 extensions. Section 5 presents the results of the converted test cases. Finally a conclusion and proposed future work is discussed in Section 6.

2. The STEP ISO 10303 Standard

2.1. STEP

ISO 10303 STEP is a large standard covering multiple engineering domains for which it provides interoperable data models. Only a short introduction of STEP is presented in this study, for more information the following resources are suggested to the reader; chapter 2 of *ISO 10303-209 - Why and how to embed nonlinear FEA* [3], chapter 2 of *Relating structural test and FEA data with STEP AP209* [4], *STEP in a Nutshell* [5] and *STEP Application Handbook* [6].

On the highest level, STEP is divided in Application Protocols (AP), which each can be used as data models in different engineering domains. Most known, and often referred to as only "STEP", are the APs; AP203 [7], AP214 [8] and AP242 [9]. These are all mainly used in CAD applications and defines the data model of exported and imported STEP files. As applications from different vendors implement these data models, they may share CAD data between them.

One level lower in the STEP architecture, we have documents referred to as *parts*. Each AP consists of a set of parts, some of which are generic and are used by multiple APs. This enable various APs to share common generic information. A typical example of such generic information is PLM data and basic geometric definitions.

At the lowest level, we have *entities*. These are defined in the *parts* documents, and define how to represent different data objects. They are similar to classes in an object oriented programming language; they can hold attributes, reference other entities, and can inherit from other entities. Some examples of entities are; **product**, which defines the representation of a product, **axis2_placement_3d**, which defines a coordinate system, **surface_3d_element_representation**, which defines a surface element in a FEM (Finite Element Method) model. AP209 consists of over 2500 such entities.

In the different parts, these entities are formally described in the EXPRESS [10] data modeling language, which is itself also standardized by STEP.

The AP data model documents can be used to define database dictionaries for systems that wish to implement a certain AP. APIs can also be generated from them to create applications that works with the same data model.

2.2. AP209

AP209 is called *Multidisciplinary Analysis and Design*. The newest edition is edition 2 (or AP209e2), and AP209 edition 3 is currently being developed. Any reference in this paper to AP209, is referring to AP209e2.

AP209 is a STEP data model used for defining a data exchange format for simulation tools, mainly FEA applications, and as a data storage format for SDM (Simulation Data Management) tools. As described in section 2.1, STEP has been used to simplify data exchange between CAD tools, and the same could ideally be done with AP209 and FEA solvers.

A number of older studies related to AP209 have been published; related to data exchange of composites and FEA in general [11, 12, 13, 14], Thermal data exchange [15], electromagnetism [16], and SDM and FEA translation [17, 18].

Other attempts of introducing standard FEM formats are; the XML based format FemML [19], the XML based format PAM [20], and using SysML (Systems Modeling Language) [20].

In these cited studies certain arguments are used for why not to use AP209.

- The *FemML study* suggests that the greatest disadvantages with AP209 are;

  "(...)*its massive specifications and custom and proprietary related tool availability*(...)".

  It is true that AP209 has massive specifications, which are complex and possibly increases implementation time. However, for a standard to cover an engineering domain as complex as FEA, and also relate it to PDM, CAD, and other information, such a complexity is required. The study is relatively old, and there exist now various tools and APIs to work on STEP data, such as; EDMsdk [21] by Jotne EPM Technology AS [22], ST-Developer [23] by STEPtools [24], STEPcode [25], an open source project previously maintained as STEP Class Library (SCL) by NIST [26] (National Institute of Standards and Technology), and JSDAI [27] from LKSoft [28].

  As the format is open, it is possible for anyone to create such tools and APIs.

- The *PAM study* states that;

  "(...)*it accommodates all the types of data from both CAD and CAE as a standard protocol, and this characteristic easily increases the size of the data.*".

  AP209 does support for multiple domains, it does not, however, have to be used. An AP209 model can be fully complete with only FEA data without any additional information.

- The *SysML study* states that;

  "*For the simulation part, AP209 defines a FEA model with a collection of non-constrained string entities (analysis type, creating software, finite element name, material...) leading to an informal FEA description, as in Figure 1. STEP AP209 provides only syntactic interoperability, not semantic interoperability. It cannot ensure the same simulation results across FEA tools or support integration with other kinds of tools.*".

This is a misunderstanding; AP209 does have very strict semantic interoperability, and information such as analysis type, finite element name, material, does not, in fact, follow non-constrained string entities, but defined entity names or type enumerations.

3. FEM Test cases and converter development

As have been presented in the previous study [3], some FEA aspects are not covered by AP209, especially concepts related to nonlinear analysis. In this study we elaborate on a few of those items:

- Analysis types (section 4.1)
- Analysis parameters (section 4.2)
- Variable dependent loads and constraints (section 4.3)

FEM analysis test cases using these items are presented in section 3.1. The solvers used for these are the following:

- Abaqus 6.14 [29]
- NX Nastran 11.0 [30]
- Ansys MAPDL 19.0 [31]

All applications were used on a Windows 10 system.

To validate the recommended extensions, AP209 converters were developed to process and convert the test cases for the different solvers. The development of these converters are discussed in 3.2.

3.1. FEM Test cases

Each of the following test cases were created in Abaqus 6.14. In Abaqus, units are not specified, but are expected to be consistent with each other. Therefore, in the next sections, units are not described.

As the focus of this study are the solution parameters, simple geometry and mesh were chosen for the models.

Test case A

The model is based on the verification test entitled *Large Displacement Elastic Response of a Hinged Spherical Shell Under Uniform Pressure Loading* from the *NX Nastran 11 Verification Manual* [32].

- **Mesh geometry**
  The geometry of the mesh is defined by a surface where $Z = 2.0285 \cdot 10^{-4}(X(1570 - X) + Y(1570 - Y))$ and limited by $X = [0.0, 1570.0]$ and $Y = [0.0, 1570.0]$.
- **Mesh**
  64 shell elements of type S4R with a thickness of 100.0
- **Material**
  $E = 69$ and $\nu = 0.3$
- **Boundary conditions and load**
  All nodes on the edges are hinged, and a pressure load of -0.1 is applied on all elements.
- **Analysis type and parameters**

  Nonlinear static solution
  Fixed increment size = 0.04
  Maximum number of increments = 25
  Total step time = 1
  Full Newton-Raphson

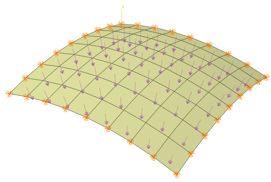


Figure 1: Test case A

Test case B

The geometry, mesh and load of this test case is the same as in test case A, but this time with arc-length control.

- **Mesh geometry**
  Same as Test A
- **Mesh**
  Same as Test A
- **Material**
  Same as Test A
- **Boundary conditions and load**
  Same as Test A
- **Analysis type and parameters**

  Nonlinear static solution with Arc-length method (Riks)
  Maximum number of increments = 950
  Initial Arc length increment = 0.005
  Minimum Arc length increment = 0.0005
  Maximum Arc length increment = 0.05
  Estimated total arc length = 1.0

Test case C

- **Mesh geometry**
  3D cantilever beam with width = height = 50.0 and length = 1000.0
- **Mesh**
  1072 8-noded linear hexahedral elements of type C3D8I
- **Material**
  $E = 206.94 \cdot 10^6$ and $\nu = 0.288$
- **Boundary conditions and load**
  All nodes on one end are hinged
  Time varying load distributed on the nodes of opposite end, defined by the following [time, load] values:
  $[0.0, 0.0], [0.01, 1.0], [0.02, 0.0], [1.0, 0.0]$
- **Analysis type and parameters**

  Nonlinear dynamic solution
  Total step time = 0.2
  Maximum number of increments = 1000
  Initial increment size = 10^{-3}
  Minimum increment size = 10^{-5}
  Maximum increment size = 10^{-3}
  Rayleigh damping factors:
  - Mass term = 0.02
  - Stiffness term = 0.001

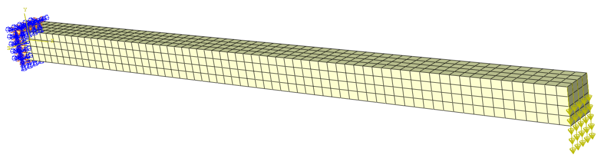


Figure 2: Test case C

3.2. Converters

The converters developed for the purpose of this study were all implemented in a command line application created in C++. The application takes in a source file of a FEM simulation, a target path for the converted file, and parameters specifying the source and target formats.
The accepted formats are Nastran, Abaqus, and Ansys ASCII input files, as well as AP209 STEP files.
At the time of writing, the converter accepts the following one-way translations:

- AP209 to Ansys
- AP209 to Nastran
- Nastran to AP209
- Abaqus to AP209

When neither source nor target formats are STEP, the source file is first converted to AP209 and then to the target format. Which means the following translations requests are also accepted by going through an AP209 conversion:

- Nastran to Ansys
- Abaqus to Nastran
- Abaqus to Ansys

The reason for going for the solver's ASCII input formats instead of using binary files or solver specific APIs, was due to complexity, lack of documentation, and limited coverage by the APIs. Using the ASCII formats might somewhat reduce the "robustness" of the converters, but simplifies the development and serves the purpose of validation sufficiently.
The interfaces to and from STEP data respresentation is done using EDMsdk^{TM} [21] which through an API creates a STEP database based on the AP schema. It is also used to generate a STEP AP209 API in C++ that interfaces the STEP database. When the converter then works with AP209 data, it accesses data-read and write directly in the database.
With exception of the to- and from- Nastran converters, the converters have a limited scope and are specifically tailored to handle the types of data of the test cases and the suggested extensions presented in this paper.

4. Proposed AP209 extension

4.1. Analysis types

In most solvers, a set of load cases on a finite element model, that relate, or are sequential, is often referred to as a *solution*. Usually, a solution will be initialized as a certain analysis type or category. Each load case that take part of this solution may be of different analysis types. However, in general, the solver sets restrictions on which load case type (or analysis type) may be under the same solution. The specific restrictions varies across the different solvers.
In AP209, the entity **control** represents the collection of load cases in the analysis; the solution. In this proposed extension, the different analysis types are

implemented as subtypes of the entity **control_analysis_step**, which represents a load case. No restrictions are proposed to which analysis types (or load case types) may relate to the same **control** (or be part of the same solution).
A common concept in FEA analysis, is to have dependent sequential load cases, where the results of one load case defines the starting point of the next. This is also possible to represent in AP209, where the results of an analysis are represented by the entity **result_analysis_step**, and may define the initial state of a subsequent load case. Another important entity is the **control_process**. It relates to the **control_analysis_step**, **result_analysis_step**, and the boundary and load conditions for the specific load case.
The inheritance diagram of the above mentioned entities is shown in Figure 3.

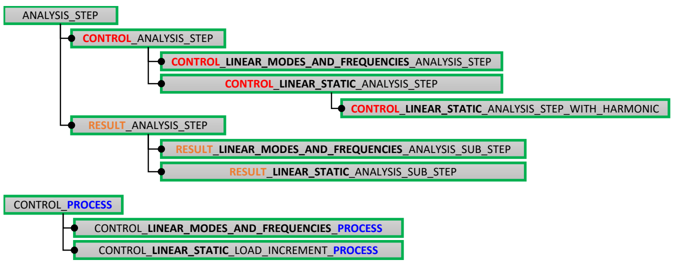


Figure 3: Existing analysis types in AP209

Based on the available analysis types in the different solvers, an extended inheritance structure is proposed in Figure 4 to cover all of those. The entities **result_analysis_step** and **control_process** need to be extended in the same manner.

4.2. Analysis parameters

4.2.1. STEP representation of analysis parameters

The current AP209 schema does not have the concept of solver or analysis parameters. A generic method is proposed in this paper to relate parameters to either the complete analysis or a specific load case. By generic we mean that we do not introduce new entity types for each possible parameter, but instead propose a property name (fea solver property name) for each parameter. Existing generic STEP entities can then be used to hold the property values and the corresponding property name. In this way, the list of allowable analysis parameters can easily be extended. Such a list of parameters should be added to a future update of the recommended practices for AP209, if accepted by the vendor community of the ISO 10303 CAx-IF [33].
The following STEP entities are re-used for the purpose of representing analysis parameters:

- Defined in Part 41 [34] of the STEP standard:

  property_definition

  Used to add a property type and value to a STEP object.

  property_definition_representation

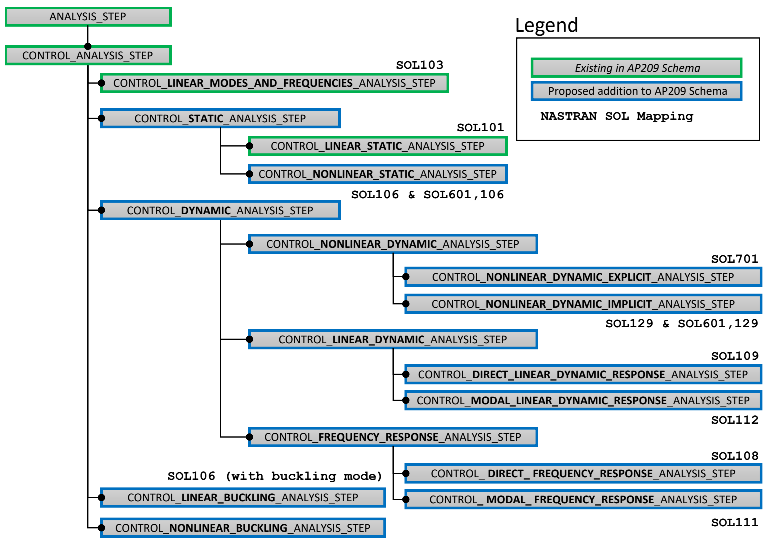


Figure 4: Proposed hierarchy of analysis types. The Nastran SOL mapping refers to equivalent solution names in Nastran.

Used to relate a **property_definition** to a **representation** that contains the property value.

- Defined in Part 43 [35] of the STEP standard:

representation

A generic entity, that for a single context, that is, type of description, collects **representation_items**, that is, values that make up this representation.

representation_item

A generic entity that is the supertype of a wide range of different entities, representing anything from simple values, such as a numerical value or a boolean, to geometric items, such as a coordinate, surface, curve etc.

The entities **property_definition** and **property_definition_representation** are proposed to be specialized into the subtypes: **fea_parameter_property** and **fea_parameter_property_definition_representation**. The **fea_parameter_property** extends **property_definition** by adding a new attribute *assigned_control_or_control_process*. The subtyping of **property_definition_representation** does not introduce new attributes, but adds semantic , which may be used for automated data validation, for example.
The EXPRESS specification of these entities are shown in Listing 1.

```
ENTITY fea_parameter_property
    SUBTYPE OF (property_definition);
    assigned_control_or_control_processes :
        SET OF control_or_control_process;
END_ENTITY;

ENTITY fea_parameter_property_definition_representation
    SUBTYPE OF (property_definition_representation);
END_ENTITY;
```

Listing 1: EXPRESS code of new entities.

The intent is that the **fea_parameter_property** defines a *category* of one or multiple analysis parameters. The entity **fea_parameter_property** relates to a **representation** via **fea_parameter_property_definition_representation**. The **representation** holds the analysis parameter category name in its *name* attribute. The individual analysis parameters that are set for the analysis in that category are defined in already existing subtypes of **representation_items** and are related to the **representation** via the *items* attribute. The names of the of analysis parameters are set in the *name* attribute of the **representation_item**. The used subtype of **representation_item** depends of the value type of the parameter (boolean, real, integer or string), and is set in the appropriate attribute.
The attribute *assigned_control_or_control_process* of **fea_parameter_property** holds a new select type that is introduced; ***control_or_control_process*** (shown in Listing 2). This select type may be one of the two following existing entities (defined in Part 104 [36] of the STEP standard):

- **control**

  An entity meant to associate analysis controls to a model. In a FEA analysis, this entity represents the complete analysis and is related to all load cases.

- **control_process**

  An entity which defines how an analysis departs from its initial state. In a FEA analysis this entity relates the loads and boundary conditions to a specific load case.

```
TYPE control_or_control_process = SELECT (
    control,
    control_process
);
END_TYPE;
```

Listing 2: EXPRESS code of new SELECT type.

The purpose of this new select type, is to allow FEA parameters to be either applicable to the whole simulation and all load cases (**control**), or only for a specific load case (**control_process**).
An example showing how analysis parameters can be related to a load case is shown in Figure 5.
The name attribute of the entities should be as follows:

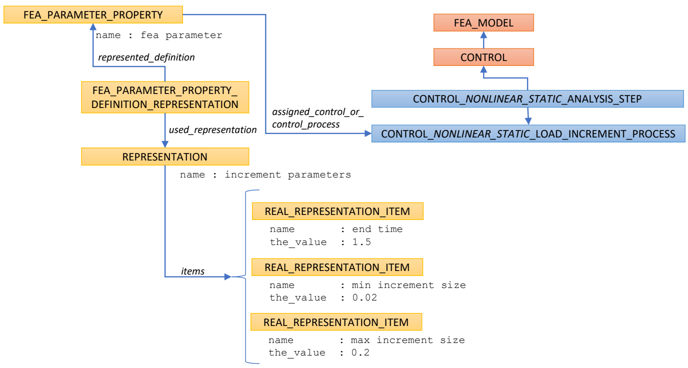


Figure 5: Instance diagram example of fea parameters related to a load case.

- **fea_parameter_property** - "*fea parameter*"
- **representation** - Category name of parameters defined in the *items* attribute.
- **representation_items** - The name of the individual analysis parameters.

In the example in Figure 5, the category of analysis parameter is *increment parameters*. Within this category we have multiple analysis parameters such as *minimum* and *maximum increment size*.
A challenge when defining a standard format that covers advanced FEA parameters is that not all solvers cover the same types of parameters. In some cases, the parameters are not the same but still convertible to an equivalent. A typical example of this is step increment sizes and increment numbers. While one solver might allow to specify increment sizes, another one might only allow to specify increment numbers. As long as the analysis time is known, it is possible to convert these parameters between the two solvers. Another scenario is if a minimum step increment size is specified in format A, but format B does not have this parameter, nor may it specify a maximum increment number. A conversion from A to B would then result in loss of information.
In the converters developed for this study, a logging system was implemented specifically for FEA parameters. In the conversion process, whenever a FEA parameter is converted between AP209 and the native format, the parameter values in source and target are logged. Additional information is also logged in cases where calculations were done to convert a certain parameter, or if a parameter is not supported by the target format.
Such logging is crucial as the engineers exchanging data from one system to another need to be aware of any information that did not properly convert. The engineer should also be informed on *how* the parameters were converted. In the converters developed for the test cases in this paper, that information is given as a text log file. A solver that implements export and/or import functionality between its native formats and AP209 (or other formats), should properly show this information to the user when the exchange is completed.

4.2.2. Proposed analysis parameters

The study [3] presents tables of supported FEA parameters in major solvers in its appendix. Based on these, table 1 lists recommended analysis parameters for AP209. Some of these were implemented using the method presented in 4.2.1, and are recommended to be included in future recommended practices of AP209.

All parameters listed under, are either valid for the whole analysis, if the corresponding **fea_parameter_property** refers to a **control** entity, or, for only a specific load case, if it refers to a **control_process** entity.

	Increment parameters
	`start time`
	Defines the start time of the total analysis time the increment parameters are applicable for. If this value is unset or not existing, the parameters are applicable from the start of the analysis if applied to the **control** entity, or the beginning of the load cases if applied to one or more **control_process** entities.
	`end time`
	Defines the end time of the total analysis time the increment parameters are applicable for. If this value is unset or not existing, the parameters are applicable throughout the whole analysis if applied to a **control** entity, or the whole load case duration if applied to **control_process** entities.
	Note on `start time` and `end time`
	• If a **fea_parameter_property_definition_representation** (**fppdr**) contains the parameters `start time` and `end time`, then all increment parameters in the same **fppdr** are applicable only for that specified time frame. • If a **fppdr** has a `start time`, without an `end time`, the parameters are valid from that specific time, until the end of the load case or analysis. • If multiple **fppdr** referencing the same **control** or **control_process** (via **fea_parameter_property**), each containing a `start time`, but no `end time`, the different `start times` should be sorted, and each consecutive `start time`, defines a time frame in which the end time is the next `start time`. Thus all increment parameters have specified time frames. An example can be seen in Figure 6.
	`fixed increment size`
	Defines a fixed increment size to be used throughout the whole analysis, whole load case, or time frame.
	`initial increment size`
	Defines an intial increment size to be used in the begining of an analysis, load case, or beginning of time frame.
	`min increment size`
	Defines a minimum increment size to be used throughout the whole analysis, whole load case, or time frame.
	`max increment size`

	Defines a maximum increment size to be used throughout the whole analysis, whole load case, or time frame.
	`fixed increment number`
	Defines a fixed increment number for each load case in the analysis, one specific load case, or within a time frame.
	`initial increment number`
	Defines an initial increment number used by the solver to calculate an inital increment size, for the begining of each load case in the analysis, one specific load case, or at the beginning of a time frame.
	`min increment number`
	Defines a minimum increment number to be used by the solver to limit the increment sizes, throughout all load cases in an analysis, a specific load case, or a time frame.
	`max increment number`
	Defines a maximum increment number to be used by the solver to limit the increment sizes, throughout all load cases in an analysis, a specific load case, or a time frame.
Iteration parameters	
	`max iterations per increment`
	Defines the maximum number of iterations to use in each increment.
	`iterations before stiffness update`
	Defines after how many iterations in an increment the stiffness matrix is updated. (This value can define if the *full* or *modified* Newton method is used.)
	`update stiffness at first iteration`
	A boolean that defines if the stiffness matrix is always updated at the first iteration of an increment.
Convergence parameters	
	`tolerance variable`
	Defines a convergance variable. Should be implemented as an extendable enumeration, including the following enumerations: `DISPLACEMENT_CONVERGANCE`, `ROTATION_CONVERGANCE`, `ENERGY_CONVERGANCE`, `FORCE_CONVERGANCE`.
	`tolerance value`
	The tolerance value for the specified `tolerance variable`.
Large displacement	
	`use large displacements`
	Boolean that defines if large displacements are accounted for in specific load case or all load cases in an analysis.
	`use large strains`
	Boolean that defines if large strains are accounted for in specific load case or all load cases in an analysis.
Arc Length control parameters	
	`min arc length`
	Defines the minimum allowable arc length in the arc length increment process.
	`max arc length`
	Defines the maximum allowable arc length in the arc length increment process.

	`min arc length adjustement ratio`
	Defines the minimum *new arc length* to *previous arc length* ratio
	`max arc length adjustement ratio`
	Defines the maximum *new arc length* to *previous arc length* ratio
	`min arc length adjustement ratio to initial`
	Defines the minimum *new arc length* to *initial arc length* ratio
	`max arc length adjustement ratio to initial`
	Defines the maximum *new arc length* to *initial arc length* ratio
	`load scaling`
	Defines a scaling factor for the arc length method.
	`method`
	Defines the method used in the arc length method process. Should be implemented as an extendable enumeration, including the following enumerations: `RIKS`, `RIKS MODIFIED`, `CRISSFIELD`.
Line search parameters	
	`use line search`
	A boolean that defines if line search should be used in the increment process.
Damping parameters	
	`constant rayleigh stiffness matrix term`
	When using Rayleigh damping the camping matrix is calculated $C = \alpha M + \beta K$ where M is the mass matrix and K is the stiffness matrix. This term sets the β value.
	`constant rayleigh mass matrix term`
	When using Rayleigh damping the camping matrix is calculated $C = \alpha M + \beta K$ where M is the mass matrix and K is the stiffness matrix. This term sets the α value.

Table 1: Recommended FEA parameters to be supported by AP209.

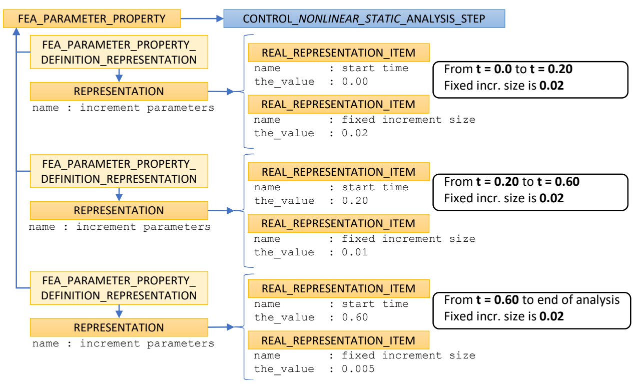


Figure 6: Instance diagram example of FEA parameters related to a load case.

4.3. Time varying loads

As mentioned in 4.1 the entity **control_process** collects the loads of a specific load case. The different load types are all subtypes of the **state_definition** entity, which includes, among others, nodal forces or element pressure loads. The relation between specific loads and the **control_process** is done with what could be called a *state tree*. Two entities; **state** and **state_relationship** are used to combine different loads, and optionally scale them. An example of such state tree in Figure 7, shows the entity structure from the loads all the way up to the load case and analysis. The figure shows how, in AP209, a load case combines; *1)* a group of pressure loads, *2)* two single nodal loads that are grouped together, and *3)*, two groups of nodal loads that are used in a **linearly_superimposed_state** which can scale the load in a **state** via a **state_component**. Each **state** may be reused in multiple load cases and in different load combinations.

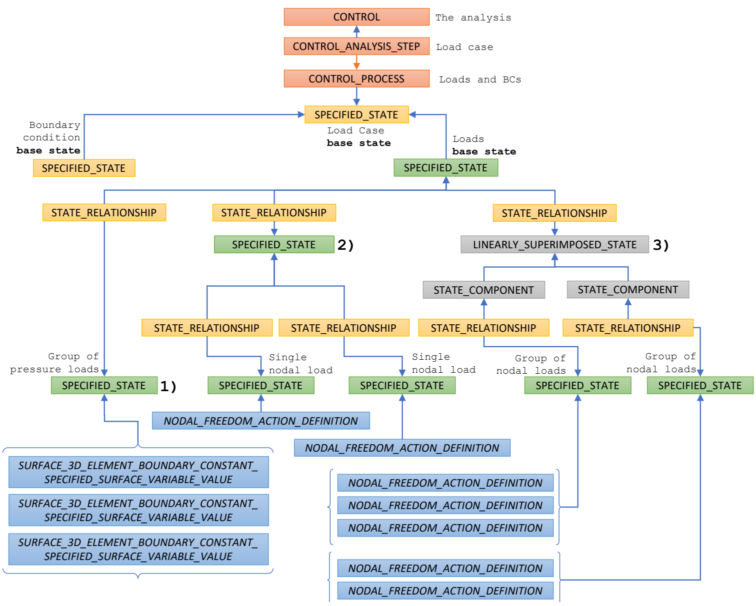


Figure 7: *State tree* example showing how **state** and **state_relationship** are used to relate loads to a load case.

No known implementations of AP209 have used the concept of loads varying with other variables. Still, the standard holds concepts that can be used to represent it, but this need to be formalized and added to the standard's *Recommended Practices*. The following is a proposal on how this could be done. Part 104 of the STEP standard defines all the FEA specific entities that have been presented so far. Part 50 [37] defines mathematical constructs such as functions and tables, while Part 107 [38] defines how to represent relations

between content in Part 50 and Part 104. (All of these parts are included in AP209).
Using the entity **standard_table_function** a generic table can be represented. The details around the construct of this table are left out of this paper. Essentially, when this table is used to hold time depending scaling factors, one column holds scaling factors, and a second column time values. Similarly, for coordinate dependent scaling factors, columns can be used for coordinate values. The concept of interpolation of values between the variables, and extrapolation of values outside the variables range, are not discussed in this study.
The recommended way of defining a load being scaled using varying values, is by assigning a **state_definition** to the **state_component** and relate it to a **property_definition** via a **distribution_view_relationship**, as seen in Figure 8. (The **property_definition** is directly related to the actual table holding the scale values, which is defined through Part 50 constructs). All loads related to the **state_component** have then been applied with a varying scale.

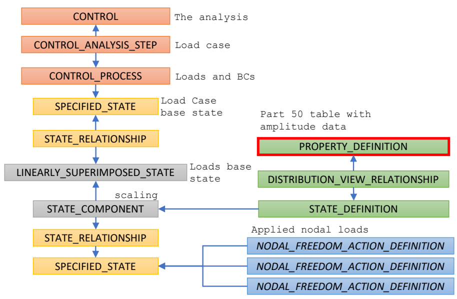


Figure 8: Example STEP structure of table defining amplification scale of nodal loads.

5. Test case results

5.1. Test Case A

As can be seen in Figure 9 the three solvers give similar result until the load factor reaches around 0.64.
The reason for the subsequent deviation, is that Abaqus, by default does not use line-search. Without modifying any line-search parameters, the analysis will stop when reaching the buckling load. NX-Nastran and Ansys, have by default line-search activated, and therefore manage to solve past the buckling point (without giving realistic results during the buckling).
Conversion of line-search parameters was not implemented in the converter, and therefore, the resulting analyses give different results. Ideally, the converter from Abaqus to AP209, should be aware that line-search is off by default and apply this information to the generated AP209 model.
All three simulations uses the same number of increments until reaching the buckling point.
The maximum relative difference in displacement between the solvers is under 2% until the load factor reaches 0.55. It then increases through the buckling.

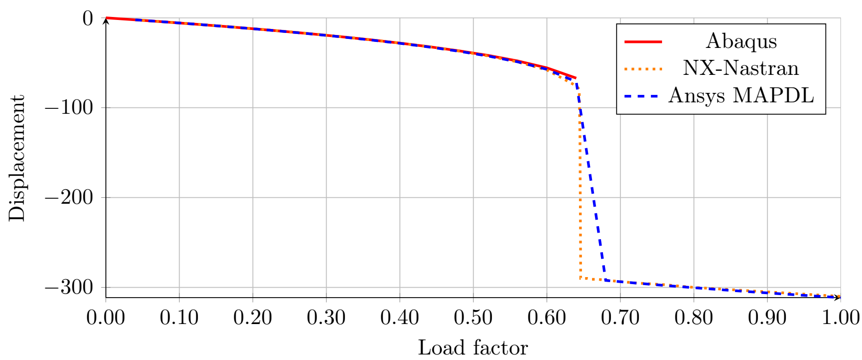
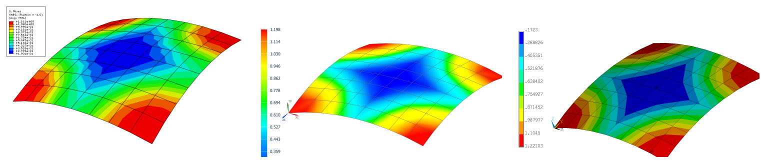


Figure 9: Test case A - load factor vs. displacement at center node.



(a) Abaqus (b) NX Nastran (c) Ansys

Figure 10: Test case A - Von Mises stress on bottom surface

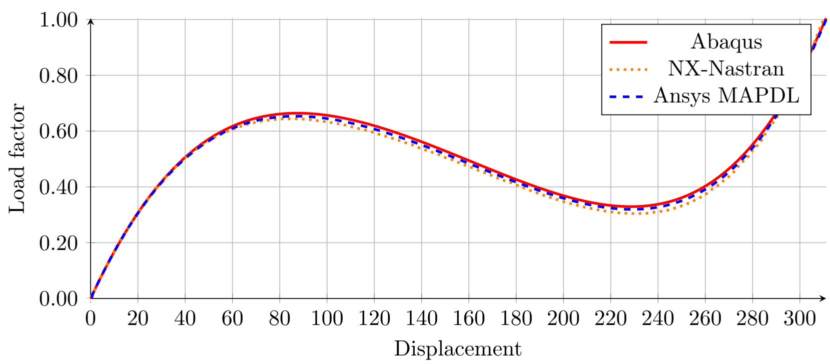
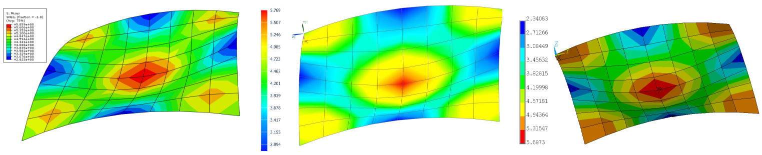


Figure 11: Test case B - load factor vs. displacement at center node.

5.2. Test Case B

As the graph in figure 11 shows, the three solvers match well. The maximum relative difference in load factor is around 8% (near displacement = 230). Both Abaqus and NX used 108 increments, while Ansys used 88. While not proven, one of the reason for the different number of increments, could be explained by which method each solver uses for the arc-length method. Abaqus



(a) Abaqus (b) NX Nastran (c) Ansys

Figure 12: Test case B - Von Mises stress on bottom surface

used the Modified-Riks method and Ansys uses the Crissfield's method. These methods are not optional. NX may use both methods, and since the model was originally created in Abaqus, the AP209 converter sets the NX model to use the Modified Riks method.
The difference could also be caused by very specific iteration parameters that are not covered by the converter, or just by internal differences in the solvers.

5.3. Test Case C

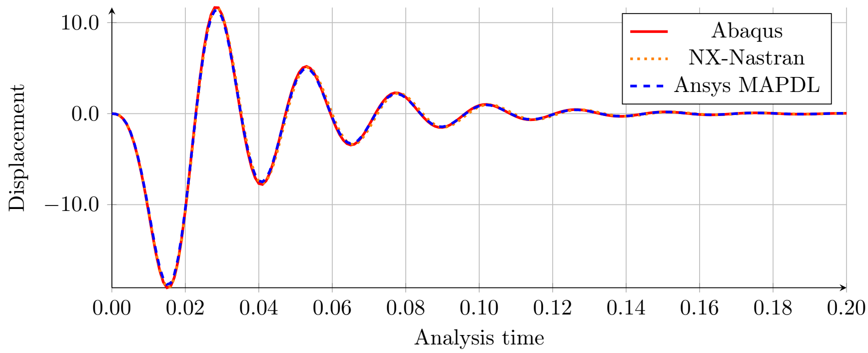
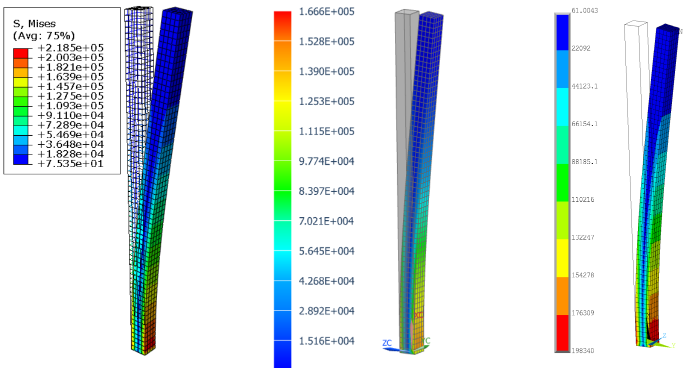


Figure 13: Test case C - displacement vs. analysis time

The graph in Figure 13 shows that the displacements of all solver match very well. The maximum relative differences between the solvers, around the peaks, are around 3-4%. The peaks of the Abaqus and Ansys results have a maximum relative difference of around 2-3%.
The three solver all used the same number of increments to solve the problem.

6. Conclusion and future work

Nonlinear FEA is a very complex domain and supporting every aspect of it in one standard would be very complicated. In this study we have taken some of these aspects and shown how they may be introduced to the AP209 standard. For representing analyses parameters, a generic approach was taken, which will make it easier to represent additional solver specific parameters. Additional



(a) Abaqus (b) NX Nastran (c) Ansys

Figure 14: Test case C - Von Mises stress

analysis types were implemented as a hierarchical class structure covering most of the available analyses in common FEA solvers. Time varying loads were implemented without making changes to the standard, but by relating existing features in the standard.

Using the implementations, test analyses were converted between different FEA formats and solved. The results presented in section 3.1 shows that, for each test case, results were, to a certain degree, similar among the solvers. As such, we have shown that the information was successfully stored in the AP209 data model and converted to the target formats.

This study only shows the possible applicability of nonlinear analysis in AP209, but to be officially part of the standard, an extension of the standard need to be proposed to the ISO STEP 10303 committee, and updated AP209 Recommended Practices need to be published.

At the time of writing, a PDES, Inc. project, ran by, among others, Jotne, Boeing, Lockheed Martin and NIST has started the finalization of AP209 edition 3, and preparing an official white paper for presenting content to be included in AP209 edition 4. Currently nonlinear FEA is one of the main topics to be added in AP209 edition 4. This study may serve as a starting point for the white paper.

References

[1] ISO 10303-1:1994. Industrial automation systems and integration – Product data representation and exchange – Part 1: Overview and fundamental principles. Standard, International Organization for Standardization, Geneva (Switzerland), 1994.

[2] ISO 10303-209:2014. Industrial automation systems and integration – Product data representation and exchange – Part 209: Application protocol: Multidisciplinary analysis and design. Standard, International Organization for Standardization, Geneva (Switzerland), 2014.

[3] R. Lanza, J. Haenisch, K. Bengtsson, and T. Rølvåg. ISO 10303 AP209 - Why and how to embed nonlinear FEA. under Review in Advances in Engineering Software, 2020.

[4] R. Lanza, J. Haenisch, K. Bengtsson, and T. Rølvåg. Relating structural test and fea data with step ap209. *Advances in Engineering Software*, 127:96 – 105, 2019.

[5] T. Kramer and X. Xu. *STEP In a nutshell, in: Advanced design and manufacturing based on STEP*. Springer - Verlag London Ltd., 2009.

[6] Step application handbook iso 10303 version 3. Technical report, SCRA, North Charleston, South Carolina, USA, 2006.

[7] 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. Standard, International Organization for Standardization, Geneva (Switzerland), 2011.

[8] 10303-214:2010. Industrial automation systems and integration – Product data representation and exchange – Part 214: Application protocol: Core data for automotive mechanical design processes. Standard, International Organization for Standardization, Geneva (Switzerland), 2010.

[9] ISO 10303-242:2014. Industrial automation systems and integration – Product data representation and exchange – Part 242: Application protocol: Managed model-based 3D engineering. Standard, International Organization for Standardization, Geneva (Switzerland), 2014.

[10] ISO 10303-11:2004. Industrial automation systems and integration – Product data representation and exchange – Part 11: Description methods: The EXPRESS language reference manual. Standard, International Organization for Standardization, Geneva (Switzerland), 2004.

[11] Keith A. Hunten. Cad/fea integration with step ap209 technology and implementation. *MSC Aerospace Users Conference Proceedings*, 1997.

[12] N. Pitre K.A. Hunten, J.W. Klintworth and T.E. Mack. New standards based data exchange bridge for design (cad), analysis (cae) and manufacturing (cam) of composite structures. In *MSC 1999 Aerospace Users Conferance Proceedings*, 1999.

[13] Edward L. Stanton, Takman Mack, Hiren D. Patel, and Jamie Klintworth. Composite beam models using iso step ap209. 2003.

[14] Keith A. Hunten, Allison Barnard Feeney, and Vijay Srinivasan. Recent advances in sharing standardized step composite structure design and manufacturing information. *Computer-Aided Design*, 45(10):1215 – 1221, 2013.

[15] Eric Lebègue, Georg Siebes, and Charles Stroom. Thermal analysis data exchange between esa and nasa with step. 07 1999.

[16] Ma Singva. Step standardâs evaluation for modeling in electromagnetism. *COMPEL - The international journal for computation and mathematics in electrical and electronic engineering*, 18(3):311–323, March 2020.

[17] Sébastien Charles and Benoit Eynard. Integration of cad and fea data in a pdm environment: Specification of a step simulation data management schema. *17th IMACS World Congress*, 02 2005.

[18] Peter Bartholomew and Christian Paleczny. Standardization of the finite element analysis data-exchange in aeronautics concurrent engineering. *Journal of Computing and Information Science in Engineering - JCISE*, 5, 03 2005.

[19] John Michopoulos, R Badaliance, T Chwastyk, L Gause, and P Mast. Femml for data exchange between fea codes. 01 2001.

[20] Byoung-Keon Park and Jay J. Kim. A sharable format for multidisciplinary finite element analysis data. *Computer-Aided Design*, 44(7):626 – 636, 2012.

[21] EXPRESS Data Manager™ Software Development Kits. http://www.jotneit.no/products/express-data-manager-edm, 2019. Accessed on 14 February 2020.

[22] Jotne EPM Technology AS. http://www.jotneit.no. Accessed on 14 February 2020.

[23] STEP Tools Software. https://steptools.github.io/. Accessed on 02 March 2020.

[24] STEP Tools, Inc. https://steptools.com. Accessed on 02 March 2020.

[25] STEPcode. http://stepcode.github.io/. Accessed on 02 March 2020.

[26] NIST (National Institute of Standards and Technology, U.S. Department of Commerce. https://www.nist.gov/. Accessed on 14 February 2020.

[27] JSDAI™. https://jsdai.net/. Accessed on 02 March 2020.

[28] LKSoftWare GmbH. https://www.lksoft.com/. Accessed on 14 February 2020.

[29] Abaqus (version 6.14). https://www.3ds.com/products-services/simulia/products/abaqus/abaqusstandard/, 2014.

[30] NX (version 11.0). https://www.plm.automation.siemens.com/global/en/products/nx/, 2016.

[31] Ansys Mechanical (version 19.0). https://www.ansys.com/products/structures/ansys-mechanical-enterprise, 2018.

[32] Siemens Product Lifecycle Management Software Inc. *NX Nastran 11 Verification Manual*. Siemens Product Lifecycle Management Software Inc., 2016.

[33] CAx Implementor FOrum (CAx-IF). https://www.cax-if.org. Accessed on 14 February 2020.

[34] ISO 10303-41:2000. Industrial automation systems and integration – Product data representation and exchange – Part 41: Integrated generic resource: Fundamentals of product description and support. Standard, International Organization for Standardization, Geneva (Switzerland), 2000.

[35] ISO 10303-43:2011. Industrial automation systems and integration – Product data representation and exchange – Part 43: Integrated generic resource: Representation structures. Standard, International Organization for Standardization, Geneva (Switzerland), 2011.

[36] ISO 10303-104:2000. Industrial automation systems and integration – Product data representation and exchange – Part 104: Integrated application resource: Finite element analysis. Standard, International Organization for Standardization, Geneva (Switzerland), 2000.

[37] ISO 10303-50:2002. Industrial automation systems and integration – Product data representation and exchange – Part 50: Integrated generic resource: Mathematical constructs. Standard, International Organization for Standardization, Geneva (Switzerland), 2002.

[38] ISO 10303-107:2006. Industrial automation systems and integration – Product data representation and exchange – Part 107: Integrated application resource – Finite element analysis definition relationships. Standard, International Organization for Standardization, Geneva (Switzerland), 2006.

Appendix B

Secondary publication

B.1 Paper 4 - Open Simulation Data Management and Testing - The CRYSTAL Project

Open Simulation Data Management and Testing - The CRYSTAL Project

Kjell Bengtsson, Jochen Haenisch, Olav Liestøl, Remi Lanza
(Jotne, Norway);

Abstract

Structural testing is well integrated in the development processes of complex systems such as aircraft, however, during execution and especially after completion of these processes, finding information is inefficient and time-consuming with data spread over many applications, files, formats and locations. Lockheed Martin Aeronautics and Jotne, explore and implement a system architecture for central storage of design, simulation and test data that enables interoperability and long-term archival and retrieval (LOTAR) through configuration control. This architecture is based on the international standard STEP, ISO 10303. STEP is already widely used for sharing of 3D models for design by file exchange. CRYSTAL (CollaboRative multi-disciplinarY design, analysiS and Test dAta management and correLation) is extending the use of this standard in the engineering analysis and simulation domain to include FEM/CFD/DEM, and adds methods of implementing it by providing access to a central product data repository through APIs and webservices. Even beyond these existing capabilities of STEP, CRYSTAL will take the lead of a standardization activity to enlarge the STEP lifecycle model to integrate also structural testing and its relationship to design and simulation, especially in the context of validating the results of those. The benefit that such a comprehensive product data repository can give to industry, however, depends on the interoperability of their current and future tools with the standard and, through the standard, with other tools. CRYSTAL will, therefore, deliver a tool kit to lower the threshold for vendors to link their corresponding domain applications into this architecture.

In addition, this presentation will also highlight results from other R&D programs, like IDEaliSM (Integrated & Distributed Engineering Services framework for MDO, which is an ITEA initiative and part of the EUREKA Cluster program) and CAxMan (Computer Aided Technologies for Additive Manufacturing, part of the H2020 Research and Innovation Action) and how they are deploying the new standardized infrastructure.

1. Engineering data management use cases

Structural testing is well integrated in the development processes of complex systems such as aircraft. However, during execution and especially after completion of these processes, finding information is inefficient and time-consuming with data spread over many applications, files, formats and locations.

Lockheed Martin Aeronautics Company (LM Aero) and Jotne EPM Technology AS (Jotne) are pursuing together with the Norwegian Ministry of Defense (NMoD) the enhancement and deployment of an open standards-based advanced system for CollaboRative multi-disciplinarY design, analysiS and Test dAta management and correLation (CRYSTAL).

The following sub-chapter describes the underlying use cases.

1.1. Multidisciplinary analysis environments

The use case in Figure 1 represents a scenario for open simulation data management of multidisciplinary aero/thermal/structural analyses (CRYSTAL). Some examples of these types of analyses are fluid structural interaction problems such as wind, wave and current pressure on oil rig support structure and pressure and thermal loads on an aircraft.

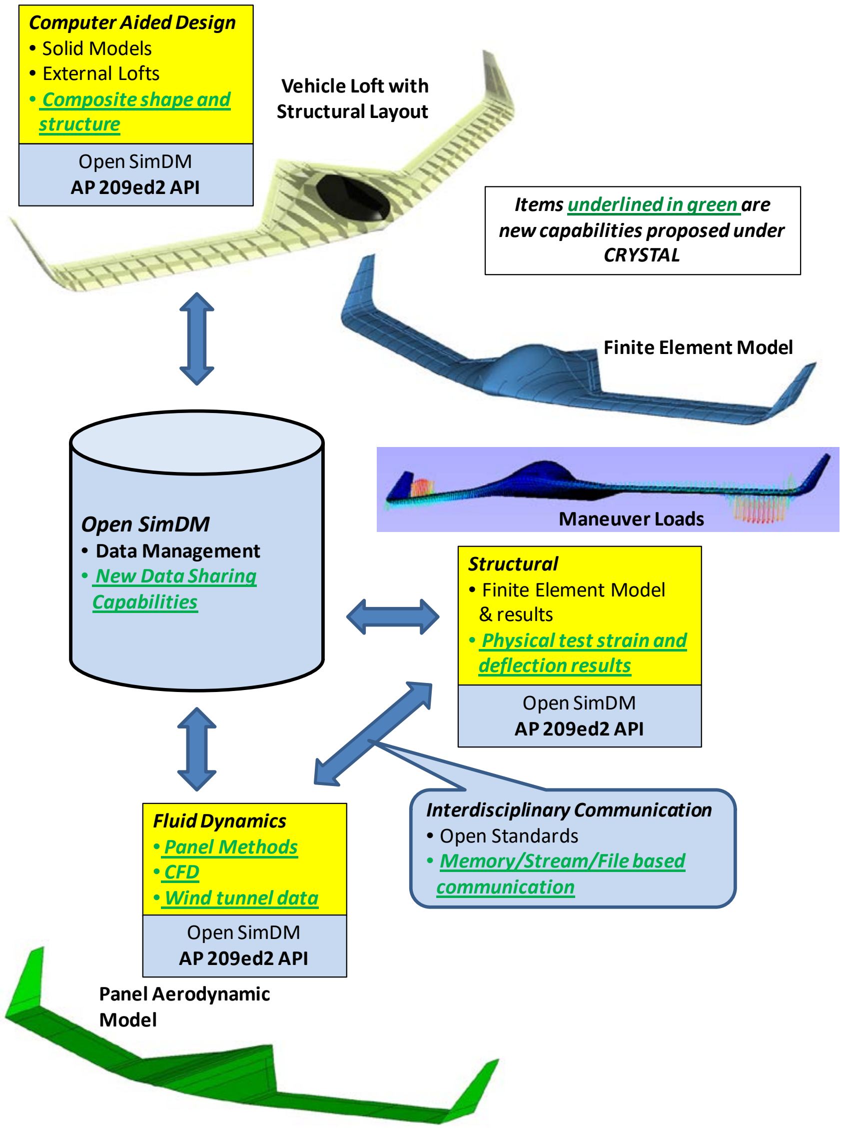


Figure 1: Use Case Scenario for Multidisciplinary Aero/Structural Analyses.

The engineering process that is supported by this scenario may follow the stages listed below:

- An initial concept of a product is designed in a CAD system.

- The created product shape and product structure is exported from the authoring CAD system to CAE-system(s) via an AP209ed2 file. AP209ed2 is a superset of AP242 and thus may contain a complete design shape representation. The unique capabilities of the AP209ed2 related design and CAE product structures may be used to organize the CAE model(s) to aid in managing complex configurations. Alternatively, some CAE systems provide design capabilities directly.
- The CAE-systems for structural, aerodynamic modelling and heat transfer analyses may, via the AP209ed2 API, request data from an Open SDM repository that other players (other projects, suppliers etc.) may have stored there. Information is shared through API calls to an Open SDM database repository or to shared memory (both in AP209ed2 schema based formats).
- Communication between disciplines in the combined analyses will consist of the information representing each of the disciplines but not the process or algorithms involved in the mapping between the disciplines. An example is pressure fields over a fluid dynamics mesh model are represented over that mesh, and are related to a structural mesh model (and potentially the pressure field over the structural mesh model). The mapping between the two pressure fields that ensures that the pressure fields are equivalent over the two different meshes is NOT represented, only the relationship between the two fields is established.
- Once complete the CAE analyses reside in the AP209ed2-based Open SDM repository. The AP209ed2 schema provides the capability to represent the associated pedigrees providing traceability and associativity between the various disciplines’ analyses and the original CAD shape. Analysis feedback may require that the product shape is returned to CAD.
- Intermediate or final results may be viewed through a web-based graphical display.
- Final results may be delivered in an AP209ed2 archive repository to customers to enable, for example, support and repair tasks.

1.2. Correlation of CAE output and physical test results

Figure 2 represents a use case scenario for open simulation data management involving correlation or linking of CAE output and physical test results for the purpose of validating the analysis results against reality.

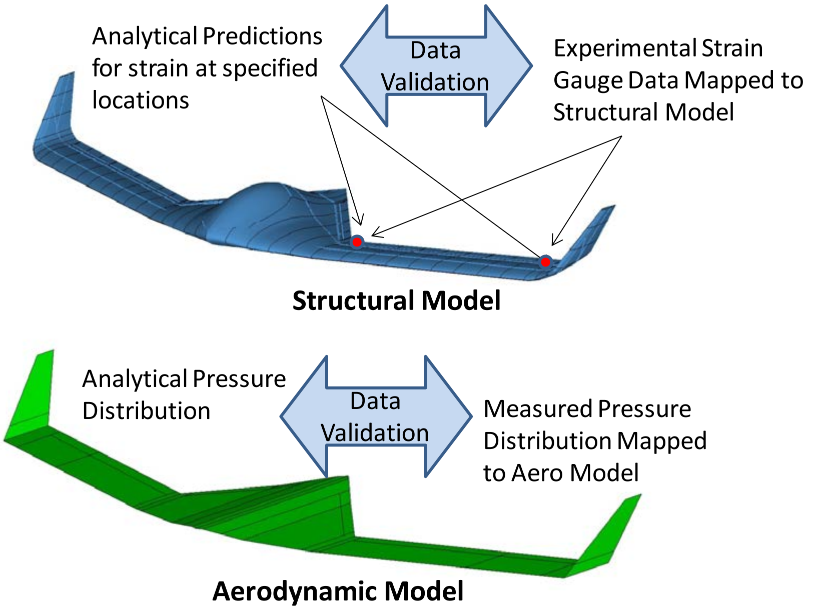


Figure 2: Use Case Scenario for Validation of Aero/Structural Analyses with Observed Test Data by Data Correlation.

The complex multi-disciplinary simulations required for aerospace design and analysis, though sophisticated, are still approximations. Many structural, aerodynamic and thermal analyses are required to be correlated/validated with actual observed testing results (correlation in that a given input produces an associated output; and validation in that an output matches or is equivalent to a reference output). This is especially true for analytical results used in the certification process of flight vehicles.

A typical scenario may be the comparison of measured structural strain and deflection with computed Finite Element analyses results and the comparison of wind tunnel pressure loads to aerodynamic simulation results and the subsequent use of derived inputs to those Finite Element analyses. These loads may also be used as a basis for the physical structural testing. An example of a large scale vehicle testing program is shown in Figure 3.

Figure 3: F-35 in a testing frame.

In another scenario large scale test environments themselves are correlated/validated with the products under test, not only during physical structure testing, but also during simulation. Such virtual testing tools have the potential to considerably reduce expenses with disruptive testing of complex products.

2. The CRYSTAL vision

Lockheed Martin Aeronautics and Jotne, explore and implement a system architecture for central storage of design, simulation and test data that enables interoperability and long-term archival and retrieval (LOTAR) through configuration control.

The objective of the CRYSTAL project is to develop a framework for the sharing and management of simulation data management (SDM), exposing design, engineering analysis and structural testing data in a form that it can be reused by multiple down-stream functions in order to streamline processes and improve efficiencies. CRYSTAL will allow integration of Product Data Information into the extended enterprise that operates to deliver complex systems within a coherent framework of engineering and manufacturing processes, including the interaction with Small and Medium sized Enterprises (SMEs).

The project is based on the use of the formal data specification and manipulation language EXPRESS (ISO 10303-11 and ISO 10303-14). Specifications written in EXPRESS are machine interpretable, such as ISO AP209ed2, the data specification for this project. EXPRESS enables database dictionaries to be generated automatically and the validity of data sets to be checked automatically. EXPRESS also allows the specification of additional rules for how to populate a data model. A data set may be in accordance to a data model, but may be invalid under a certain rule schema that is defined on top of the data model.

Based on the use case definition data, above, conversion and sharing software will be developed based upon the AP209ed2 schema code for sharing information between finite element analysis, panel aerodynamics, CFD tools, and structural testing applications. The CFD tool interfaces will be based upon layering the AIAA CGNS API standard over the AP209ed2 API.

In addition, three external CAE tools will be involved to implement the resulting capabilities for sharing and managing information in their toolsets.

With the input and output data to multidisciplinary computations cast into 10303-209ed2 open standards based format, the information can also be used for downstream delivery in product lifecycle processes.

Such data are also candidates for long term retention, for example, to support design, reuse or liability proof. The CRYSTAL archival application will be maintained and enhanced to:

- verify the quality of AP209ed2 data sets;
- enable the storage of the quality assessment, of the meta data with CAX input models and of the data themselves.

Whereas the correctness of the data can be assessed automatically by functions of EDM, the usefulness of the data cannot. An additional validation mechanism is needed that enables the user to browse through the data in their AP209ed2 representation. Also this display and check capability will be delivered.

The completed solution will be tested using typical LM Aero processes and test data, along with a set of openly sharable (ITAR-free) data.

2.1. Envisaged engineering process

The engineering process that is supported by these scenarios may follow the stages listed below, which are also shown in Figure 4:

- The CRYSTAL platform enables querying for information required to support a particular analysis. The information obtained gives insight to the existence, pedigree and applicability of the required data.
- In the case of Finite Element analysis of flight vehicles, aerodynamic loads and load cases are developed to meet the stated system requirements. These loads are mapped to the AP209ed2 data model and become a part of the product information model. This data can be from CFD analysis, wind tunnel testing, actual flight testing or combinations of all three.
- Mapping these loads onto structural FEA models is facilitated by application of tools that understand the underlying standardized information model. Multi-disciplinary methods discussed in the following section are often limited by the ability to map data efficiently across domains.
- Examples include un-ambiguous definitions of field data locations, orientations, approximations, interpretations and assumptions used in the original data generation and the mapping processes to other domains. Even a common descriptive vocabulary facilitates data sharing.
- Analyses are performed using these loads and stored in the CRYSTAL platform.
- In the example of structural testing, purpose built analysis models of the test article and the test environment can be retrieved into CAE systems along with other relevant data sources. These models are used to facilitate detailed planning for physical tests, sharing information in an un-ambiguous fashion.
- Structural predictions can be shared with the testing environment to validate instrumentation installation, linearity assumptions, and to improve monitoring for safety.
- Additional post processing of strain and deflection results from the physical structural testing are mapped to the structural FEA model and stored in AP209ed2 format. Again, checking the strain and deflection results for consistency with respect to coordinate spaces, orientations and assumptions.
- Intermediate or final results are managed in the CRYSTAL platform in AP209ed2 format providing associativity between the simulation and test results with pedigree and descriptive metadata.
- Final results may be delivered in an AP209ed2 archive repository to customers to enable support and repair tasks.

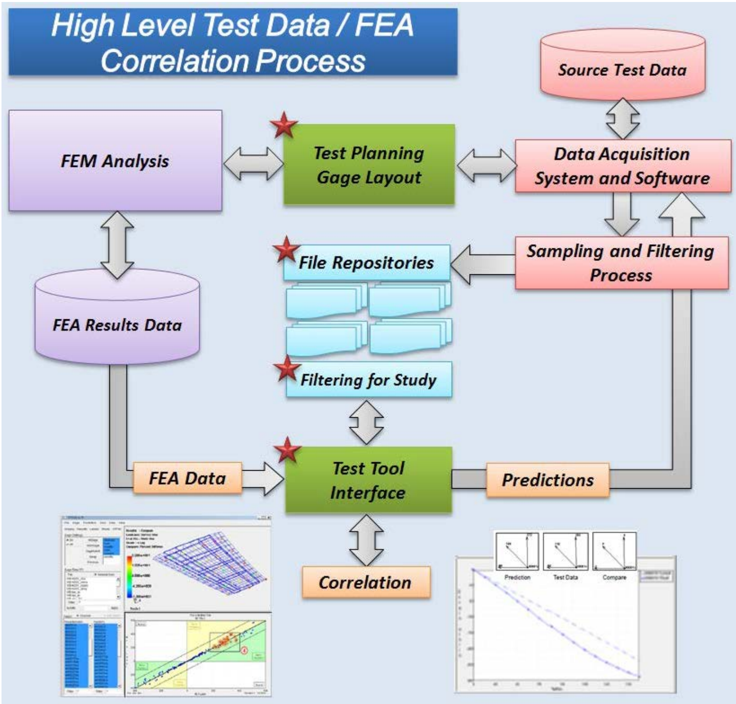


Figure 4: Gaging/Prediction/Correlation Process.

2.2. Multi-disciplinary integration

There are different types of Multi-disciplinary Integration (cited from reference #2):

1) “Serial Integration
   a. Applicable to coupled but separable physics
      i. Product of first analysis is input to second analysis
   b. Examples
      i. Precision structure borne optical systems
      ii. Thermally driven structure fatigue life prediction
      iii. Quasi static analysis where one domain is quite fast relative to the other.
   c. Implementation approach
      i. Use stand alone solver for first domain
      ii. Transfer output solution to input for second solver

2) Loosely Coupled
    a. Cyclically coupled where the output of one solution drives the input to the other, but that response affects the first domain.
        i. Not separable and serially solvable
    b. Amenable to iterative, cyclic solution
        i. Assume an input and run the sequence to compute an updated starting point
        ii. Requires strong/good enough convergence
    c. Examples
        i. Deformed aircraft wing shape under static aerodynamic loads
3) Integration Frameworks
    a. Framework for integration of a wide variety of existing analysis tools
    b. Commonly applied to optimization of performance
    c. Captures and automates an analysis process
    d. Seeks to leverage existing tools as black boxes
        i. Executed via invocation commands
        ii. Manipulate parameters exposed in input decks and commands
    e. Supported by several commercial products
4) CoSimulation
    a. For inseparable domains that require time synchronous solution
    b. Each solver can run independently for a few (but different) steps, but meet regularly to synchronize
    c. Requires transfer of variables/response between domains at synchronization points
    d. May require iteration for equilibrium at synchronization points

For realistic complex problems (beyond serially separable and optimization), process knowledge is embedded in the solver executable. Model data definition is not sufficient. Case control, execution parameters are not sufficient. For example, successful CFD requires complex numerical processes that are sensitive to the implementation expression. As a result, co-simulation is required for the current multidisciplinary analyses of interest."

It is envisioned that the standards for data exchange already developed by the ISO-STEP community, and additional data communication capabilities developed under CRYSTAL will benefit approaches 1 through 3 listed above.

Integration of new capabilities with approach 3 will be the focus in CRYSTAL, as process improvements in this area are natural enablers for approaches 1 & 2.

2.3. AP209ed2 converter strategy

To ease AP209ed2 converter development is the most important means of implementing a standards-based interoperability solution in industry for engineering analysis. CRYSTAL, therefore, aims at providing a both powerful and easy-to-use converter development kit.

Data translation processes can have different types of “databases” for source and target datasets:

- Both source and target are Express models;
- Source or target is a non-Express model, random accessible;
- Source or target is a sequential file.

The AP209ed2 converter development kit is designed to handle AP209ed2 datasets as either source or target. The same applies for random accessible non-Express models and sequential files. For non-Express models, both random accessible or sequential, the converter development kit can be used to integrate AP209ed2 export/import directly into solvers to become their AP209 pre- and post-processors.

For Express data models the converter development kit will generate C++ classes to provide to the programmer a familiar environment and an easy-to-use API.

When both source and target datasets are random accessible, it is possible to implement a recursive conversion algorithm. When converting objects from an object oriented model like AP209ed2, this is big advantage. This implementation pattern requires that each object class that shall be converted has its own converter method. When reaching an attribute that references another object during execution of such a method, a call to the conversion method of the referenced object will be used to update the attribute.

When the source dataset is a sequential file the basic converter algorithm applies a three step procedure where the first step is to read data from the source file into computer memory, before converting the data and then writing the target data set to a file. When all data of the input file is stored in main memory, the conversion to AP209ed2 starts. This strategy has a drawback: There is an upper limit of the size of the input file that a computer can read, and this will in many cases be reached. This problem shall be solved in CRYSTAL. The solution is to:

1. Define the intermediate data format in EXPRESS instead in C++ and store the intermediate data in a random accessible EXPRESS database (EXPRESS Data Manager).
2. Enhance the AP209ed2 API to store objects in the database as soon as they are created and remove them from memory.

Working with FEM data is time consuming due to big data sets and heavy computational requirements. By introducing concurrent processing for both data validation and data conversion one can reduce execution time. Modern computers have several CPUs making concurrent processing possible. The CRYSTAL project will explore these possibilities of parallel FEM data validation and conversion.

The CRYSTAL project shall also enhance the formal procedure of specifying, designing, implementing and testing data converters. Mapping between objects in source and target models can be specified in an EXPRESS model on entity data type level. These specifications are used to generate skeletons of mapping code. In CRYSTAL the mapping specification EXPRESS model shall be enhanced to also specify mapping on attribute level. By this the amount of automatically generated mapping code skeletons will be increased.

Each data converter needs to be verified for its correctness. One of the methods that this project will elaborate on especially is the use of validation properties. The concept itself is promoted by the LOTAR initiative as one of the crucial features in engineering data archiving. Whereas validation properties are well established for product shape, for example, centre of gravity and cloud of points, agreements still need to be found within domains, such as, composites, FEM and CFD. The project will explore this together with LOTAR; see chapter 3.

A good introduction to the use of validation properties can be found on https://cax-if.org/ and in their recommended practices document “rec_prac_gvp_v44.pdf” (see reference #1).

2.4. Validations across design, analyses and testing

A data correlation/validation tool is being developed to import and link structural test results data with analysis data. The use of the tool can be described in a step by step manner, of which the main steps are described below. The prerequisites are to have a valid AP209ed2 FEM analysis model and the completed structural test of the corresponding product.

1) **Structural test data import**
   - The data is imported from the testing tool's native format. This will involve multiple files, if multiple load cases have been carried out.
   - The results will be viewable and verifiable for all sensors and test cases.
2) **Sensor Descriptions**
   - If sensor descriptions are available from the imported file, they can be imported and verified; otherwise, descriptions will be added manually.
   - Sensor locations are defined relative to reference coordinate systems.
3) **Loading Descriptions**
   - Each imported data set is linked to a load case (e.g. force magnitude and direction, vibration intensity…).
4) **Create Raw Data Model**
   - With the above data in place, an AP209ed2 model of the raw data can be created. This will hold the results without any transformation, together with sensor and load descriptions and product data.
5) **Create Test Correlation/Validation Model**
   - The AP209ed2 model of the FEM analysis can then be imported, and a global coordinate system between the two models can be established.
   - The raw test results will need to be mapped to corresponding result types in the FEM model (e.g. strain in direction x on element surface).
   - A link between the test cases and the load cases in the analysis need to be defined.
6) **Validation**
   - With the coordinate systems defined, and corresponding results mapped to one another, the raw results can be converted and displayed. A dual display showing analysis and test results, and the possibility of creating comparison graphs, reports and other

post-processing utilities that enable the validation are envisioned.

The above description is valid for simple loading tests, which is the initial focus of the project, while other test types (e.g. modal/frequency, thermal and aerodynamics tests) would be included in the next phase.

The result will be three related models; Analysis Model, Test Correlation/Validation Model, and Raw Test Results Model, all described in the common AP209ed2 format and linked together. They can be imported to the SDM framework and prepared for long-term archival.

CRYSTAL is currently creating representative test data (see figure 5) both on the analysis side and the structural test side. These data will be used to verify the applicability of the CRYSTAL Platform.

Figure 5: CRYSTAL Test Data: an Ultralight Glider Winglet.

This CRYSTAL Platform will be based on a functioning collaboration server for timely distribution of accurate and complete product data, based on the second edition of ISO 10303-209:2014 (AP209ed2), “Multidisciplinary analysis and design”, or possibly a third edition if needed. The commercial product that will be derived from this will provide a capability to integrate and share CAD, CAE and structural test data.

The consolidated database will be complimented by a long term data retention (LTDR) solution based on international standards to meet the requirements for retaining digital product data over long periods of time, typically lifetimes greater than 50 years.

3. Engineering data archival

A motivation and a use case for CRYSTAL is long-term archival of engineering data. Especially in the aeronautics industry the availability of design and analysis data over the lifetime of the aircraft is a pre-condition for a flight or air worthiness certificate. The lifetime of an aircraft will most probably exceed the lifetime of the engineering software and formats by far. Tools and methods are needed that guarantee readability of the data independently of the availability of the original authoring tools.

LOTAR International, or short: LOTAR (LOng Term Archiving and Retrieval), offers such tools and methods.

LOTAR is a project being conducted by leading OEMs and suppliers in the aerospace and defense industry under the joint auspices of ASD-STAN, AIA, PDES Inc. and the ProSTEP iViP Association. The LOTAR project consortium consists of user companies from around the world. Member companies include Airbus, BAE Systems, Boeing, Dassault Aviation, General Dynamics, Goodrich, IAI, Lockheed Martin, SAFRAN, Sandia, and Spirit.

Details are on http://lotar-international.org/ .

From the technical point of view and for the scope of this paper it is important to note the two pillars that LOTAR builds upon, that is, the following two standards:

1) ISO 14721:2012: Reference Model for an Open Archival Information System (OAIS) and
2) ISO 10303: Industrial automation systems and integration — Product data representation and exchange (also known as STEP, Standard for the Exchange of Product model data).

Whereas ISO 14721 describes the process and environment of archival, ISO 10303 provides the product lifecycle data model that allows to collect and store complex product data consistently.

In our endeavours to deliver long-term availability of product data in our use cases, the CRYSTAL project builds upon the LOTAR environment and is in a relationship of mutual give and take with LOTAR.

4. The role of STEP

The CRYSTAL Platform is based on the international standard STEP, ISO 10303. STEP is already widely used for sharing of 3D models for design by file exchange. CRYSTAL is extending the use of this standard in the engineering analysis and simulation domain to include FEM/CFD, and adds methods of implementing it by providing access to a central product data repository through APIs and web services. Even beyond these existing capabilities of STEP, CRYSTAL will take the lead of a standardization activity to enlarge the STEP lifecycle model to integrate also structural testing and its relationship to design and simulation, especially in the context of validating the results of those.

To cope with the heterogeneous nature of the software environment in industry, data specifications have been standardized in ISO/TC 184/SC 4, the ISO sub-committee for "Industrial Data", over the last 30 years. The resulting standards cover the information requirements for a wide range of industries including shipbuilding, oil and gas, process, aerospace, automotive, and built environment.

The ISO standards basis of CRYSTAL shall support both the communication between design/analysis engineers and test engineers and the link between test environments and products under test in a standard AP209ed2 format of the digital data involved in these types of scenarios.

Within the total scope of ISO 10303, the series of Application Protocols provide capabilities in specific engineering domains. The edition 2 of ISO 10303-209 (AP209ed2), Multidisciplinary analysis and design, specifies computer-interpretable composite and metallic structural product definition data representation, such as: shape, idealized analysis shape, finite element analysis (FEA) model, analysis results, and material properties. The design and related analysis information are managed within a PDM product structure.

The scope of AP209ed2 includes not only engineering analysis data, but also the design data that are the starting-point for analyses. The representation of product design data alone is covered by another application protocol, ISO 10303-242 (AP242), Managed Model Based 3D Engineering. AP209ed2 simply includes AP242 in its entirety. A high-level planning diagram that provides an overview of the contents of AP209ed2 is shown in Figure 6, below. The complete scope of AP209ed2 will be used in this project.

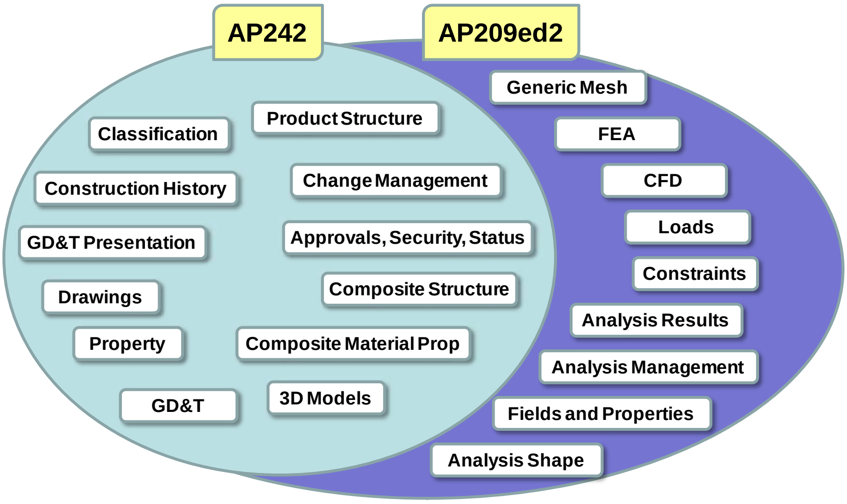


Figure 6: Overview of ISO 10303-209ed2 Contents

AP209ed2 has integrated a generic Engineering Analysis capability complimented by specific CFD and generalized mesh based numerical analysis capabilities to the AP209 Edition 1 classical Finite Element Analysis capabilities. The CFD capability is based upon the NASA/AIAA CGNS standard and the Volvo Aero EAR-model work. The generalized structured and unstructured analysis and mesh capabilities are based upon work done in the Generic Engineering Analysis Model (GEM) project from the European Union. In addition there is a complete discrete/continuous mathematical field representation capability that has been added based upon the David Taylor Labs/Boeing DT-NURBS package.

For the purpose of integrating also structural test data, extensions of AP209 into an edition 3 may be needed and are being considered.

The architecture of the CRYSTAL platform and particularly the AP209ed2 API integrates various disciplines within an organization like LM Aero to enable complex multidisciplinary analyses required for aerospace design and analysis. The ISO standards basis of CRYSTAL supports communication to suppliers and maintainers in a format suitable for long term archival of the digital data involved in this type of scenario.

Copyright Nafems 2020 Licensed solely to Kjell Bengtsson from Jotne EPM Technology AS(kjell.bengtsson@jotne.com) for single use only.
Distribution,reproduction and editing strictly prohibited-www.nafems.org/downloadterms LICENSE ONLY VALID WITH A CURRENT NAFEMS MEMBERSHI

5. Related R&D projects

In addition, this presentation will also highlight results from other R&D programs, like IDEaliSM (Integrated & Distributed Engineering Services framework for MDO, which is an ITEA initiative and part of the EUREKA Cluster program) and CAxMan (Computer Aided Technologies for Additive Manufacturing, part of the H2020 Research and Innovation Action) and how they are deploying the new standardized infrastructure.

5.1. IDEaliSM

The IDEaliSM - Development framework for Multidisciplinary Design and Optimisation - solutions lie in three main deliverables: an advanced integration framework for distributed Multidisciplinary Design and Optimisation, an Engineering Language Workbench (a set of domain specific and high-level modelling languages, ontologies and data standards) and a methodology for service-oriented development processes to redefine the product development process and information architecture to enable collaboration between service oriented Competence Centres in Distributed Development Teams. The project is an ITEA initiative and part of the EUREKA Cluster programme.

A video will be made available that demonstrates how the ISO 10303-209 repository connects to several commercial applications through the use of the NOESIS applications and others like CATIA, PATRAN, NASTRAN and more. For more information about the IDEaliSM project see http://www.idealism.eu/ .

5.2. CAxMan

Another R&D program of a similar standards-based collaboration architecture is CAxMan (H2020-FoF-2015-680448), which is also funded via the European Commission.

The objectives of Computer Aided Technologies for Additive Manufacturing (CAxMan) are to establish Cloud based Toolboxes, Workflows and a One Stop-Shop for CAx-technologies supporting the design, simulation and process planning for additive manufacturing. More specifically the objectives are to establish analysis-based design approaches with the following aims:

- To reduce material usage by 12% through introducing internal cavities and voids, whilst maintaining component properties;
- To optimize distribution and grading of material for multi-material additive manufacturing processes; and
- To facilitate the manufacture of components which are currently impossible or very difficult to produce by subtractive processes (e.g., cutting and abrasive operations);

- To enhance analysis-based process planning for additive manufacturing including thermal and stress aspects, and their interoperability with the design phase;
- To enable the compatibility of additive and subtractive processes in production in order to combine the flexibility of shape in additive manufacturing with the surface finish of subtractive processes.

Also in this project interoperability is achieved through the application of mainly the following international standards:

- ASD/AIA LOTAR solutions for archival of digital engineering data based on ISO 14721.
- ISO 10303 (STEP) *Product data representation and exchange* in particular ISO 10303-209 edition 2 (AP209ed2) *Multidisciplinary analysis and design*.

6. References

1. Jochen Boy, Phil Rosche (2016). *Recommended Practices for Geometric and Assembly Validation Properties, Release 4.4*: CAx Implementor Forum. 31 pages.
2. Clark Briggs (2012). *Multi-Disciplinary Mechanical Modeling, Management and Exchange*: ATA Engineering.

Appendix C

Contact and Nonlinear Materials in STEP

The following presents a topic which was originally intended as a paper, but is instead included in this theses as its own appendix.

Paper 3, which implements some of the proposed AP209 extensions recommended from Paper 2, covers nonlinear analysis and analysis parameters. This appendix, further recommends extensions to the AP209 standard, for covering FEM contact interactions, and certain nonlinear material models.

Section C.1 presents a set of finite element test cases using contact interaction and nonlinear material models. The converter mentioned in 3.1.1 was extended to cover the AP209 extensions, and is used to convert the test cases to other formats. The original models were created in Abaqus, then converted to AP209, and translated to Nastran and Ansys formats. All models were solved in their respective solvers. A detailed discussion on the results of these analyses is not included, however, visualizations of the results are presented.

Section C.2, and section C.3, covers some details on the contact and nonlinear material implementations.

C.1 Test Cases

Test Case A

The first test case is based on a NAFEMS benchmark test case, *Benchmark 2: 3D Punch (Rounded Edges)* described in [56].

- **Geometry**
  - Top cylinder diameter: 100mm
  - Top cylinder height: 100mm
  - Top cylinder fillet radius: 10mm
  - Bottom cylinder diameter: 200mm
  - Bottom cylinder height: 200mm
- **Mesh**
  - Using symmetry, only a quarter is meshed
  - Top cylinder: 1960 elements
  - Bottom cylinder: 16100 elements
  - All elements are of type linear hexahedral (Abaqus: `C3D8R`)
- **Time step incrementation**
  - Automatic time stepping
  - Max incr.: 100
  - Initial incr. size: 0.01
  - Min. incr. size: $1.0E-5$
  - Max. incr. size: 0.01

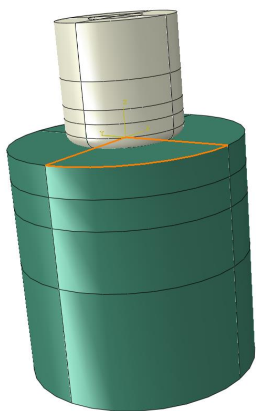


Figure C.1: Test case A geometry

- **Material**
  - Top cylinder: $E = 210GPa$ and $\nu = 0.3$
  - Bottom cylinder: $E = 70GPa$ and $\nu = 0.3$
- **Contact interaction**
  - Contact defined on element surfaces, between bottom face of top cylinder, and top face of bottom cylinder.
  - Penalty friction formulation method with friction coefficient $= 0.1$ and elastic slip $= 0.005$
- **Boundary conditions and load**
  - Bottom surface of bottom cylinder has fixed translation.
  - Constraints to simulate symmetry.
  - Distributed pressure load on the top surface of the top cylinder with 100MPa.

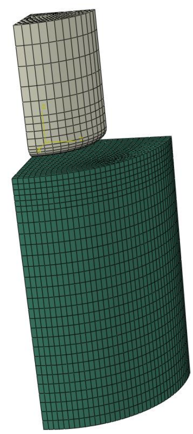


Figure C.2: Test case A mesh

Figure C.3 shows the resulting deformation for the different solvers at the last time step. The graphs in Figure C.4 shows the Z-displacement on the nodes along a path going from the center of the top surface of the bottom cylinder, to the outer edge of the cylinder.

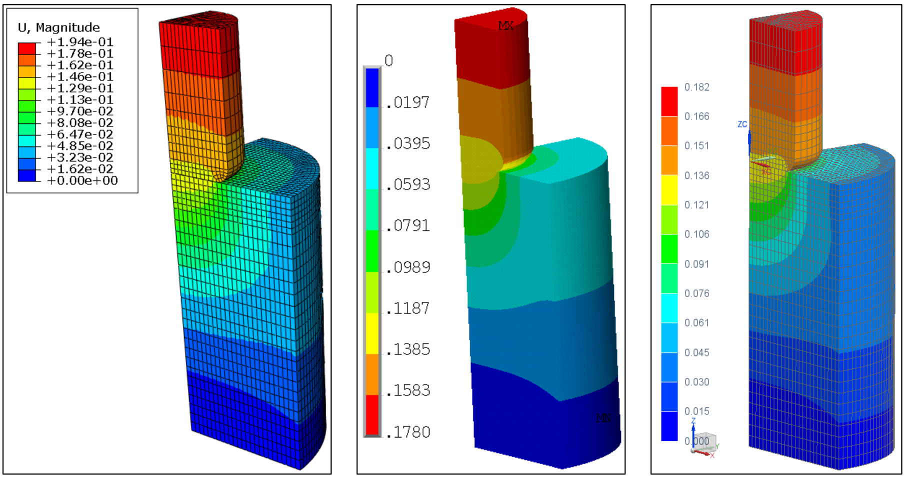


Figure C.3: Test case A displacements visualization (Abaqus / Ansys / Nastran)

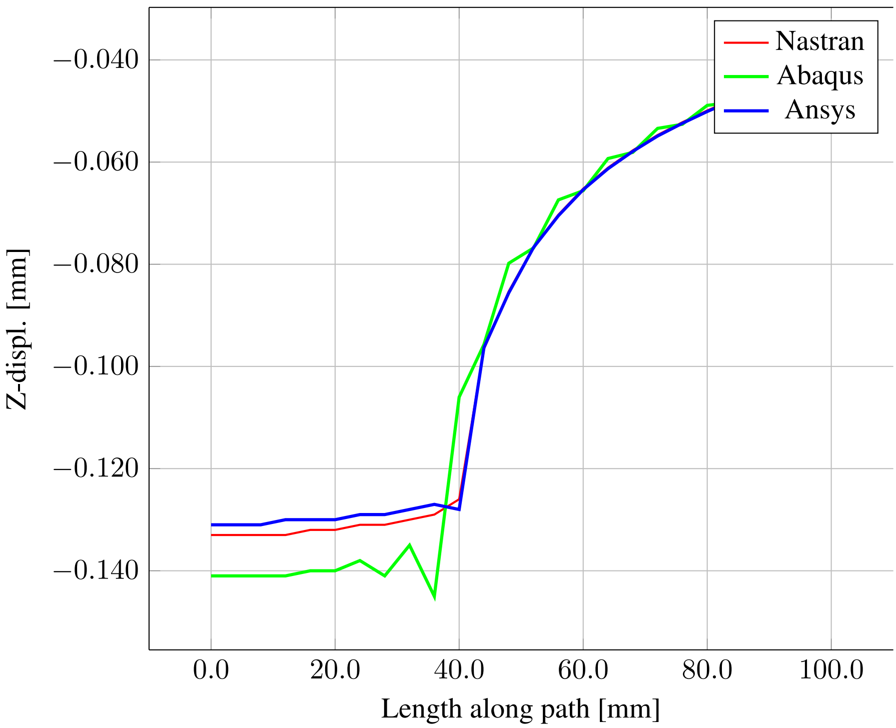


Figure C.4: Z-displacement, along path on top of bottom cylinder, from center to edge

Test Case B

This test case is also based on a NAFEMS benchmark test case, *the three bar problem* described in [57].

- **Geometry**
  - Three bars with dimensions 10x10x100 (mm)
  - Center to center spacing = $30mm$
- **Mesh**
  - Each bar composed of 80 elements (2x2x20)
  - All elements are of type linear hexahedral (Abaqus: `C3D8R`)
- **Time step incrementation**
  - Automatic time stepping
  - Max incr.: 100
  - Initial incr. size: 0.01
  - Min. incr. size: $1.0E-5$
  - Max. incr. size: 0.01
- **Material**
  - All bars: $E = 210GPa$ and $\nu = 0.3$
  - Bar left: $\sigma_{yield} = 200MPa$
  - Bar center: $\sigma_{yield} = 300MPa$
  - Bar right: $\sigma_{yield} = 400MPa$
- **Boundary conditions and load**
  - Nodes on top faces of bars are fixed in all translation degrees of freedom.
  - Nodes on bottom faces of bars connected to reference point with linear equations.
  - Forced displacement downwards on reference point from 0 to $0.275mm$

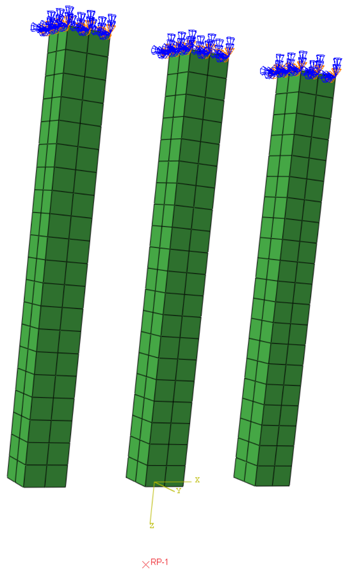


Figure C.5: Test case B

Solving the analyses, results in the first bar reaching yield at 200MPa, second at 300MPa, and last at 400MPa. All three solvers agree on these results.

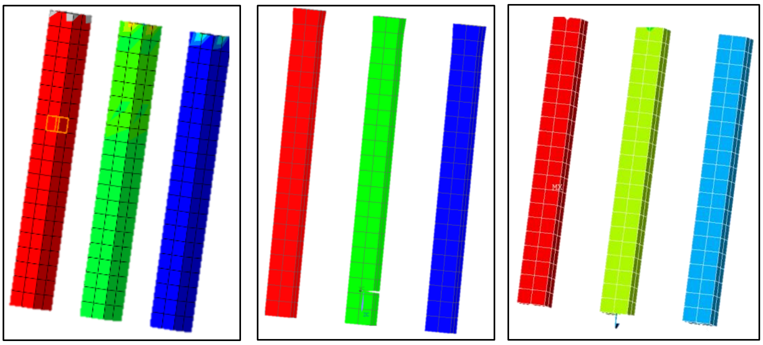


Figure C.6: Test case B Von-Misses visualization, red = 400MPa, green = 300MPa, blue = 200MPa (Nastran / Abaqus / Ansys)

Test Case C

- **Mesh**
  1000 shell elements of type S4R with thickness 0.175mm.
- **Material**
  - $E = 50GPa$ and $\nu = 0.3$
  - Multilinear plasticity model, as seen in Figure C.8

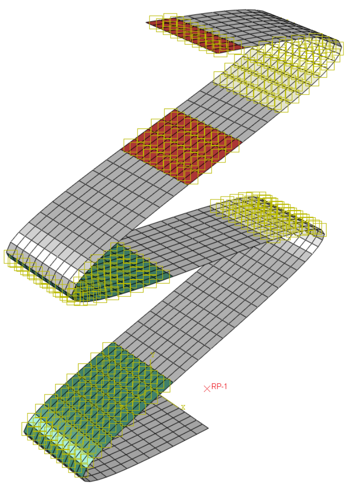


Figure C.7: Test case C

- **Time step incrementation**
  - Automatic time stepping
  - Max incr.: 400
  - Initial incr. size: 0.01
  - Min. incr. size: 10^{-5}
  - Max. incr. size: 0.1
- **Boundary conditions and load**
  - Nodes on one long side is fixed in X.
  - Nodes on one short side is fixed all dofs.
  - Forced displacement along one short side in Y; from $0.0mm$ to $-17.5mm$ to $17.5mm$ to $-17.5mm$ and back to $0.0mm$.
- **Contact interactions**
  - Three surface regions (as seen with colors on Figure C.7) have self contact defined.
  - Penalty friction formulation method with friction coefficient = 0.1 and elastic slip = 0.005

Figure C.9 shows a visualization of the resulting deformations in the three solvers.

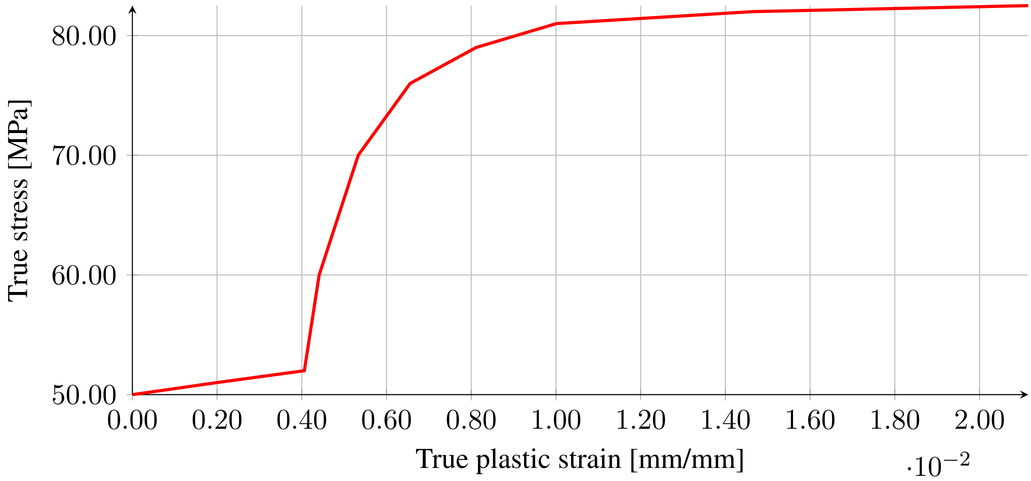


Figure C.8: Test case C, multilinear material model

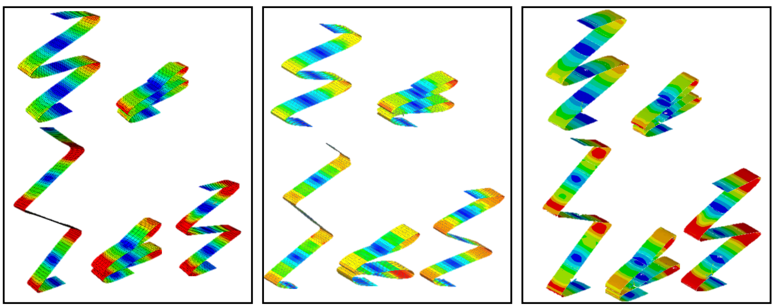


Figure C.9: Test case C Von-Misses visualization (Abaqus / Nastran / Ansys)

Figure C.10 shows the von Mises stress and the plastic strain, on a path defined by the center nodes along the long edge of the mesh, at the last time step of the analysis. The results are following the same path, with the exceptions of some peaks. A detailed discussions of these differences is not included.

C.2 Contact Interaction

The first concept required by AP209 to support contact, is the definition of surface and edge regions. In this chapter we focus on surface contact region, yet the same concepts are applicable for edge regions.

AP209 supports the definitions of element groups, however, it does not relate the group to any surface or edge identifier. We extended the entity **element_group**, by adding the subtype **element_aspect_group**. The already existing data type ***element_aspect***, can represent the aspect of an element; edge, face, and volume. This new entity has an attribute *aspect* which can hold an ***element_aspect*** value. With this, a surface region can

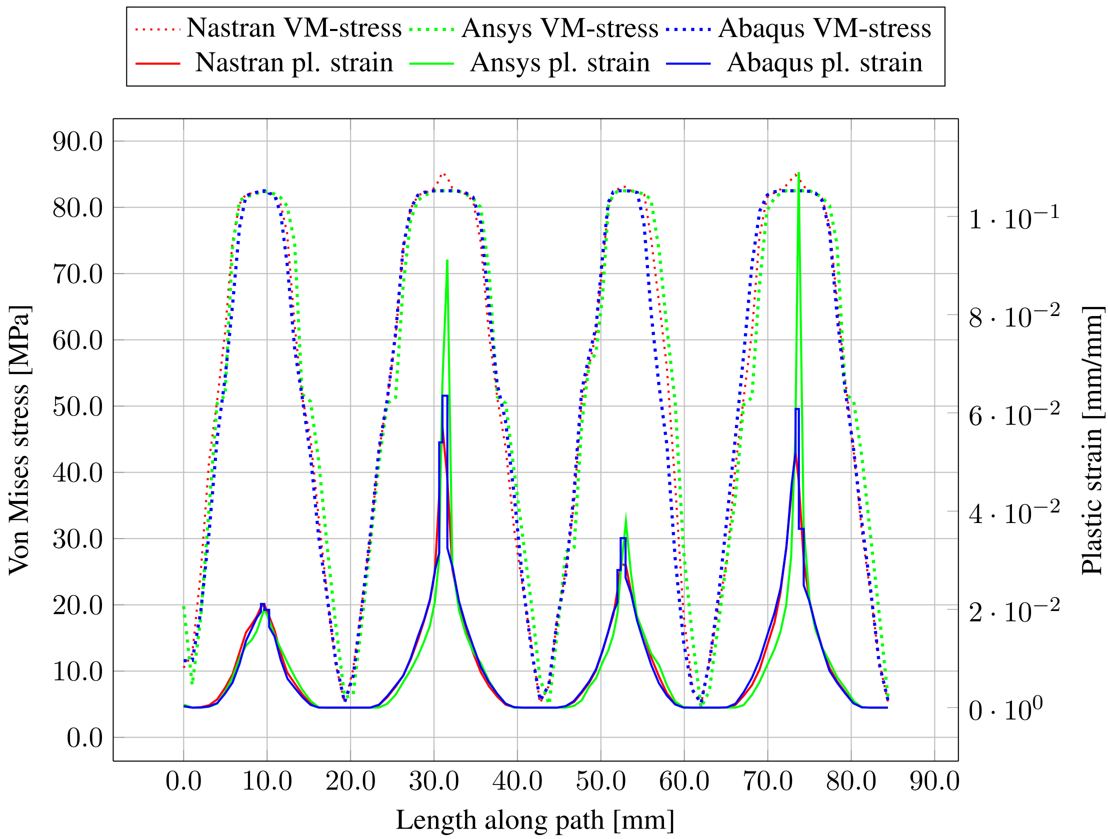


Figure C.10: Von Mises stress and plastic strain

be represented by referencing a set of elements, and the face identification of the elements belonging to the surface, for surface and volume elements.

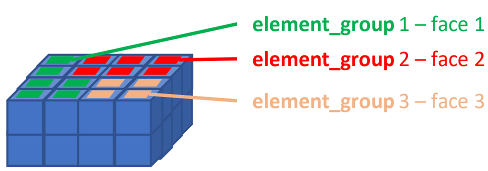


Figure C.11: Element groups and faces

The EXPRESS code in D.1, shows the definition of **element_aspect_group.** Note `WR1`, a *where rule*, which upon model validation, will check that there is a consistency between the element type and the selected aspect. For example, a face is not allowed on a 1D element. The code of the function used in this *where rule* is left out for simplicity.

```
ENTITY element_aspect_group
   ABSTRACT SUPERTYPE OF(ONEOF(
      volume_3d_element_aspect_group,
      volume_2d_element_aspect_group,
          surface_3d_element_aspect_group,
          surface_2d_element_aspect_group
   ))
   SUBTYPE OF (element_group);
   aspect : element_aspect;
WHERE
   WR1 : consistent_elements_aspect(aspect, elements);
END_ENTITY;
```

Listing C.1: EXPRESS code of **element_group**

element_group is already a supertype of element specific groups; volume 3D, volume 2D, surface 3D, etc. These entities hold *where rules* that restricts the element types allowed in the group. Additional entities were added, which inherits from both existing element type groups, and the abstract supertype **element_aspect_group**. An *abstract* entity, is an entity that is only allowed to be instantiated through one of its non-abstract subtypes. Figure C.12 shows all new implemented entities.

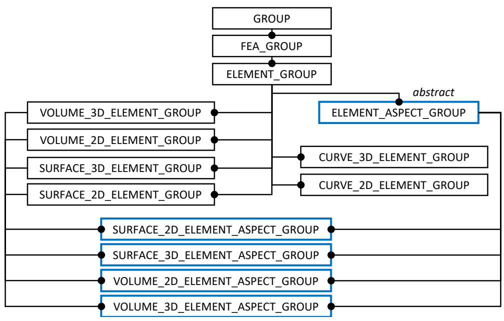


Figure C.12: Element group entities (In blue; new entities)

Now, by using the element specific aspect groups entities, validation rules, if checked, will ensure that the appropriate element types and element aspects are used.

To allow for a region with different face IDs across its belonging elements, multiple groups can be defined, and related with the already existing entity **element_group_relationship**.

Further, parameters for the contact interactions are required to be represented. In AP209, the entity **state_definition** is used for everything that is load and boundary condition related through a set of different subtypes. A natural place to add support for contact interactions is as new subtypes of **state_definition**. Multiple subtypes were added to not only support contact interaction, but also *glue* interaction, and potential other types of

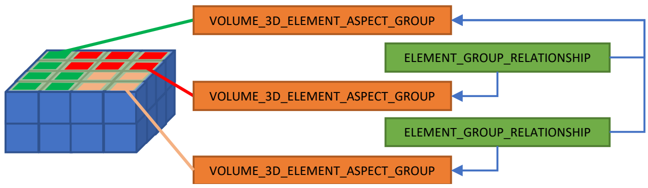


Figure C.13: Instance diagram; Relating multiple sub-regions with different face IDs.

interactions between mesh regions. An entity **fea_interaction** was added as a subtype of **state_definition**. Furthermore, a hierachy of new entities where added as subtypes of **fea_interaction**, as seen Figure C.14.

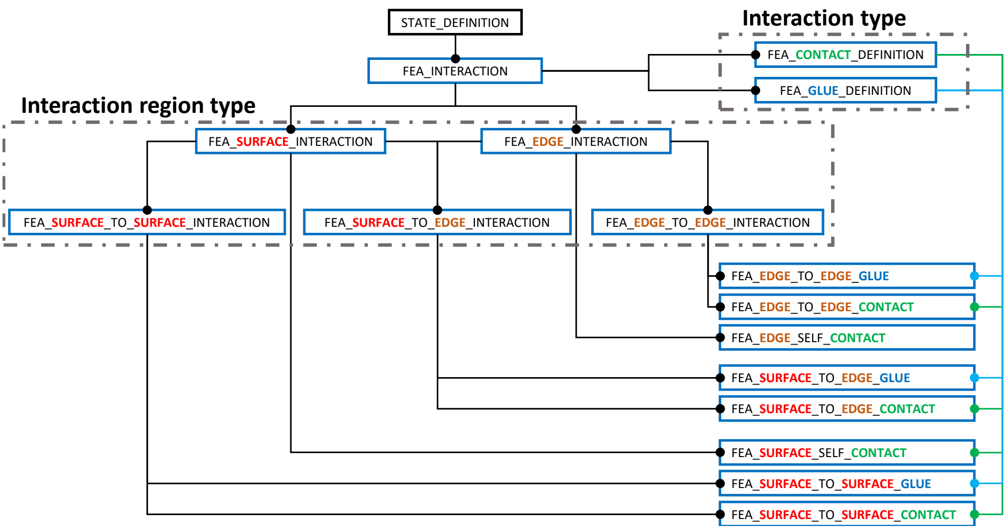


Figure C.14: Hierachy of new FEA interaction entities. Text colors are added for clarity

The **fea_contact_definition** and **fea_glue_definition**, are used to define an interaction as contact or gluing. Possibly other interaction types could eventually be implemented in the future, and follow the same structure. They hold no attributes, but are used to collect the specific parameters describing the interaction properties used in the solver.

For each different type of region interaction; single surface, surface to surface, single edge, edge to edge, and surface to edge, there exists an entity inheriting from **fea_interaction**, allowing for both self interaction, and interaction across regions. **fea_surface_interaction** and **fea_edge_interaction** has one attribute referencing one region (**fea_group**) as a *master* region. The *surface to surface*, *edge to edge*, and *surface to edge*, adds a second attribute, referencing a *slave region*. Each of these hold *where rules*

which restrict the type of **fea_group** that may be referenced, and which element aspect the groups have. For example, **surface_interaction** enforces a *where rule* which specifies that the referenced **fea_group**, has to either be a subtype of **element_aspect_group** with aspect set to a face, or a **node_group** (a surface can in certain solvers be defined by a set of nodes).
The express code for the new entities can be found in Appendix D.

C.3 Nonlinear Materials

Support for nonlinear materials in AP209, is not documented, however, materials in STEP are supported in a very generic way. There are specific entities and types for linear material properties such as E-modulus, poisson ratio, density, and a few others. These are specializations of generic material properties.
In this study, we didn't attempt to cover all nonlinear material concepts, but focused on two simple models; perfectly plastic, and multilinear material model.
Figure C.15 shows an example of instances specifying a FEA material. In this case, only the E-modulus and poisson ratio are specified for the linear analysis.

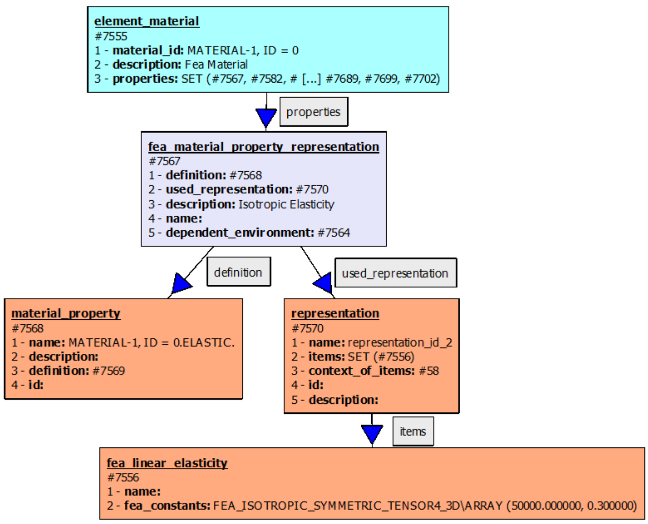


Figure C.15: Instance structure specifying a linear material with E-modulus and poisson ratio

A FEA material in STEP is identified by the entity **element_material**. It collects one or more material properties instance structures, which together defines the material properties. For each property, a structure of **fea_material_property_representation**, **material_property**, **representation**, and **representation_item** is needed. In the example, the **representation_item** is a **fea_linear_elasticity**. For nonlinear material properties, such specialized entities do not exist, and the use of more generic entities are used. For a simple perfectly plastic material model, we only need an additional property and value, specifying the yield strength. We recommend that this is done by a similar structure as the above example, but set the value of the yield point in a **measure_representation_item**, as seen in Figure C.16. The value is a stress value, and the *name* can be used to specify this is a *true stress* value, as opposed to *engineering stress*, or other types of stress. To identify the property as a yield point, the *description* attribute of the **fea_material_property_representation**, or the *name* attribute can be used and set to the string identifier `Yield point`.

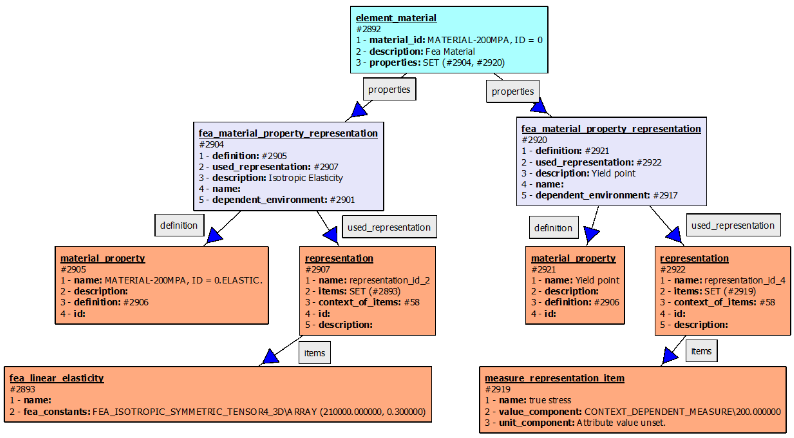


Figure C.16: Instance structure specifying a yield point on a material.

For a multilinear model, which in FEA is commonly specified in a tabular form, still needs to be specified as individual properties of the material. Multilinear models can be specified in different forms, but in this case we use data points of true stress - true plastic strain values. Each value is treated as properties to the associated material as seen in Figure C.17

The values are identified by the same attribute as for the plasticity model, with the string identifier `Plasticity`. This identifier is used, instead of specifying it as a `multilinear plasticity material model`, because in the context of the STEP data model, these values are only properties of the material, which *can be used* to specify a multilinear material model by a system processing the data.

To identify the pairs of values, the entity **property_definition_relationship** is used.

These constructs for specifying data, which in FEA are relatively simple, become rather complex in STEP. This is because STEP data models are intended for relating to multiple domains, and additional data can be associated to most entities. For example, each **fea_material_property_representation**, has a *dependent_environment* attribute, which can be used to specify the environment conditions under which the material values were determined in. The representation of a physical test performed, which was used to establish the material model, could also be related to the values. In addition, documentation,

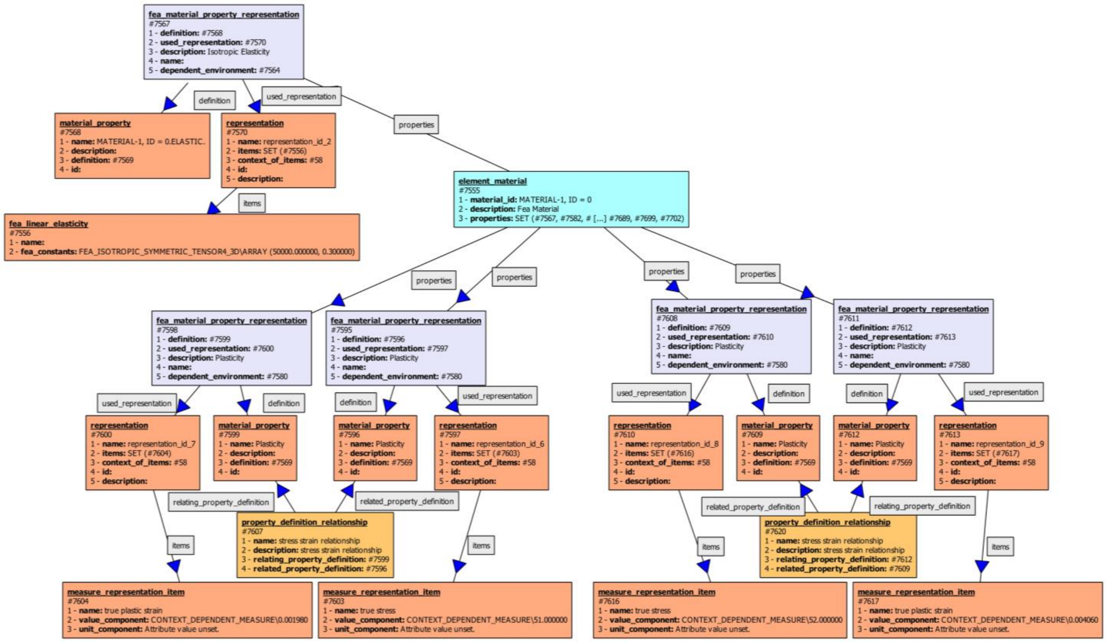


Figure C.17: Instance structure specifying a multilinear material model.

pictures, and other metadata, could be attached using STEP data.
When AP209 is used for a simple translation of a single FEM analysis, all of this additional information is not required. However, in the context PLM and SDM, and storing the data for long-term archiving, these features becomes crucial.

Appendix D

fea_interaction EXPRESS code

```
ENTITY fea_interaction
   SUBTYPE OF (state_definition);
   name           : label;
   description    : OPTIONAL text;
END_ENTITY;

ENTITY fea_surface_interaction
   SUBTYPE OF (fea_interaction);
   master_face_region : fea_group;
(*WHERE:
    Master region is ELEMENT_ASPECT_GROUP with
    face type aspect or NODE_GROUP*)
END_ENTITY;

ENTITY fea_edge_interaction
   SUBTYPE OF (fea_interaction);
   master_edge_region : fea_group;
(*WHERE:
Master region is ELEMENT_ASPECT_GROUP with
edge type aspect or CURVE_3D_ELEMENT_GROUP or
CURVE_2D_ELEMENT_GROUP or NODE_GROUP*)
END_ENTITY;

ENTITY fea_surface_to_edge_interaction
   SUBTYPE OF (fea_surface_interaction, fea_edge_interaction);
   edge_region_is_master : BOOLEAN;
END_ENTITY;

ENTITY fea_surface_to_surface_interaction
   SUBTYPE OF (fea_surface_interaction);
   slave_face_region : fea_group;
(*WHERE:
```

```
Slave region is ELEMENT_ASPECT_GROUP with
face type aspect or NODE_GROUP*)
END_ENTITY;

ENTITY fea_edge_to_edge_interaction
   SUBTYPE OF (fea_edge_interaction);
   slave_edge_region : fea_group;

(*WHERE:
Slave region is ELEMENT_ASPECT_GROUP with
edge type aspect or CURVE_3D_ELEMENT_GROUP or
CURVE_2D_ELEMENT_GROUP or NODE_GROUP*)
END_ENTITY;

ENTITY fea_contact_definition
        ABSTRACT SUPERTYPE
        SUBTYPE OF (fea_interaction);
--WHERE
-- WR1: restrict properties to contact properties;
END_ENTITY;

ENTITY fea_glue_definition
        ABSTRACT SUPERTYPE
        SUBTYPE OF (fea_interaction);

--WHERE
-- WR1: restrict properties to glue properties;
END_ENTITY;


ENTITY fea_surface_self_contact
   SUBTYPE OF (fea_contact_definition, fea_surface_interaction);
END_ENTITY;

ENTITY fea_surface_to_surface_contact
   SUBTYPE OF (fea_contact_definition, fea_surface_to_surface_interaction);
END_ENTITY;

ENTITY fea_surface_to_edge_contact
   SUBTYPE OF (fea_contact_definition, fea_surface_to_edge_interaction);
END_ENTITY;

ENTITY fea_edge_to_edge_contact
   SUBTYPE OF (fea_contact_definition, fea_edge_to_edge_interaction);
END_ENTITY;

ENTITY fea_surface_to_surface_gluing
   SUBTYPE OF (fea_glue_definition, fea_surface_to_surface_interaction);
END_ENTITY;
```

```

ENTITY fea_surface_to_edge_gluing
   SUBTYPE OF (fea_glue_definition, fea_surface_to_edge_interaction);
END_ENTITY;

ENTITY fea_edge_to_edge_gluing
   SUBTYPE OF (fea_glue_definition, fea_edge_to_edge_interaction);
END_ENTITY;
```

Listing D.1: EXPRESS code for **fea_interaction** and its subtypes

ISBN 978-82-471-9399-0 (printed ver.)
ISBN 978-82-471-9676-2 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

NTNU

Norwegian University of
Science and Technology