Mapping Between Analysis Models and Other Type Models in CIS/2

(Based on LPM500, Version from January 27, 2000)

C. Eastman

Purpose:

An analysis model will have different information than a conceptual design model or than a detailed manufacturing model. Each of these may be composed of different components that do not correspond one-for-one to components in other models. There needs to be some way to track the correspondence between the instances in different types of models. Here we consider the correspondence between analysis models and the other types.

Structure:

A model used for analysis is structured at the top level as an analysis_model, while all other models are defined as a subclass of assembly. A design model is called an assembly_design while a manufacturing model is another subclass of assembly, structured at the top as a assembly_manufacturing. If a corresponding analysis model exists for a design or manufacturing model, this is identified at the top level with an analysis_model_mapping, as shown in orange in Figure One. It identifies if the elements of an analysis_model correspond to parts in other models. The reference made is to a subclass of assembly.

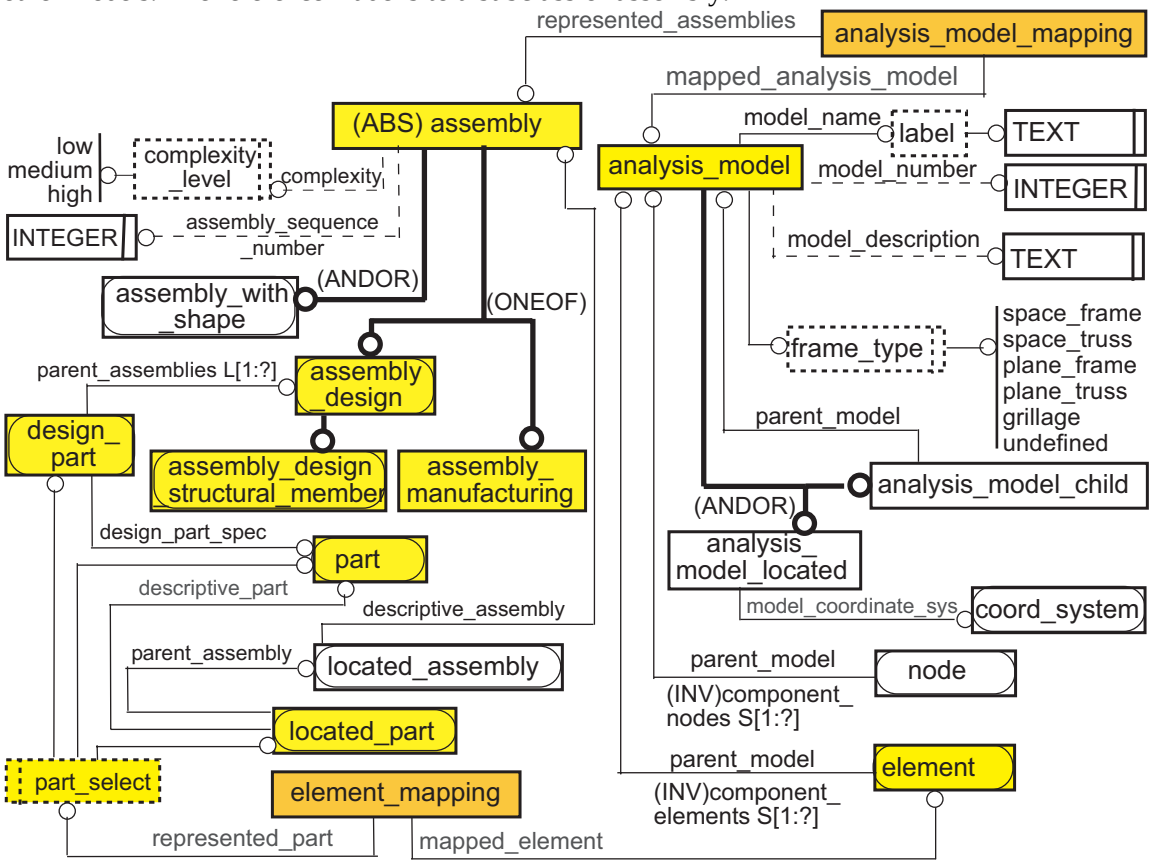

Figure One: The EXPRESS-G structure for mapping between different life cycle stage steel models.

At the level of individual structural members, the correspondence between analysis elements and design or manufacturing parts is captured by element_mapping. It associates an analysis element and either a part, a design_part or a located_part. The choice is allowed by use of the select type part_select. A part specifies the geometry of a repeated part, a design_part specifies a part within one or more assemblies or a located_part specifies a component of an assembly_manufacturing. An analysis element may be associated with only one such part of one type. But multiple elements may be associated with the same part, such as when a beam supports multiple girders.

For the high level mapping the EXPRESS code is:

```
ENTITY analysis_model_mapping;
      mapped_analysis_model : analysis_model;
      represented_assemblies : SET [1:?] OF assembly;
END_ENTITY;
```

At the level of individual structural members, the EXPRESS code is:

```
TYPE part_select
 = SELECT
      (part, design_part, located_part);
END_TYPE;

ENTITY element_mapping;
      mapped_element : element;
      represented_part : part_select;
END_ENTITY;
```

The correspondences between instances of the various types of models must be defined within the export process of the application that created the related entities. CIS/2 has the capability to capture and pass on such relations, but not to identify them if they are not provided by the generating application.

Another structure can optionally be used for mappings between the three different types of models. Its intention is different than the example above, but may be appropriate in some cases. There is the notion, especially found in a number of European modeling efforts, of carrying a requirements specification for some designed element and also the specification of the performance of the realized design. These are called “as required” and “as realized” respectively. A mapping structure exists that provides for associating the “as required” analysis loads with the “as realized” design resistances for a structural member.

This structure is shown in Figure Two. It begins with the abstract supertype design_result, which can be used to define connections, joint systems, members or parts. Design_result specifies the capacity of a structural component. These are defined in terms a result_name and a resistance. Using the ANDOR subclass option, these subtypes may optionally include the attributes of design_result_mapped and/or design_result_resolved. Design_result_mapped adds the analysis_result_set attribute class, which resolves into a load_case. This allows loads to be associated with a structural member, along with its design resistances. The optional ANDOR supertype design_result_resolved adds the reaction force resulting from the loading conditions and the design_factor.

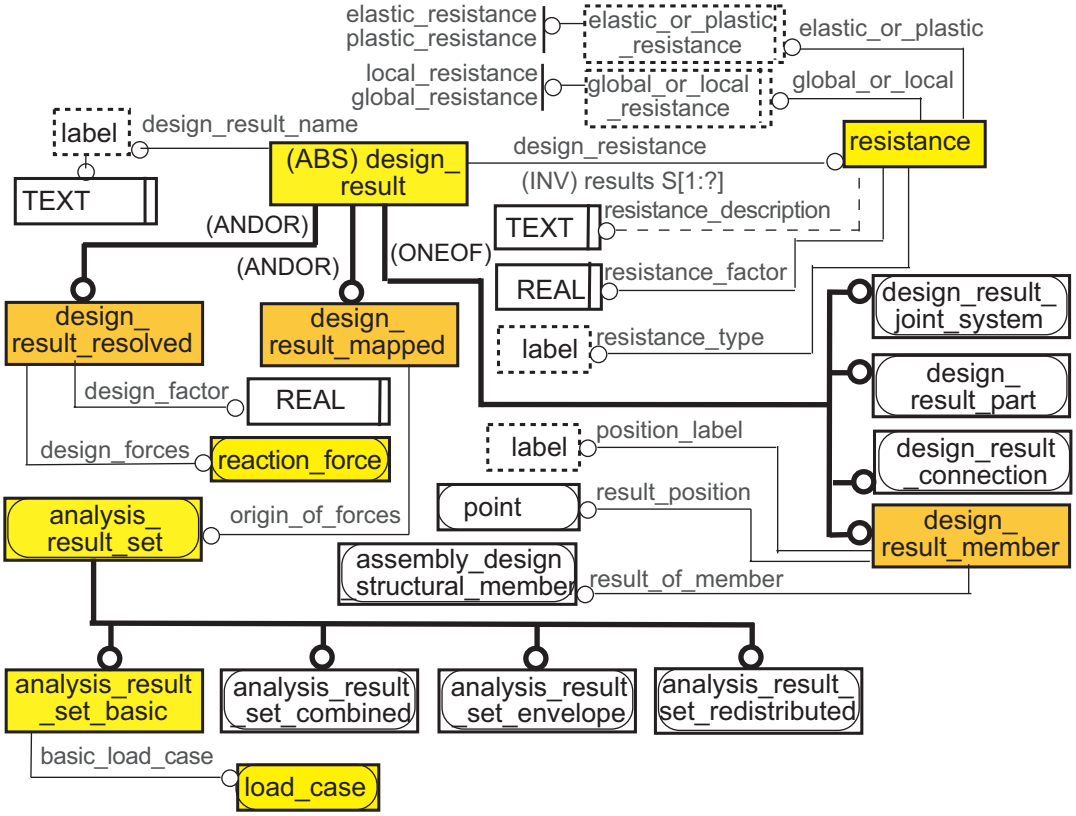

Figure Two: Another structure for mapping between design members, design resistances, loads and stresses.

```
ENTITY design_result
ABSTRACT SUPERTYPE OF (ONEOF
            (design_result_connection,
            design_result_joint_system,
            design_result_member,
            design_result_part) ANDOR
            design_result_mapped ANDOR
            design_result_resolved);
      design_result_name : label;
      design_resistance : resistance;
END_ENTITY;

ENTITY resistance
ABSTRACT SUPERTYPE OF (ONEOF(resistance_bending, resistance_shear,
resistance_axial));
      resistance_type : label;
      resistance_description : OPTIONAL text;
      resistance_factor : REAL;
      elastic_or_plastic : elastic_or_plastic_resistance;
      local_or_global : global_or_local_resistance;
INVERSE
      results : SET [1:?] OF design_result FOR design_resistance;
END_ENTITY;
```

```
ENTITY design_result_mapped
SUBTYPE OF (design_result);
      origin_of_forces : analysis_results_set;
END_ENTITY;

ENTITY design_result_member
SUBTYPE OF (design_result);
      result_for_member : assembly_design_structural_member;
      result_position : point;
      position_label : label;
END_ENTITY;

ENTITY design_result_resolved
SUBTYPE OF (design_result);
      design_forces : reaction_force;
      design_factor : REAL;
END_ENTITY;

ENTITY design_result_member
SUBTYPE OF (design_result);
      result_for_member : assembly_design_structural_member;
      result_position : point;
      position_label : label;
END_ENTITY;
```

The expanded structure for design_result_member, if it inherits all the possible supertypes is:

```
ENTITY design_result_member
 (design_result:
      design_result_name : label;
      design_resistance : resistance;
      );
(design_result_mapped:
      origin_of_forces : analysis_results_set;
      );
(design_result_resolved:
SUBTYPE OF (design_result);
      design_forces : reaction_force;
      design_factor : REAL;
      );
      result_for_member : assembly_design_structural_member;
      result_position : point;
      position_label : label;
END_ENTITY;
```

Notice that the analysis_results_set attribute is optional, because of the ANDOR clause, and the reaction_forces and design_factor together are optional, also because of the ANDOR clause. The other attributes are mandatory.