STRUCTURAL ANALYSIS EXTENSION OF THE IFC MODELLING FRAMEWORK

SUBMITTED: November 2002
REVISED: April 2003
PUBLISHED: July 2003 at http://www.itcon.org/2003/14
EDITOR: Väino Tarandi

Matthias Weise, Research Assistent,
Institute of Applied Computer Science in Civil Engineering, University of Technology Dresden, Germany
email: Matthias.Weise@cib.bau.tu-dresden.de

Peter Katranuschkov, Senior Research Assistent, Dr.-Ing.,
Institute of Applied Computer Science in Civil Engineering, University of Technology Dresden, Germany
email: Peter.Katranuschkov@cib.bau.tu-dresden.de

Thomas Liebich, Dr.-Ing.,
Thomas Liebich Consulting, München, Germany / Partner AEC3 Ltd., Thatcham, UK;
email: liebich@uumail.de

Raimar J. Scherer, Professor, Dr.-Ing.,
Institute of Applied Computer Science in Civil Engineering, University of Technology Dresden, Germany
email: Raimar.J.Scherer@cib.bau.tu-dresden.de

SUMMARY: *The further development of the IFC2x platform gains in importance with its rapidly expanding acceptance in real design and construction practice. This is underpinned by the 23 IAI projects inaugurated in the last few years, aiming to produce model extensions that can adequately support different AEC/FM processes and sub-domains. This paper provides an overview of the structural analysis extension to the IFCs, being developed as part of the IAI ST-4 project. Presented are the scope of the model, major concepts that have guided the conceptual modelling work and the envisaged actors and usage scenarios leading to data exchange views. An illustrative example for the possible use of the proposed model is also provided. The introduced top-level entities are briefly discussed, together with their intended usage at the detailed levels of the model. At the end, the current development status, planned further activities and future plans, and expected results for the practice are briefly outlined.*

KEYWORDS: *IFC, Product Data Modelling, Structural Analysis.*

1. INTRODUCTION

After nearly seven years of development work and a number of prior Research and Technology Development efforts that have provided the necessary background, the Industry Foundation Classes (IFC) of the IAI are now rapidly advancing towards a practically applicable industry standard for product data exchange and sharing in the AEC/FM industry. The current modelling framework IFC2x (Wix & Liebich 2001) defines the basic structure to support general-purpose and architectural CAD functionality, as well as some high-level management processes. However, to provide comprehensive features at more detailed discipline levels in the separate AEC/FM domains (architecture, structural engineering, building services engineering, cost survey, construction management etc.) and to enable better coverage of a broad range of construction project phases, specific domain extensions to this basic structure are needed as well. Therefore, the *platform* concept has been introduced (Wix 2001). It defines a clear baseline for the conceptual development of further additive domain models, as well as a methodology and a road map for the deployment of such developments in practice.

In accordance with that, several industry related efforts targeting specific modelling extensions for the next IFC Release have been started recently by the IAI (cf. IAI R3 2002 - http://cic.vtt.fi/niai/technical/ifc_3). Among these are four projects in the structural engineering domain. One of these is the ST-4 project “Structural Analysis Model and Steel Constructions” addressed in this paper.

ST-4 was inaugurated in November 2000 by the German Speaking Chapter of the IAI. Recently, it was accepted as part of the next IFC release. It develops two domain extension models within the overall IFC framework (see FIG. 1):

- *structural analysis*, and
- *steel constructions*.

The second is a large model, based on the available comprehensive German industry standard for data exchange in the steel construction domain *"Produktschnittstelle Stahlbau"* (DSTV 2000) but fully re-worked to fit into the IFC framework.

In contrast, the structural analysis extension, initially proposed in (Weise et al. 2000) is a lean extension model which does not only target stand-alone usage for structural analysis purposes, but aims at supporting the steel constructions model, and in fact all other structural extension models as well. Its scope, basic modelling concepts, envisaged actors, usage scenarios and a "how-to" example, together with the current development status and the planned activities to the end and after the project, provide the focus of the presentation in the paper.

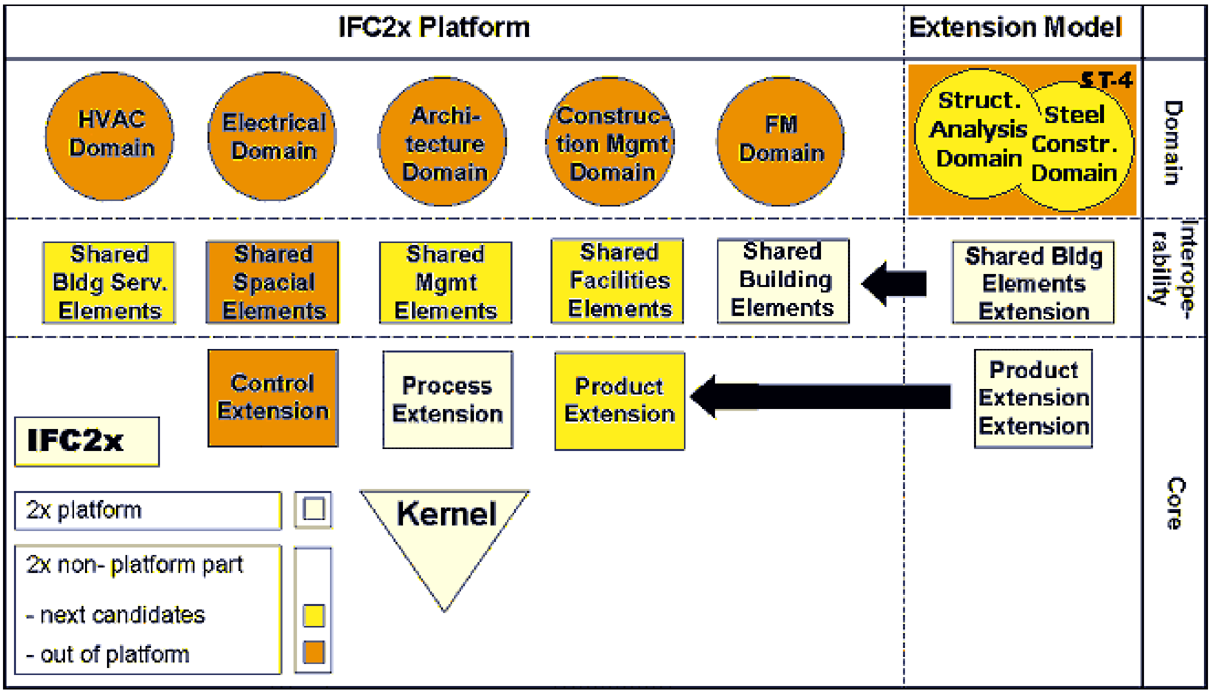

FIG. 1: The ST-4 extension models within the IFC 2x platform architecture / adapted from (Wix 2001) /

2. SCOPE

A major design objective of the work on the structural analysis model has been to *tightly integrate* the structural engineering domain into the information management process provided by the IFCs.

It is therefore intended that all already available building data are re-used, extended as needed, and associated with structural assumptions. The focus is on ensuring that the IFC model can capture important decisions of the structural engineer and make these visible to other related domains. The proposed model extension concentrates on the provision of features that will enable faster achievement of these objectives whereas "pure" structural engineering problems are left for future extensions.

Accordingly, the following features are considered in the current model scope:

- *Defining the bearing structure of a building* - all designers should understand the building structure and recognise the building elements needed to carry loads;

- *Defining planar and/or spatial structural analysis models* that can be used by structural analysis applications - this may, however, need further settings as the structural data are not stored in full detail;
- *Specification of different structural analysis submodels* needed to describe and analyse different aspects or parts of the building; dependencies between these models can be saved for further use;
- *Defining relationships* between the existing building elements and the elements of the structural analysis models;
- *Specification of loadings* including point, line and planar loads and the assignment to load groups, load cases and load combinations;
- *Analysis results* (only basic definitions of forces and displacements).

Currently out of scope are data related to dynamic analysis, the definition of prestressing, some stability problems, and the detailed description of finite elements used in numeric analysis computations.

However, in order to support the practical implementation of the proposed model, additional restrictions need to be defined, to reduce the requirements with respect to the structural extension model. This strategy is in fact suggested by the IAI, with the definition of "data exchange views" and, though in a different way, is also followed in the SABLE project within BLIS (cf. SABLE 2002). In contrast to the original IAI vision, it is a pragmatic approach aiming at reducing the data definitions of the IFC model that have to be understood and managed by CAD, HVAC and other highly specialised applications. Indeed, to combine the vision of a steadily growing data repository collecting all relevant project data with the concept of using "data exchange views", additional model management tools are needed. However, it can be expected that such tools will become available when the IFCs are used on broader scale.

Hence, whilst on the basis of the IFC modelling framework many theoretical possibilities can be envisaged, it is reasonable in a first stage to avoid "overstretching" the scope, so that the rapid development of structural applications that can yield practical proof of concepts is facilitated. For this purpose, it is important to define appropriate use cases and application scenarios, identifying the specific requirements for each envisaged process, i.e. which actor is sending what information according to which data exchange view and to which (type of) application. This important aspect of the model development process is documented in "Volume 0 – Application Scenarios" of the ST-4 documentation (cf. IAI ST-4 2002).

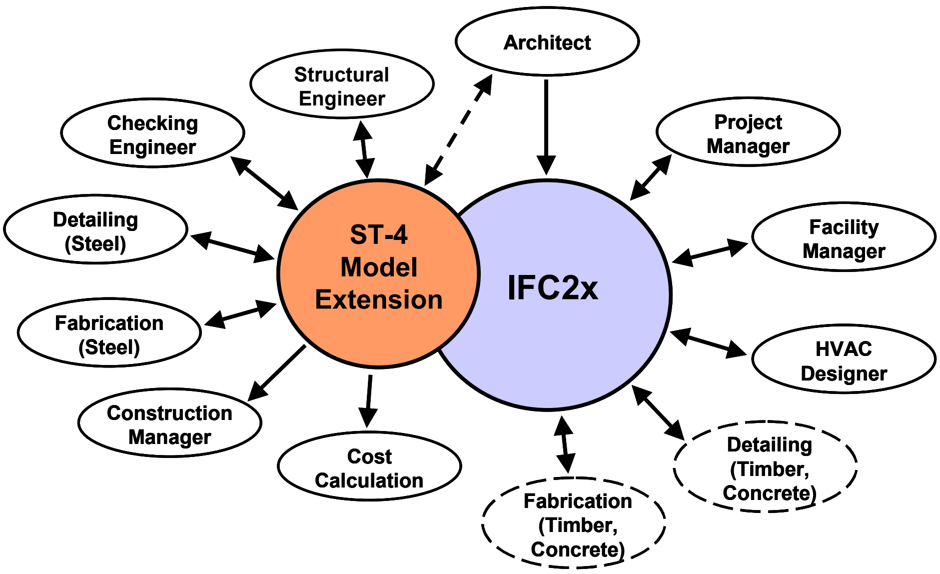

FIG. 2: Overall application scenario and the involved actors.

FIG. 2 schematically shows the overall application scenario supported by IFC2x and the ST-4 extensions, and the respectively involved actors. For the structural analysis extension mainly the roles of the *structural engineer*, the *checking engineer* (specifically for Germany) and the *detail designer* are of interest. However, other roles which are not directly within scope are also often involved in the process by sharing (or referencing) relevant structural data. Here especially relevant tasks of *architects* and *construction managers* need to be taken into account.

In the next two chapters we first concentrate on the features and concepts of the developed model to return then in chapter 5 to a more detailed discussion of actors and usage scenarios and proceed, in chapter 6, with an illustrative example showing how the model can be implemented for practical data exchange tasks.

3. MODELLING INTEGRATION APPROACH

While the definition of usage scenarios and the specification of data exchange views is mainly an implementation issue, the exploitation of the 'ladder principle' defined in the IFC framework provides the principal modelling approach for the creation of domain specific views. Thus, even though the 'ladder principle' (or layer concept) has currently no direct practical use, the proposed extension follows this idea. However, using it requires from a domain application to be able to understand, additionally, at least the *Interoperability Layer.* An application cannot only support its own domain data, it has to understand also those data objects which are needed for the efficient information exchange *between* different domains. However, the definition of "data exchange views" can significantly simplify the needed application interfaces.

Concerning the implementation of the model, actors and use cases are of primary interest. Normally, actors from other domains will not need the full range of available structural information. Rather, they would want to be informed about basic decisions which may affect their own work. Such commonly used information is separated from special domain information by the layered structure of the IFCs. The *Interoperability Layer* and the underlying layers are responsible for the domain-independent data exchange. On the other hand, each *Domain Layer* used by the respective domain experts and applications is built on top of the *Interoperability Layer* and interlinked with it. Therefore, each independent domain can contain the full set of domain specific data.

The work of the structural engineer typically starts with existing architectural information, as shown on the left side of FIG. 3. Two main steps are then performed in the process of defining the structural analysis model. First, the existing architectural information must be appropriately re-structured to specify all load-bearing elements. Secondly, relevant building elements must be idealized and assembled to adequate structural analysis models. FIG. 3 schematically illustrates the process.

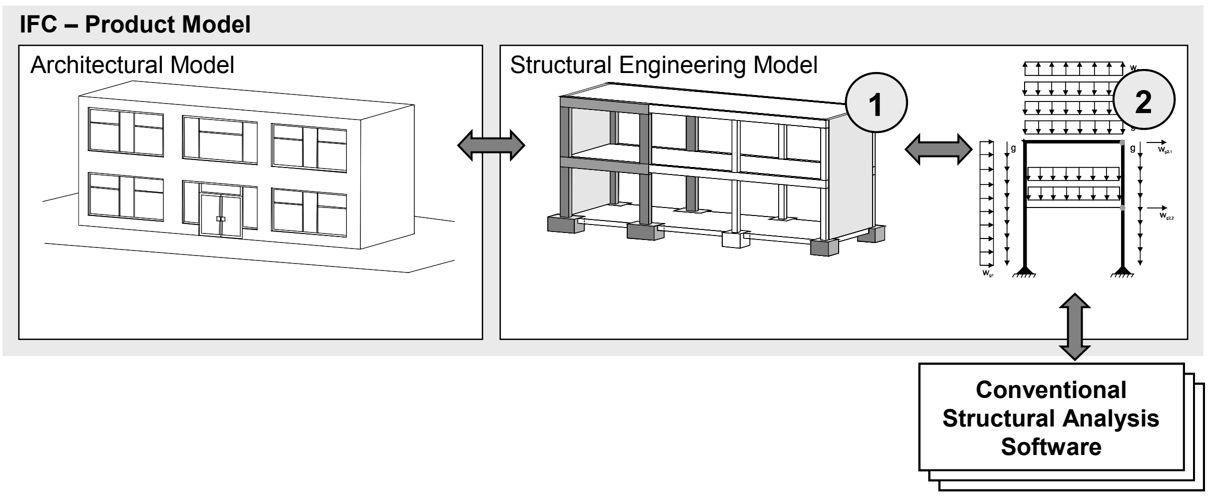

FIG. 3: The supported interoperability process with the developed extension.

However, defining the connections to the architectural building data at the very outset is not the way structural engineers work today. Furthermore, since this data may not always be available, typical Finite Element applications rarely support functions that can enable the interconnection of structural information with high-level building data. Therefore, compared to the initial proposal of the model (Weise et al. 2000), a direct relationship between the IFC data and the structural analysis model is provided (as shown by the arrow on the right side of FIG. 3), whereas the relation between the two process steps and the thereby created information is only optional.

The description of structural analysis models is needed in all cases and therefore the focus of the developed model extension is primarily on step two. Also, since many existing IFC classes can be used to represent step one, the main information gaps to be covered are actually related to the description of structural analysis models.

The software tools shown in the scenarios provide only possible examples. Within the German Working Group of the ST-4 close interest and participation in the model development work is expressed by the German software vendors SOFiSTiK, RIB, BOCAD, SCIA, DLUBAL, Friedrich & Lochner and Nemetschek, and by the Finnish company TEKLA.

4. MODELLING CONCEPTS

As already indicated, the structural analysis extension should provide the representation of the load-bearing system, the load-bearing elements and the external influences (loads) on the structure of a building, by appropriate usage of the available generally defined building elements in the IFC core model. This serves two objectives:

1) Capturing of basic structural engineering decisions about the bearing structure which should enable the tracing and understanding of the rationale by which structural analysis models are derived from the architectural layout, and

2) Enabling the access of other AEC/FM professionals to structural design data – to allow them to consider the structural properties and behaviour of building elements, especially when changes of the latter need to be conducted.

FIG. 4 below provides an example for the definition of structural models, performed on the basis of decisions about the mechanical behaviour and modelling of the building, which can be captured in the overall building model through the constructs of the proposed domain extension. This should be helpful both for structural engineers and other AEC/FM professionals to obtain in a simple and formally well-defined manner basic structural information.

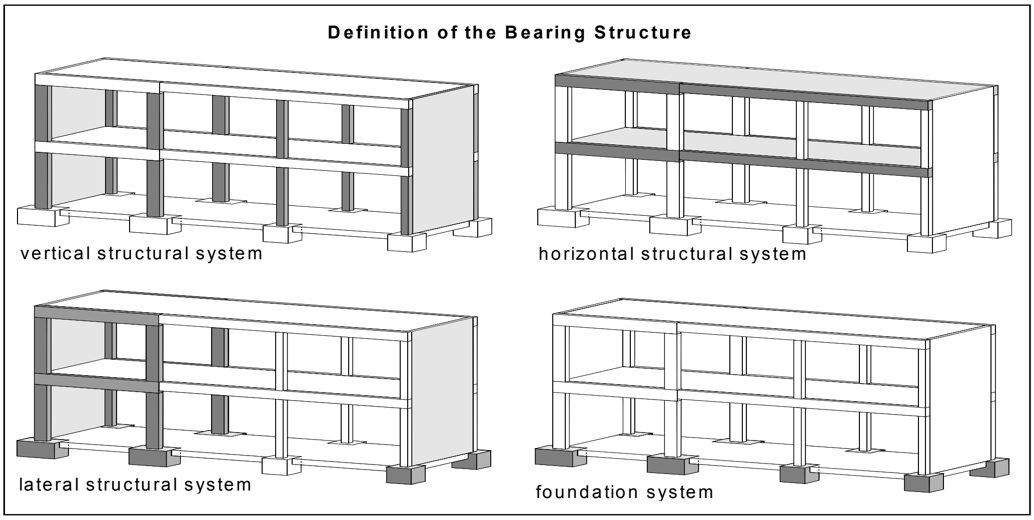

FIG. 4: Defining the bearing structure configuration using the basic IFC building elements.

A major conceptual design goal is the achievement of a logical connection between specific mechanical idealisations and the overall building model data. Thus, through the integration of the structural data in the building product model, their availability to all designers and throughout the building's life cycle is realised.

4.1 Strategy for integration

As understood for this extension, the integration of the relevant design domains is provided by the *Interoperability Layer* which is referenced by several domains encapsulated in the *Domain Layer*.

FIG. 5 illustrates the basic principle. Building elements are used as core concepts of the building to be designed. These building elements are referenced by elements of the respective aspect models which are used to model different functions of the building satisfying different owner or end user requirements. Therefore, a building element representing a physical object of a virtual building has a connection to its representations in which functional behavior or other aspects are described.

The cardinalities of the relationships shown in FIG. 5 represent the principal connection of an individual building element (or a group of building elements) to n representations used to model individual aspects. The basic idea is that one building element is not represented by any other building element at the same time (the composition of building elements into higher order assemblies is a feature that does not contradict to that statement). In contrast, one building element can have different representations in the same domain and at the same time.

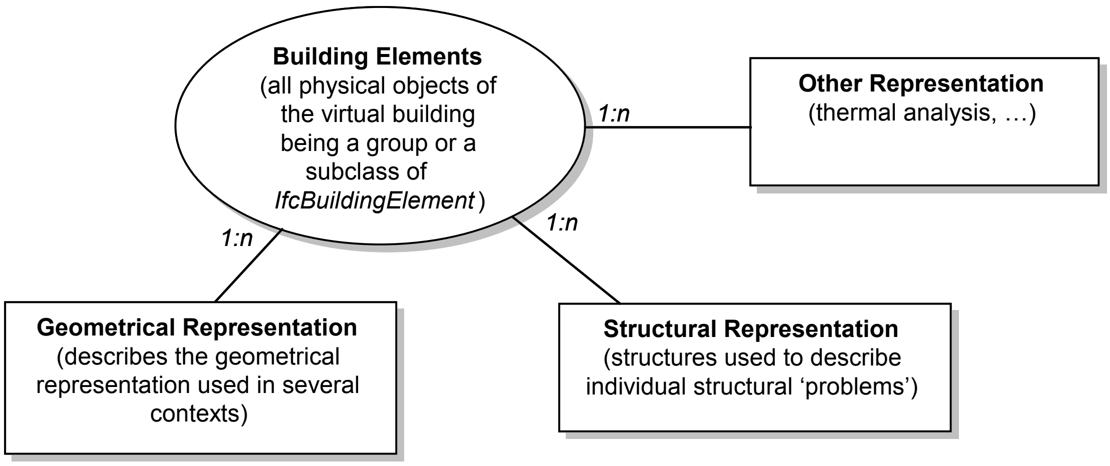

FIG. 5: Principle for integration of different domains using IFC building elements as core concept.

The IFC project model defines several classes representing building elements typically used for 'architectural design' (*IfcBuildingElement* and its subclasses). These building elements and their classification are not directly suitable for structural analysis tasks and therefore need to be extended by new classes dealing with the specific requirements of that domain. Thus, the idea of the proposed extension is to provide adequate 'mechanical elements' which represent the structural function of one or more building elements. This is achieved in first place through the introduced new concept *IfcStructuralItem*.

By the principal separation of the building elements from their mechanical representation(s), it is also possible to *overcome the semantic differences* between the available building element types in IFC and the element classification and usage appropriate for the structural engineering domain. A great number of such semantic differences exists. FIG. 6 below provides one typical example. Other examples include the different subdivision of the flat of a storey (floors of rooms and other spaces in the architectural domain, opposed to slabs spanning the distances between the supporting bearing elements in the structural domain), the different treatment of girders (individual beams in architecture, opposed to a continuous beam – eventually even with non coinciding segments in the structural domain) etc.

With the proposed approach, there exist practically no restrictions to the mechanical modelling process. Building elements can be joined together, grouped or used alone, in one or more co-existing and/or alternative structural analysis models.

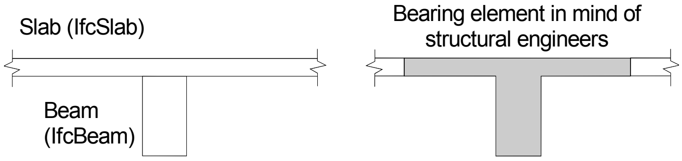

FIG. 6: Example of the different semantics used by architects and structural engineers.

4.2 Model overview

The EXPRESS-G diagrams shown on FIG. 7 and FIG. 9 provide an overview of the *inheritance structure* and the *major relationships and attributes* defined in the developed structural analysis extension model. On these figures, the main entities of the model and its integration with the IFC2x platform schemas can be recognised.

The two top-level entity classes *IfcStructuralItem* and *IfcStructuralActivity* both inherit their properties from the kernel entity class *IfcProduct*. Thereby general representational features such as *placement*, s*hape* and other fundamental product data attributes are consistently provided to *all* structural analysis entities in more or less the same manner as for all other tangible IFC objects. The use of the attribute 'Representation' is changed so that topological objects always reference the global coordinate system instead of the local coordinate system defined by the attribute 'ObjectPlacement'. This is necessary because topological objects (like e.g. a vertex point) have to be shared in order to define a proper overall topology of the mechanical structure. Other inherited attributes basically retain their intended meaning as defined for the IFC core model.

IfcStructuralItem is the root entity class for all classes representing *structural objects*, whereas *IfcStructuralActivity* is the root class for the objects representing *external impacts* (loads and other actions) on these objects.

IfcStructuralItem is further specialised along two main axes – structural members and structural connections which are linked through the relationship class *IfcRelConnectsStructuralMember*, subtype of the kernel class *IfcRelConnects*. In this way, the connectivity of the bearing structure is explicitly established. Additionally, the relationship between building elements and structural members can be captured by using the relationship class *IfcRelAssignsToStructuralMembers* (not shown) which inherits all necessary attributes from the class *IfcRelAssignsToProduct* but defines also some additional constraints for its usage. FIG. 8 shows the principal usage of structural members, structural connections, interrelations between them and their connection to building elements.

Structural members are further subtyped into linear and planar elements, and structural connections into point, line / curve and face connections respectively. As mentioned above, they *always* use a topological representation referring to the global coordinate system to provide for a more efficient data structure for structural analysis applications. Furthermore, connections can optionally be associated with boundary conditions that may specify node and edge restraints in terms of linear and rotational stiffness, as well as warping. Finite elements (which are a pure numerical abstraction) are not part of the model.

IfcStructuralActivity is specialised in the two branches *IfcStructuralAction* and *IfcStructuralReaction*, allowing to differentiate between external and internal actions. For the definition of their location they normally share the topological representation of "loaded" structural items. The relationship between structural elements (members, connections) and "activities" upon them is established through the *IfcRelConnectsStructuralActivity* class which is also subtyped from the *IfcRelConnects* class. Beside the topological definition, this relationship is also needed to avoid ambiguities and to enable the specification of load eccentricities. Additionally, each activity (load or other impact) is grouped either by *IfcStructuralLoadGroup* (for actions) or *IfcStructuralResultGroup* (for reactions) and associated to one or more analysis models (see FIG. 9). *IfcStructuralLoadGroup* can be further differentiated to represent load groups, load cases and load combinations.

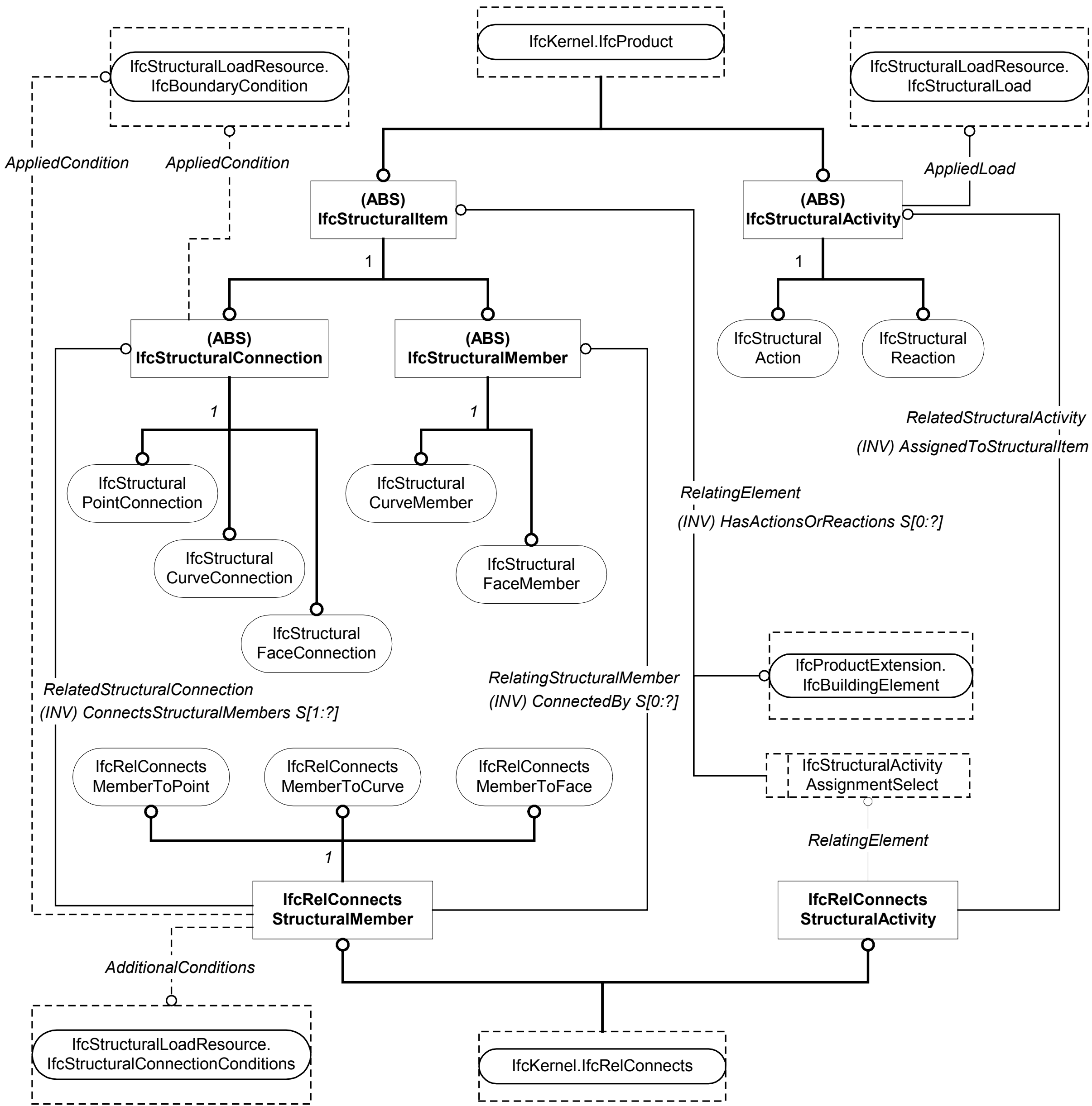

FIG. 7: High-level view of the main structural objects of the structural extension model in EXPRESS-G

External actions are subtyped into single loads, linear loads and planar loads, to represent the load dimension and define appropriate load quantities. The capability to re-use load definitions within an instantiated product model, as well as in other structural domain extension models, such as the models developed in the IAI ST-2 and ST-3 projects, is achieved by defining the load quantities themselves in a specific “structural” resource schema. In a similar way, material and section properties are made available by importing and appropriately extending the respective resource schemas of the IFC platform.

Within this extension approach the definition of reactions is only of minor priority. Hence, unlike *IfcStructuralAction,* reactions are not further subtyped. They can be represented in a simple to use *uniform* manner, in correspondence to the weaker requirements to the current model scope in that respect.

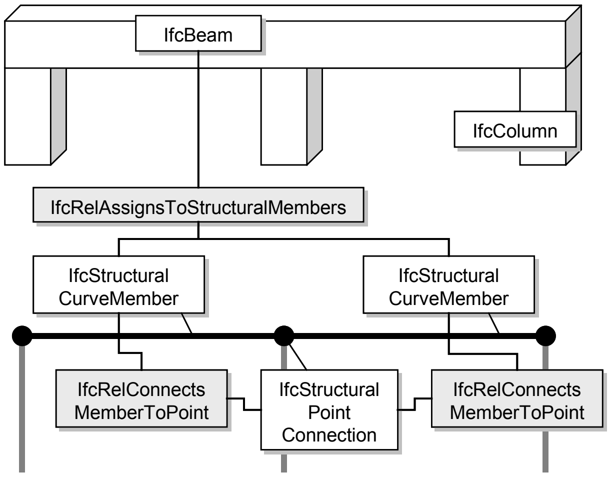

FIG. 8: Principal usage of structural items and their connection to building elements.

All abovementioned objects are single components which have to be assembled to structural analysis models. For this purpose the class *IfcStructuralAnalysisModel* is introduced to collect all participating structural members, structural connections (using the inherited grouping mechanism), applied loadings (by a reference to *IfcStructuralLoadGroup*) and calculated results (by a reference to *IfcStructuralResultGroup*). It extends the attributes provided by *IfcSystem* by enabling the specification of high-level information related to structural analysis, such as the model type (2D, 3D), the orientation of each 2D substructure etc.

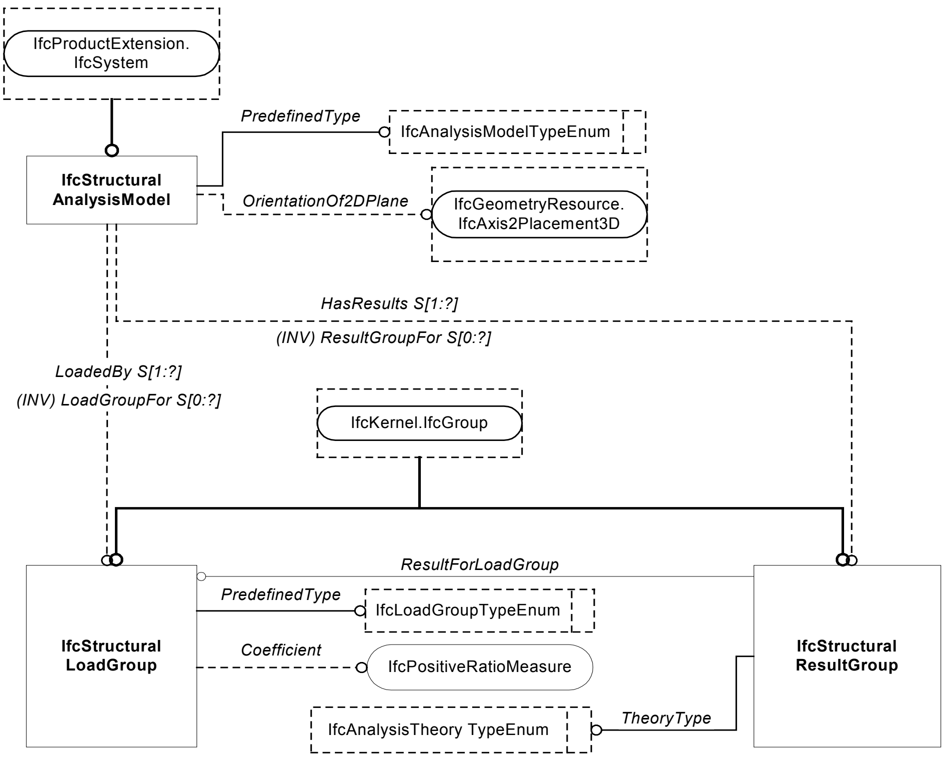

FIG. 9: High-level view of element grouping in EXPRESS-G

A comprehensive description of the developed model and the complete model schemas can be found in the on-line documentation (IAI ST-4 2002).

4.3 Extensions within existing IFC schemas

In the development of the extension model as outlined above care has been to define domain schemas in a downstream fashion, using and extending existing IFC platform concepts but suggesting as little as possible changes to the core schemas. However, for two reasons it was nevertheless necessary to make a few slight modifications within existing IFC schemas.

First, as the IFC modelling framework currently does not directly consider structural domain aspects, there are some concepts that are missing in the platform but are inevitable for the structural extension and must also be considered in interoperability scenarios.

Secondly, resource entities such as measures and material, that do already exist had to be extended by some additional properties to avoid unnecessary introduction of new (or even partially overlapping) concepts by subclassing.

Hence, in addition to the two new domain model schemas “Steel Construction” and “Structural Analysis”, the following changes have been suggested:

- Resource Layer
  - Measures -- added measures needed for structural analysis, such as mass per unit length, moment of inertia, temperature gradient, warping, etc.;
  - Material -- overall material properties + steel material properties, framework for adding properties for other structural materials, such as reinforced concrete, aluminium, wood;
  - Profile -- added new profile types, defining profile properties in the same way like material properties;
  - Geometry and Topology -- minor modification to fulfill some dedicated needs, as well as the inclusion of *loops* as defined in (ISO 10303-42 1999);
  - Load quantities -- *new resource* for the specification of load quantities, boundary and support conditions;
- Core and Interoperability Layer
  - minor additions -- assignment of profile properties, two other small additions for the steel construction domain.

Coordinated with the other structural projects in IAI, these suggestions have been forwarded to the IAI MSG (model support group) and are accepted for inclusion in the next official IFC release.

5. ACTORS, USAGE SCENARIOS AND VIEW DEFINITIONS

The IAI proposes a steadily growing data model collecting all relevant data of the building lifecycle (FIG. 10). However, with the widely used file-based data exchange which typically implies exchange of *all* model data, the receiving application may not be able (or not prepared) to handle the full range of that, partially non relevant information. Taking that into consideration, the importance of a clear statement regarding actors, usage scenarios and views from the model becomes obvious. Accordingly, this “view approach” has been adopted at the very outset of the ST-4 project.

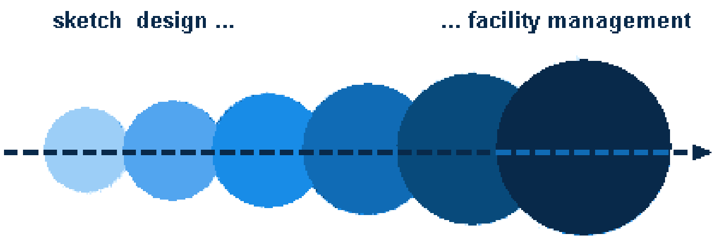

FIG. 10: IAI-vision of a steadily growing data model over building lifecycle

Actors and usage scenarios are collected and described in Volume 0 of the project documentation (IAI ST-4 2002), where views are also principally defined, to be further elaborated with the project's progress. Such "implementation views" are seen as a straight-forward implementation agreement to use the benefits of the IFCs while concentrating as much as possible on the own business logic and specific domain expertise.

FIG. 11 below shows the typical envisaged usage scenarios within the structural domain. The shown boxes are subdivided into three parts: actor, principal models and/or modelling objects involved, and the main targeted software application type(s). The middle part is omitted from the lower row of boxes as it is the same as for the respectively related upper boxes.

The outlined scenarios cover (1) the data exchanged between the structural engineer and the checking engineer - whose task is to validate the proposed structural design solution and verify the performed analysis, (2) the data exchanged between different analysis application performed by one or more engineers for different purposes - e.g. detailed nonlinear analysis of selected parts of the structure after performing a more general 3-D linear analysis of the full structure, and (3) downstream use of structural analysis data for various purposes. Specifically, within ST-4 *steel detailing* is being considered, but the procedure is similar for reinforced concrete or wood structures, as well as for some other related activities. Typical applications involved are CAD and Finite Element Analysis (FEA) systems, where FEA collectively describes applications based on the finite element method which require structural data as input.

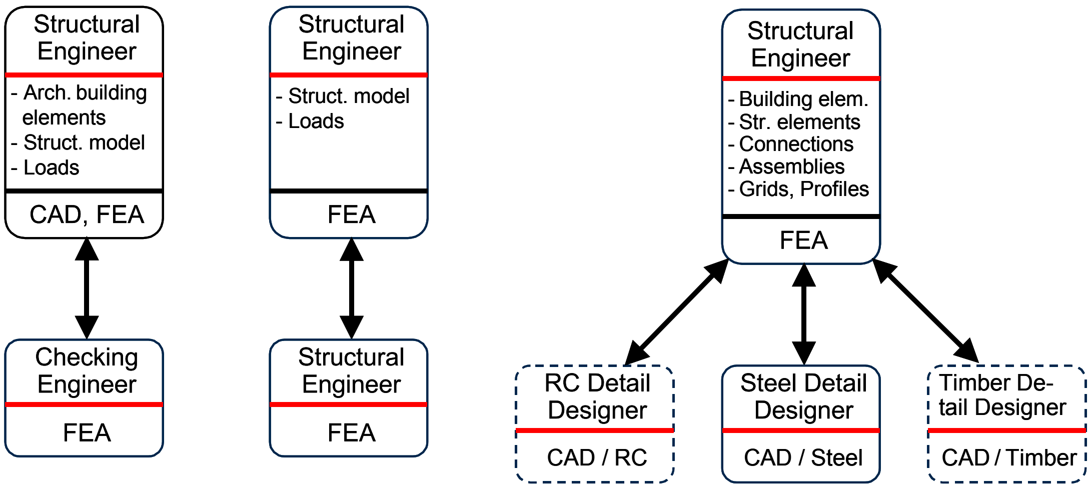

FIG. 11: Typical scenarios within the structural engineering domain and for the structural detailing process.

Especially in the last scenario (FIG. 11, right), the *raw* outline design of a steel structure is represented as an object model generated by a CAD system at a low level of detail. Fasteners, connection elements, processing features etc. are not part of that raw design. Interoperability from structural analysis to CAD and vice versa is possible by agreement that a CAD application can properly interpret the structural data. While such capabilities will be defined later in a special view, the ST-4 specifications provide the necessary prerequisites to enable the process.

FIG. 12 shows some intended interoperability with other domains. That consideration of inter-domain issues has had consequences on the whole model design, as discussed in the previous chapter, but aspects related to the definition of views are treated basically in the same way as by the purely "structural" use cases.

Note, however, that view definition a là IAI are currently insufficiently formalized. More efforts are needed to make such views unambiguously "understandable" and "implementable" for application developers. This is ongoing work both in the IAI, the BLIS project and the ST-4 project itself.

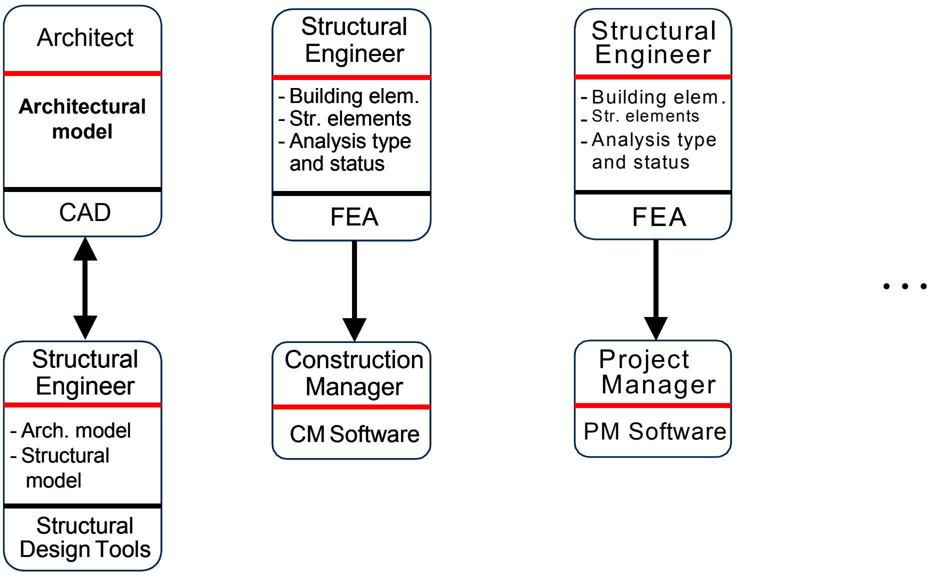

FIG. 12: Interoperability with other domains.

6. ILLUSTRATIVE EXAMPLE

To complete the presentation of the developed structural analysis extension model, in this chapter a small example is discussed. It should help to understand the usage of the extension model as well as to get an idea of its capabilities from a more practical point of view. The example itself is not related to any of the specific views that were briefly introduced in the previous chapter. It shows the integration of structural data by adding them to already existing IFC data as proposed by the IAI. Thereby it describes a view to the whole model, illustrating the *ideal* connection to existing 'architectural data'.

6.1 Background

Before going into the details of the example, in this section some relevant background information will be given to clarify issues that may be misinterpreted.

At first it has to made clear that the IFC model is a container to collect all relevant building data. For our extension approach it means to describe *all* needed structural aspects, i.e. the used structural analysis models, within a particular '*project repository*'. This is motivated by the possibility to maintain different structural models as needed to define different structural problems or alternative representations. The structural engineer should not be forced to use spatial structures (as it is frequently assumed) only because there exists a 3D architectural model as a background. Rather, it should be possible to define both planar and spatial structural models, and have the freedom to select the best fitting structural representation(s) for each individual building element, depending on the specific analyses needed. This means that types of building elements should not enforce a predefined structural representation. For example, walls do not necessarily have to be modelled as planar structural elements. If appropriate, they can also be defined as linear structural elements (see also section 4.1).

Thus, the 'project repository' concept provides added value for describing structural analysis models which should be taken into account by using the model extension. It helps to realize dependencies between different structural models as well as to the architectural view of a building. It can also support useful data navigation options, for instance from building elements to their structural representations, or from the building structure as a whole to detailed structural problems.

Another aspect of applying the model extension is the use of existing concepts of the IFC modelling framework. New entities are mostly integrated by deriving them from existing entities. That fulfils the requirement from the IAI to reuse as much as possible already defined features. Most interesting for ST-4 are in that respect the classes *IfcRoot*, *IfcObject*, *IfcProduct* and *IfcSystem*. Additionally, the entity class *IfcProject* is widely used (as a key object for each 'project repository' capturing important data like project hierarchy or used measures and units). A more detailed specification of these features is provided in (Wix & Liebich 2001) and in (Liebich 2002).

6.2 Overview of the given example

FIG. 13 shows a simplified view of the 'architectural' building elements for the presented example. As mentioned before, the structural engineer now has to determine the bearing structure and may eventually add or change the architectural data before proceeding with the definition of appropriate structural analysis model(s) in accordance with the ST-4 model extension.

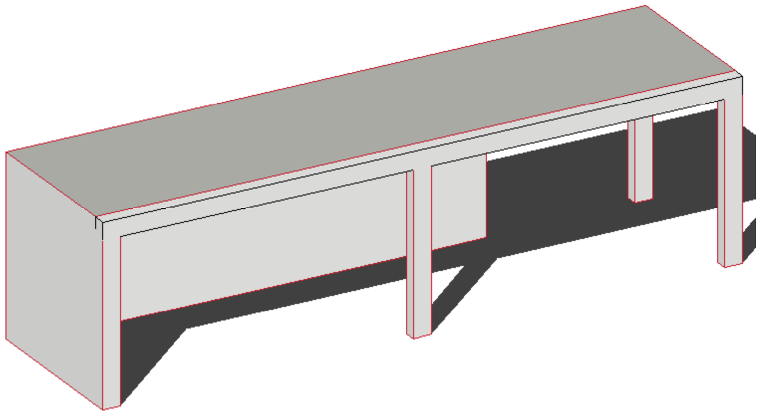

FIG. 13: 'Architectural design' of the example

Here it has to be noted that the examination and structuring of architectural data, and the kind of support provided by the ST-4 extension model were controversially discussed within the ST-4 group. Finally it was decided to use the existing general grouping mechanisms to define the bearing structure with its component building elements and to add possibilities to represent such kind of data explicitly in the next version of the extension model. More detailed information dealing with that topic is given in (Weise et al. 2000), which also discusses possible extensions to include foundation elements, and in Volume III of the ST-4 documentation.

6.3 Structural representation of building elements

Having in mind the configuration approach presented in section 4.2 above the example starts with the definition of structural elements, supports and their connections. These elements must be defined in the 3D space, taking in consideration the structural analysis model(s) in which they shall be used. Very important, especially for the configuration of planar structural models, is the appropriate definition of the local coordinate system and the proper assignment of all needed properties (material, cross sections, connection and support properties). FIG. 14 shows two structural analysis models that are of interest for further elaboration.

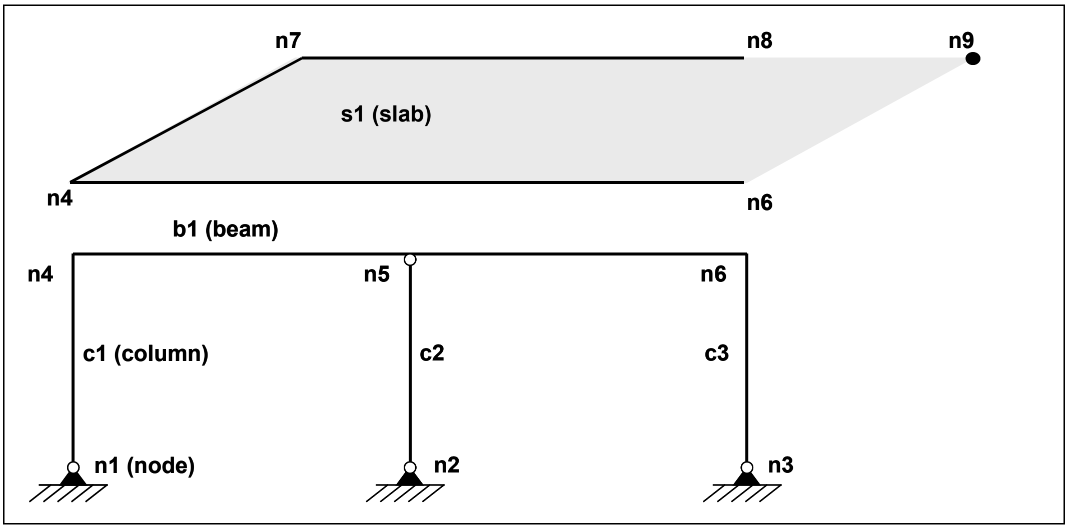

FIG. 14: Derived structural analysis models

All structural elements have to be modeled using an appropriate structural representation. We start with the nodes (n1 to n9) that are the basis for the topological specification. They are represented by objects of type *IfcStructuralPointConnection*. Using the defined nodes linear elements, such as columns and beams (*IfcStructuralCurveMember*), and linear supports (*IfcStructuralCurveConnection*) can then be defined. Finally, the slab s1 is represented as a planar element by an object of type *IfcStructuralFaceMember*. The topology itself is defined by using the 'Representation' attributes inherited from *IfcProduct* which reference respective objects of the topology resource. The latter define both the connectivity of the structure and their own geometry representation. This leads to the usage of *IfcVertexPoint*, *IfcEdgeCurve* and *IfcFaceSurface* instead of *IfcVertex*, *IfcEdge* or *IfcFace*. Typically, many of these objects will need to be reused to define the topology of the overall structure. For instance, in FIG. 14 the instances of *IfcVertexPoint* defining the locations of nodes n1 and n4 are respectively reused to define the connectivity of column c1.

After that, structural data like material and cross section properties as well as support conditions need to be specified. For that purpose, instances of the entity classes *IfcMechanicalMaterialProperties* (+ *IfcMaterial* and *IfcRelAssociatesMaterial*), *IfcStructuralProfileProperties* (+ *IfcRelAssociatesProfileProperties*), as well as *IfcBoundaryCondition* (and its respective subclasses) are used. These objects are related to the axes of the respective local element coordinate systems defined by means of the attribute 'ObjectPlacement' inherited from *IfcProduct*.

FIG. 15 shows the definition of column c1 (#400), node n1 (#430) and node n4 (#434) as an excerpt from a STEP physical file (ISO 10303-21 1996). By examining more closely the data for column c1 the following can be noticed:

- the local coordinate system is described by the objects #401, #402, #403, #404 and #405,
- the representation is given with the objects #406, #407, #408, #409, #410, #411, #412, #413, #414, #415 and #416,
- material properties are defined by #417 and the extension of #139,
- cross section properties are specified by #423 and #424,
- finally, the connection between the building element #124 and its structural representation #400 is established by the relationship #399.

A first impression here might be that these are 'a lot of objects'. However, many of them can be reused for the definition of other structural elements. In the provided example this is in fact not possible only for the objects numbered #406, #407, #409 and #412. All other objects are used multiple times in the full STEP physical file, and not only for a single structural element.

```
...
#124 = IFCCOLUMN ('3E0jDq$XH5HxNc4H9OtrTO', #6, $, $, $, #137, #135, $);
#138 = IFCMATERIAL ('reinforced concrete');
#139 = IFCRELASSOCIATESMATERIAL ('0vZC4S0914RPEt9JnGG0PV', #6, $, $, (#124, #400), #138);

//extension using ST-4 concepts
#399 = IFCRELASSIGNSTOSTRUCTURALMEMBERS ('internal_ID_of_rel1', #6,
        'connection between column 1 and curve member 1', $, (#400), .PRODUCT. , #124);
//column 1
#400 = IFCSTRUCTURALCURVEMEMBER ('internal_ID_of_S1', #6, 'column 1', $, $, #401, #406,
        RIGID_JOINED_MEMBER.);
#401 = IFCLOCALPLACEMENT ($, #402);
#402 = IFCAXIS2PLACEMENT3D (#403, #404, #405);
#403 = IFCCARTESIANPOINT ((0., 0., 0.));
#404 = IFCDIRECTION ((1., 0., 0.));
#405 = IFCDIRECTION ((0., 0., 1.));
#406 = IFCPRODUCTREPRESENTATION ($, $, (#407));
#407 = IFCTOPOLOGYREPRESENTATION (#408 , $, $, (#409));
#408 = IFCREPRESENTATIONCONTEXT ('Mechanical Structure', 'Design');
#409 = IFCEDGECURVE (#410, #411, #412, .T.);
#410 = IFCVERTEXPOINT (#413);
#411 = IFCVERTEXPOINT (#414);
#412 = IFCLINE (#413, #415);
#413 = IFCCARTESIANPOINT ((0., 0., 0.));
#414 = IFCCARTESIANPOINT ((0., 0., 2.85));
#415 = IFCVECTOR (#416, 2.85);
#416 = IFCDIRECTION ((0., 0., 1.));
//defining additional structural properties
#417 = IFCMECHANICALMATERIALPROPERTIES (#138, $, 30000., $, $, $);
#423 = IFCSTRUCTURALPROFILEPROPERTIES ('profile type 1', $, $, $, $, $, 0.09, $, $, 67500.,
        $, $, $, $, $, $);
#424 = IFCRELASSOCIATESPROFILEPROPERTIES ('internal_ID_of_rel2', #6,
        'cross section properties for column 1', $, (#400), #423, $);
//node 1
#430 = IFCSTRUCTURALPOINTCONNECTION('internal_ID_of_K1', #6, 'node 1', $, $, #401, #431, #433);
#431 = IFCPRODUCTREPRESENTATION ($, $, (#432));
#432 = IFCTOPOLOGYREPRESENTATION (#408 , $, $, (#410));
#433 = IFCBOUNDARYNODECONDITION ('rigid support', $, $, $, $, $, $);
//node 4
#434 = IFCSTRUCTURALPOINTCONNECTION('internal_ID_of_K4', #6, 'node 4', $, $, #401, #435, $);
#435 = IFCPRODUCTREPRESENTATION ($, $, (#436));
#436 = IFCTOPOLOGYREPRESENTATION (#408 , $, $, (#411));
…
```

FIG. 15: Excerpt from a STEP physical file showing the definition of structural elements.

Up to now the definition of connection properties is missing. These can be defined by using objects of type *IfcRelConnectsStructuralMember* and their respective subclasses and referenced resource objects, capturing the properties themselves. In the given example the connection between the node n1 and column c1 is established by objects of type *IfcRelConnectsMemberToPoint* and *IfcBoundaryNodeCondition.* All other connections are defined in a similar way. These shall not be discussed in further detail as the principle should be generally clear. More details can be found in Volume III of the ST-4 documentation (cf. IAI ST-4 2002).

Finally, the relationships between building elements and their structural representation has to be mentioned. They accomplish the integration into the 'project repository'. The cardinality of such relationships can be 1:1, 1:n and m:n. However, as the object class *IfcRelAssignsToStructuralMembers* defined for that purpose can handle only 1:1 and 1:n relationships, in the last case m relationship objects need to be used. These may represent different semantics. For example, in the case of m:1 one structural element may stand for equally loaded building elements, or just define the structural behaviour for a group of building elements. Currently such differences have to be tackled by the applications.

6.4 Load definitions

The load definitions for a particular structural analysis model include both individual 'load elements' and their grouping to load cases and load combinations. FIG. 16 below shows the loads for the discussed example.

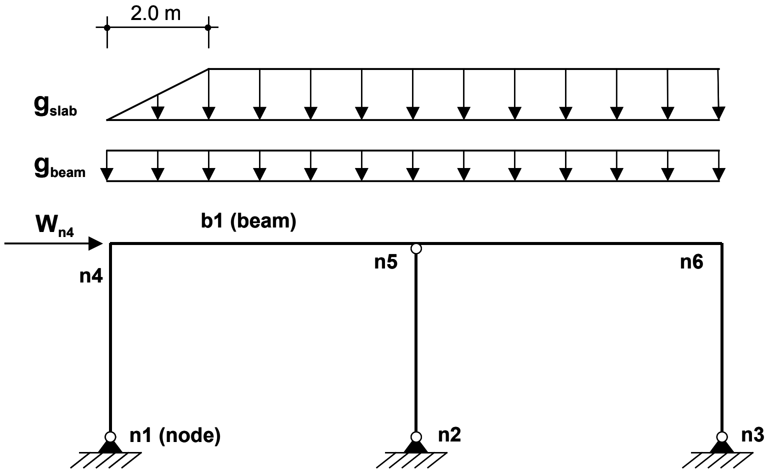

FIG. 16: Loadings for the example structure.

The definition of load elements follows the idea of using topological objects to specify the load locations. In accordance with that, the point load w_{n4} shown on FIG. 16 uses the *IfcVertexPoint* already used by node n4 and column c1 (see #411 in FIG. 15), and the linear loads g_{beam} and g_{slab} use the *IfcEdgeCurve* of beam b1 respectively. These topological objects are referenced by a load element representing the dimension of the load, i.e. point loads are of type *IfcStructuralPointAction*, linear loads are of type *IfcStructuralLinearAction* or *IfcStructuralLinearActionVarying,* and planar (or surface) loads are of type *IfcStructuralPlanarAction* or *IfcStructuralPlanarActionVarying*. Besides the definition of the load locations they define also the load values by reference to objects of the load resource related to the local or the global coordinate system. In the case of linear and planar loads using the global coordinate system the load values can be given in relation to the true or the projected element length. The resource objects themselves can be shared to provide for more efficient data structuring and to enable grouping. The latter can be useful for example in the case of changing load values for a set of equally loaded elements. Without going into great detail, on FIG. 17 the load definition of w_{n4} and its connection to node n4 using the relationship *IfcRelConnectsStructuralActivity* is shown. This relationship is also useful to specify load eccentricities.

```
//wind on node 4
#1000 =  IFCSTRUCTURALPOINTACTION ('internal_ID_of_L1xxxxx', #6, 'wind on node 4', $, $,
              #401, #435, #1001, .GLOBAL_COORDS., .F., .TRUE_LENGTH., $);
#1001 =  IFCSTRUCTURALLOADSINGLEFORCE ('wind', 2400., $, $, $, $, $);
#1002 =  IFCRELCONNECTSSTRUCTURALACTIVITY ('internal_ID_of_rel11xx', #6, 'wind on node 4', $, #434, #1000);
```

FIG. 17: Definition of load w_{n4}.

When all load elements are defined, they have to be grouped to load cases and load combinations. For that purpose a generic class has been defined to enable the description of load cases and combinations in a recursive manner. The *IfcStructuralLoadGroup* class itself is specialised from *IfcGroup* and uses the already existing generic class *IfcRelAssignsToGroup* to refer to the load elements (subtype of *IfcStructuralActivity*) and/or other load groups assigned to it. Additionally, a load type can be selected and a coefficient for its usage can be given.

The proposed nesting of load groups is often useful and even necessary for load combinations that are logically a group of load cases. To avoid incorrect usage, the nesting mechanism is constrained depending on the load group type. This approach can shortly be expressed as follows:

- a ‘load case’ is defined by load elements,
- a ‘load combination group’ (as an intermediate step for defining load combinations) is defined by ‘load cases’ having the same coefficient for a particular combination, and finally,
- a ‘load combination’ is defined by ‘load combination groups’.

More detailed explanations are provided in Volume III of the ST-4 documentation (cf. IAI ST-4 2002).

6.5 Configuration of structural analysis models

In a last step, the structural analysis models must be 'configured' by using the previously discussed structural objects. For that purpose, the grouping mechanism inherited from *IfcGroup* is used to collect the relevant data, i.e. structural items (subtypes of *IfcStructuralMember* and *IfcStructuralConnection*) and structural activities (subtypes of *IfcStructuralAction* and *IfcStructuralReaction*). However, instead of using the grouping mechanism, load and result groups are simply referenced through the optional attributes 'LoadedBy' and 'HasResults' (see FIG. 9).

In addition to the outlined configuration of structural objects the *type of the structural analysis model* must be specified. Possible options here are 2D loading (further subdivided in 'in plane loading' and 'out of plane loading'), and 3D loading. For 2D loadings the considered plane can be explicitly defined to enable automated checking of the consistency of the configuration against its defined structural properties. A possible definition for the frame from FIG. 14 is provided on FIG. 18 below. Here, along with the definition of the 2D plane (#1501, #1502, #1503 and #1504) and the configuration of structural items and activities (#1510), the connection to the 'serviced' building is also established by means of an object of type *IfcRelServicesBuildings* (#1520).

```
//definition of frame 1
#1500 =  IFCSTRUCTURALANALYSISMODEL ('internal_ID_of_AM1xxxx', #6, 'frame 1', $, $, .IN_PLANE_LOADING_2D.,
              #1501, (#1050, #1082, …), (#1100, …));
#1501 =  IFCAXIS2PLACEMENT3D (#1502, #1503, #1504);
#1502 =  IFCCARTESIANPOINT ((0., 0., 0.));
#1503 =  IFCDIRECTION ((0., 1., 0.));
#1504 =  IFCDIRECTION ((1., 0., 0.));
//grouping of structural items and structural activities
#1510 =  IFCRELASSIGNSTOGROUP ('internal_ID_of_rel22xx', #6, 'frame 1', $,
              (#400, …, #430, #434, … , #1000, …), .NOTDEFINED., #1500);
//connection to building
#1520 =  IFCRELSERVICESBUILDINGS (('internal_ID_of_rel23xx', #6, 'connection of frame 1 with building', $, #1500,
              (#30));
#30 =    IFCBUILDING ('1XQgykdSzBLO_sQs6Pktvb', #6, 'Building of the ST-4 example 1',
              'Default instance which contains no further information', $, #29, $, $, .ELEMENT., $, $, $);
```

FIG. 18: Configuration of structural objects to a structural analysis model.

The described procedure presents a typical sequence of work and can thus be used as a template for many situations.

7. STATUS AND FUTURE PLANS

The ST-4 project aiming to develop a model extension in the domain of structural analysis and steel construction is now in its final stage. By the time of this writing, the internally released model extension has been handed over to the IAI Modeling Support Group (MSG) which harmonizes the defined concepts with other extension efforts within the IAI, namely the projects ST-2 "Reinforced Concrete Structures and Foundations" (led by the IAI Japan Chapter) and ST-3 "Precast Concrete Construction" (led by the IAI Nordic Chapter). It is expected that the proposed extension will be integrated in the next version of IFC, i.e. the IFC 2x Edition 2 (IFC2x2) which is scheduled for early summer 2003.

The German working group and the authors will further support the development of the ST-4 extension project until the final release to provide reviews and feedback.

After summer 2003 the start of commercial implementations by software vendors in the structural engineering domain is envisaged. However, several interesting research topics will still remain, such as the support of collaborative work through more robust model servers and web services, the implementation and management of the bi-directional links between two physically separate models (architecture and structural engineering), and the life-cycle aspects of the work. Automated code checking of structural plans to support the work of checking engineers is another important area for further work.

Some of these issues are addressed in the projects ISTforCE and iCSS (cf. Katranuschkov et al. 2001; Juli & Scherer 2002), but at the development time of these projects an enhanced IFC structural model had not yet been available. There is still space for more to come.

8. CONCLUSION

The inauguration of the ST-4 group consisting of industry and research partners underpins the overall interest in creating a widely accepted project data standard for AEC/FM. The current development status and the expectations to the IFCs have motivated the undertaken activities for its extension in the field of steel constructions and in the presented structural analysis domain. The latter is seen not only as an individual effort within the specific area of interest but also as an important building block for the harmonisation of all structural projects within the IAI and the integration of the structural domain with many other evolving domain models.

The results provided in this paper give an insight into the work of the ST-4 group, the scope and the rationale of the model, the major modelling concepts used and top-level entities defined, and the basic implementation issues to be considered.

It was shown that a coherent domain extension of the IFCs can be effectively achieved, even for specialized domains where considerable semantic differences to the core IFC elements exist. The gained experience verified that a lot of concepts can be reused or extended by following the idea of the given classification. This is especially the case for ‘resource objects’ provided that they are part of a shared repository for the whole model. The classes of the domain extension themselves are mostly specialised from core classes but set up a separate classification which is then interlinked through objectified relationships with the architectural data. This principle of separating the building elements from the mechanical representations allows a very flexible usage of the model extension.

The scope of the presented extension is limited to functionalities that balance added capabilities with a reasonable complexity of the model. However, the specification also considers as much as possible further extensions, as e.g. for concrete and wood structures, or for the detailing of steel constructions. Additionally, a more sophisticated connection to architectural data is envisaged which will highly improve the cooperation between architects and structural designers. Thus, a first step has been made towards practical implementations. Other steps have to follow to bring the vision of the IAI more and more to reality.

9. ACKNOWLEDGEMENTS

The presented work has been enabled through the support of the EU ISTforCE project (IST 1999-11508) and the German iCSS project. The contribution of the funding agencies (EC, BMBF) is gratefully acknowledged. The active cooperation of all members of the German Working Group and of the other IAI Structural projects has also greatly stimulated the development process. These efforts are highly appreciated.

10. REFERENCES

DSTV 2000. Standard Description Product Interface Steel Construction. EDP Working Group of DSTV, Vers. April 2000. German Steel Construction Assoc. (DSTV), Düsseldorf, Germany. Complete documentation available from: http://www.deutscherstahlbau.de (partially in German).

IAI R3 2002. The IFC extension projects – overview. Available from: http://cic.vtt.fi/niai/technical/ifc_3.

IAI ST-4 2002. The ST-4 Project "Structural Analysis Model and Steel Constructions", Volumes 0, I, II, III, III (Example). Available from: http://cib.bau.tu-dresden.de/icss/structural_papers.

ISO 10303-21 IS /Cor.1/ 1996. Industrial Automation Systems and Integration -- Product Data Representation and Exchange -- Part 21: Implementation Methods: Clear Text Encoding of the Exchange Structure, International Organisation for Standardisation, ISO TC 184/SC4, Geneva.

ISO 10303-42 IS /Cor.1-2/. 1999. Industrial Automation Systems and Integration -- Product Data Representation and Exchange -- Part 42: Integrated Generic Resources: Geometric and Topological Representation, International Organisation for Standardisation, ISO TC 184/SC4, Geneva.

Juli R. & Scherer R. J. 2002. iCSS – Ein integriertes Client-Server System für das virtuelle Planungsteam. VDI-Berichte 1668. VDI Verlag, Düsseldorf, Germany (in German).

Katranuschkov P., Scherer R. J. & Turk Z. 2001. Intelligent Services and Tools for Concurrent Engineering - An Approach Towards the Next Generation of Collaboration Platforms, ITcon Vol. 6, special issue: “Information and Communication technology advances in the European construction industry”. Available from: http://www.itcon.org, (2001).

Liebich T. 2002. IFC 2x Model Implementation Guide, © International Alliance for Interoperability, Modeling Support Group. Version 1.4 of July 5, 2002. Available from: http://www.iai-international.org/iai_international/Technical_Documents/iai_documents.html

SABLE. 2002. The SABLE Project – Project Description. Available from: http://www.blis-project.org/~sable.

Scherer R. J. (ed.) 2002. A Concurrent Engineering Collaboration Platform for Multi-Project Participation, Second Annual Report, Deliverable D13, EU Project IST-1999-11508 ISTforCE.

Weise M., Katranuschkov P. & Scherer R. J. 2000. A Proposed Extension of the IFC Project Model for Structural Systems. In: Goncalves R., Steiger-Garcao A. & Scherer R. J. (eds.) Product and Process Modelling in Building and Construction, Proceedings of the 3rd ECPPM, Lisbon, 25-27 Sept. 2000. Balkema, Rotherdam, the Netherlands, pp. 229-238.

Weise M., Katranuschkov P. & Liebich T. 2002. Structural Analysis Extension for the Next IFC Release. In: Turk Ž. & Scherer R. J. (eds.) Product and Process Modelling in Building and Construction, Proc. of the 4th ECPPM, Portorož, Slovenia, 9-11 September 2002. Balkema, Rotherdam, the Netherlands, pp. 379-386.

Wix J. (ed.) 2001. IFC 2x Extension Modelling Guide, © International Alliance for Interoperability. Available from: http://cig.bre.co.uk/iai_international/Technical_Documents/documentation/.

Wix J. & Liebich T. 2001. Industry Foundation Classes IFC 2x, © International Alliance for Interoperability. Available from: http://www.iai-ev.de/spezifikation/IFC2x/index.htm

All URL references last accessed in Oct. 2002.