

# AN IFC-BASED PRODUCT MODEL FOR RC OR PC SLAB BRIDGES

Nobuyoshi Yabuki, Ph.D., P.E. and Tomoaki Shitani  
*Department of Civil Engineering and Architecture, Muroran Institute of Technology*  
*yabuki@news3.ce.muroran-it.ac.jp*

## SUMMARY

The authors developed a product model for reinforced concrete (RC) or prestressed concrete (PC) slab bridges on the basis of the Industry Foundation Classes (IFC) in order to enable interoperability of various application systems throughout the lifecycle of structures. Based on the analysis of the characteristics of RC or PC slab bridges, new classes for slabs, prestressing strands, sheaths, voids, rebars, and anchoring devices were defined. A feature of the model is that it clearly represents the relationship that the concrete slab contains elements such as rebars, prestressing strands, and voids by representing the concrete slab as a spatial structure element by B-rep. The ifcXML was selected to implement the developed product model after comparing with other XMLs. The authors implemented the schema of the product model and an instance of a prestressed concrete hollow slab bridge using ifcXML. To check the validity and practicality of the model, the product model was integrated with three application systems, by developing converter programs. The application test case shows improved efficiency and validity of the system.

## INTRODUCTION

Research and development on product models have been carried out in order to enable the interoperability of various application systems in a lifecycle of products and structures. Various standards and specifications for product models such as ISO (International Organization for Standardization) 10303 (ISO, 1994) known as STEP (Standards for The Exchange of Product model data), Industry Foundation Classes (IFC) of International Alliance for Interoperability (IAI, 2002), CIMSteel Integration Standards (Crowley et al, 2000), etc., have been proposed. A number of related research projects such as (Faraj et al, 1999) have been executed. The authors developed their product models for steel frames, penstocks, and prestressed concrete hollow slab bridges, and implemented them using eXtensible Markup Language (XML) to demonstrate the efficiency improvement by achieving the interoperability among application systems including 3D-CAD, code checking, quantity calculation, cost estimation, scheduling, and inspection for maintenance (Yabuki et al, 2001a) (Yabuki et al, 2001b) (Yabuki et al, 2002). However, the modelling approach employed then was a classical one that each class contains its all attributes inside of the class but not having its property classes outside.

Since a standard modelling methodology has not been established yet, different methods and approaches have been employed by various organizations for their objectives. Furthermore, there are various XML schemata and other languages for implementing product models. Therefore, the authors believe that it is important to develop a standard approach for both modelling and implementation for future development of product models for various products and structures.

In this research, IFC has been selected as a base for developing product models for civil engineering infrastructure such as bridges since IAI is an international organization aiming at ISO standards for buildings, which are similar to the infrastructure. This paper presents the IFC-based product model for reinforced concrete (RC) and prestressed concrete (PC) super-structures of bridges, implementation issues and results, and a system for data exchange operation among several systems.

## IFC

### Basic elements of IFC2x

The current version of IFC is Release 2x and is called IFC2x. In IFC2x, the shape representation of members, definitions of attributes such as elastic modulus, definitions of relationships among members



are represented in the following four classes: IfcObject, IfcPropertyDefinition, IfcRelationship, and IfcRepresentationItem (Figure 1).

IfcObject class and its sub classes define physically existing objects, including beams, columns, spaces, and sites. For classes such as beams and columns, geometric information about objects can be represented as attributes of IfcProduct class, which is a sub class of IfcObject. IfcPropertyDefinition class and its sub classes can contain supplementary information, which is not inherited in IfcObject class and its sub classes. For example, material properties such as elastic modulus and yield stress of the IfcColumn class are defined in the corresponding IfcPropertySet class. By separating the object classes and properties in this way, the model can obtain the flexibility that plural objects can have the same property set and that a single object class can have plural property sets. IfcRelationship class and its sub classes define the relationship among objects and property sets. Each relationship has its own specific relationship in a sub class of IfcRelationship. Although IfcObject, IfcPropertyDefinition, IfcRelationship classes are defined as sub classes of the IfcRoot class, IfcRepresentationItem class does not share the root with other three classes as shown in Figure 1. IFC2x provides resources on geometric information in the IfcRepresentationItem class and its sub classes separately as shown in Figure 2.

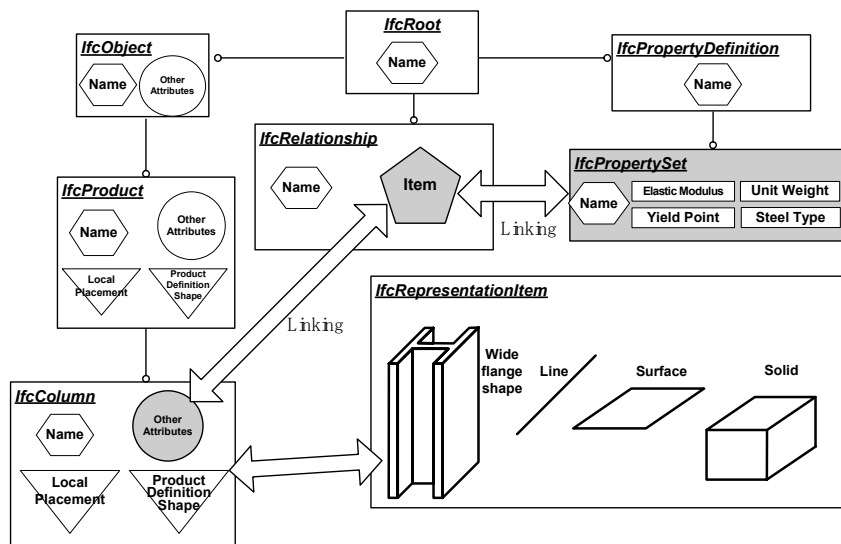


Figure 1 Conceptual diagram for representing relationships among classes

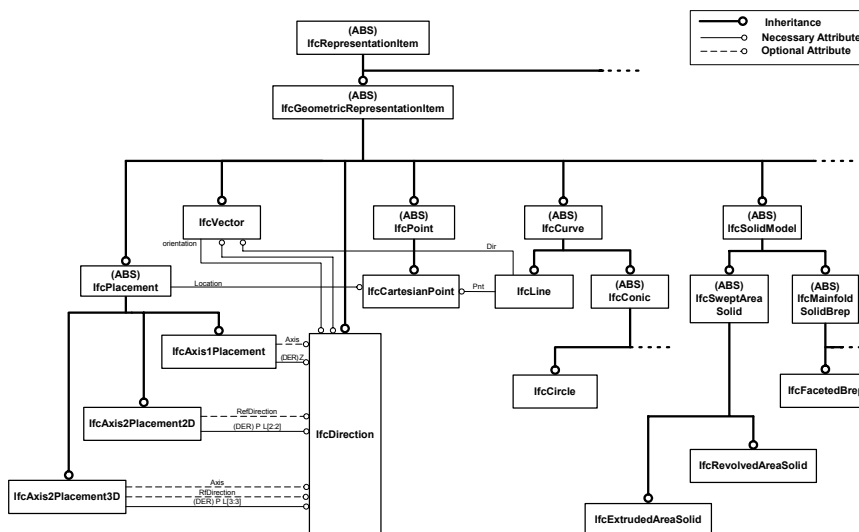


Figure 2 IfcRepresentationItem class and its sub classes

## Characteristics of IFC2x

IFC2x provides building elements such as beams, columns, doors, windows, etc., in the sub classes of `IfcBuildingElement` class. One of the significant differences in IFC2x compared to IFC2, the previous version, is that properties of objects are separated from the object classes and are defined in property sets. IFC2x provides property sets for doors and windows but not for beams, columns, etc., yet. Users can make their own property sets for these objects by using `IfcPropertySet` class. In this case, the object class and its property set class are linked by `IfcRelDefinesByProperties` class.

Each element class has attributes called `ObjectPlacement` and `Representation`. The attribute `ObjectPlacement` contains a class called `IfcAxis2Placement3D` that represents a location. The attribute `Representation` contains a class called `IfcRepresentation`, which contains a class `IfcRepresentationItem` that represents geometry of vectors, points, lines, curves, surfaces, solids, etc.

In IFC2x, internal elements, e.g., rebars, which are inside a member, can be represented as properties of the member such as beams and columns. However, classes for representing internal elements are not defined in IFC2x. Thus, it is very difficult to represent the detailed 3D geometric information of internal elements as objects. In order to check the interference among internal elements including rebars, pipes, cables, sheaths, etc. and covering of rebars in a pre-construction or construction stage, 3D geometric models of internal elements would be necessary.

## A PRODUCT MODEL FOR RC AND PC BRIDGES

### Concept for modelling

Since IFC is developed for modelling buildings, it is difficult to directly apply it to bridges. But it is inefficient to develop a completely new model for bridges from scratch. As IFC is developed to become an international standard in ISO, it is advantageous to develop product models based on IFC from an international standard point of view. Thus, the authors have decided to develop our bridge product models based on IFC2x, keeping its basic structure, adding only necessary classes, while having generality to apply it to other kinds of infrastructure. And a type of prestressed concrete hollow slab bridges has been selected as a sample for validation and demonstration.

Prestressed concrete hollow slab bridges consist of members such as concrete, voids (hollow pipes), prestressing strands, anchoring devices, sheaths, rebars, etc. One of the issues is the concrete slab “contains” voids, prestressing strands, anchoring devices, sheaths, and rebars. And since rebars have high geometric freedom and have embodiment lengths and joints as properties, the issue is how rebars should be represented in the model. In the following subsections, we discuss these issues.

### Concrete members

Concrete members in bridges have more geometric freedom than typical building concrete members such as beams, columns, walls, and slabs. And since concrete members contain rebars, voids, sheaths, etc., if we define concrete members as perfect solids, we have to subtract contained members, which is cumbersome. On the other hand, if we define a concrete member as a set of single surfaces, it is difficult to apply 3D finite element mesh generation and quantity calculation to concrete members. Thus, the authors represented a concrete member as a simple solid model comprised of a set of surfaces having a property of inside or outside of the member in our product model. And, contained members clearly indicate that they are “contained” in the concrete member.

In IFC, the `IfcRelationship` class has a sub class named `IfcRelContainedInSpatialStructure` (**Figure 3**). This class is used to represent that `IfcBuildingElement` members such as `IfcBeam`, `IfcColumn`, `IfcSlab`, etc., are “contained” in the `IfcBuilding` class as shown in **Figure 4**. Further, the basic geometry of `IfcBuilding` is represented in `IfcFacetedBrep` (Brep or B-Rep), which is a closed solid comprised of a set of surfaces and which can store the information of the inside. Thus, Brep satisfies the two conditions described above.

Therefore, as the concrete structure of a PC hollow slab bridge has the same characteristics of IfcBuilding, a new class SlabOfBridge was defined in the same story of IfcBuilding. Furthermore, we defined new member classes, Rebar, AnchoringDevice, Void, PrestressingStrand, and Sheath were

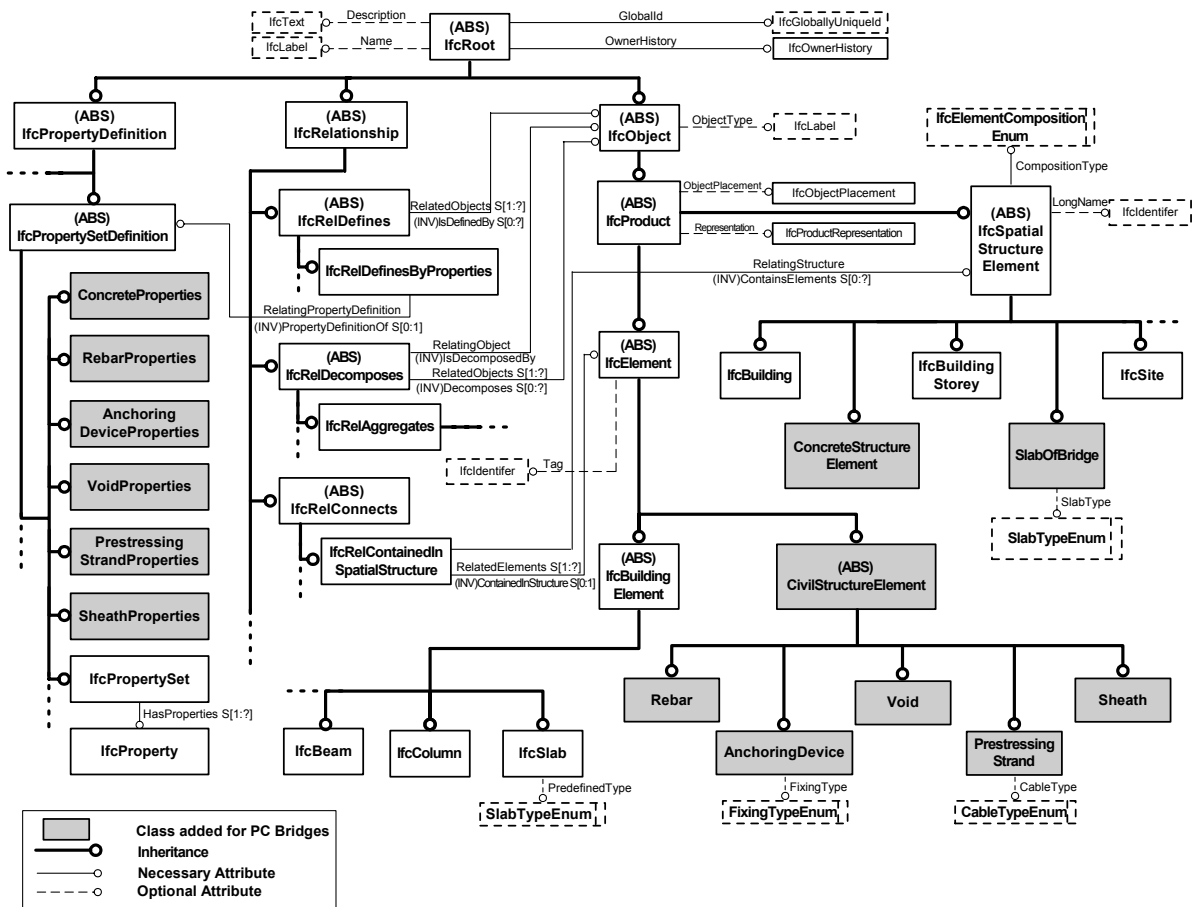


Figure 3 Class model developed in this research in Express-G

also defined in the same story of IfcColumn and IfcBeam. The authors also defined new property sets to represent the property information such as material characteristics of concrete, rebars, prestressing strands. Each property set and the corresponding member class are linked by the class IfcRelDefinesByProperties.

### Rebars

In our product model, each rebar is represented as an object. The geometry of a rebar can be represented by extruding a circle to a direction expressed in a vector or revolving the circle in a curve, as IfcExtrudedAreaSolid or IfcRevolvedAreaSolid, respectively.

As the anchorage part of rebars usually has no difference in appearance, the authors defined data such as embodiment length, location, type, etc., in a property set. A lap splice of rebars is not represented as two bars in our model but is represented as a part of continuous bars having a property that the part is a lap splice. Other attributes of rebars such as rebar type, nominal name, elastic modulus, etc., are defined in the property set (Figure 5).

### Other members

We defined the class of prestressing strands were defined in the same manner for rebars. The properties except the geometry of prestressing strands are represented in the property set, PrestressingStrandProperties. Thus, real property data can be stored in the instance of PrestressingStrandProperties class. However, not all data should necessarily be stored in the property set. Since data such as tensile load, elongation, relaxation, nominal area, unit weight, etc. are automatically designated by specifying the steel type and nominal name, such data can be retrieved

from a linked steel database system. Other member classes and property sets were also defined for voids, sheaths, etc.

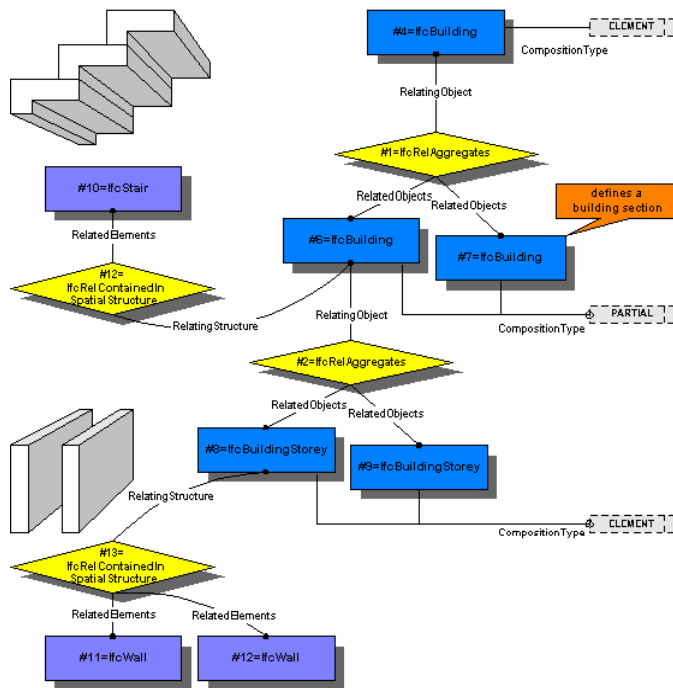


Figure 4 IfcRelContainedInSpatialStructure

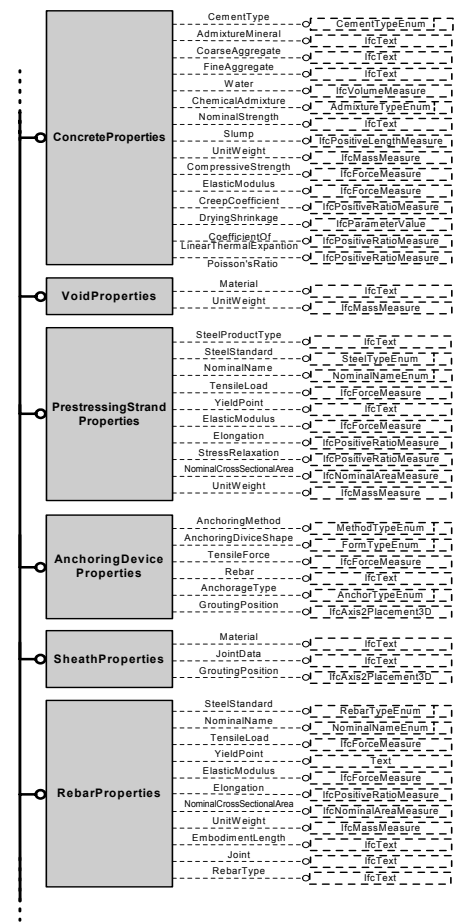


Figure 5 Property set for new defined members

## IMPLEMENTATION ISSUES

Although schemata of product models are to be defined in EXPRESS in ISO STEP, XML is widely used for implementing product models. In this section, we compare and discuss the following three XMLs: aecXML (aecXML, 2002), BLIS-XML (BLIS-XML, 2002), and ifcXML (ifcXML, 2001).

### aecXML

The aecXML is an XML schema that has been developed by IAI North America Chapter for the field of Architecture Engineering and Construction/Facility Management (AEC/FM). Terminology, grammar, and layout for business messages in AEC/FM are defined in the aecXML. Generally, the aecXML have been developed for the documentation of reports and catalogues but not for exchanging product model data.

### BLIS-XML

In BLIS (Building Lifecycle Interoperable Software) -XML, XML Data Reduced (XDR, 1998) is used as a schema. BLIS-XML schema can be translated to EXPRESS and the instance files are equivalent as Part21 files. Since BLIS-XML has an attribute-oriented structure and is easy for processing, we previously used it for implementing our product model for steel frames consisting of beams, columns, connections, etc.

However, BLIS-XML has a problem that we cannot verify the relationship between the schema and instances in terms of attribute type. For example, property data such as location of the IfcProduct class is represented as a class and its name is not expressed in the attributes of IfcProduct class but is limited

to express as "idref." Further, since BLIS-XML cannot represent the inheritance from a class to its sub classes, it is necessary to declare all attributes for each class, which is cumbersome.

## ifcXML

The ifcXML has been developed by IAI, and the XML Schema (XML Schema, 2002) is used as a schema. XML Schema is a relatively new schema language, which was recommended by World Wide Web Consortium (W3C) in May 2001. The ifcXML supports various data types unlike the Document Type Definition (DTD) for documents. Thus, we believe that it will be widely used as a format for various data exchange.

The ifcXML can be transformed to EXPRESS and the instance files are equivalent to Part21 files. Unlike BLIS-XML, ifcXML has a hierarchy-oriented structure and if a class has attributes, the class can specify the property class that the class has. And ifcXML can represent the inheritance relationship among classes. For these reasons, we have adopted ifcXML for implementing our product model.

## IMPLEMENTATION USING ifcXML

The authors implemented the schema of the product model for prestressed concrete hollow slab bridges using ifcXML, and then, developed an instance file for a real PC hollow slab bridge on the basis of the developed product model. **Figure 6** shows a part of the product model schema representing SlabOfBridge class. As shown in the figure, inheritance among classes and internal data structures can be identified more easily than BLIS-XML (**Figure 7**). **Figure 8** and **Figure 9** show parts of the instance files representing SlabOfBridge and Relationship, respectively. The lower part of **Figure 8** shows the data of Cartesian points determining the geometry of the slab of the bridge. **Figure 9** clearly shows that the IfcRelDefinesByProperties class links SlabOfBridge to ConcreteProperties and that the IfcRelContainedInSpatialStructure class links SlabOfBridge to members contained in the slab such as rebars.

Since ifcXML can represent inheritance among classes, the schema structure is simple and clear compared with BLIS-XML. Although the number of lines of ifcXML schema and instance files is much larger than that of BLIS-XML, the authors believe that the reliability of model data is improved because it becomes possible to verify the consistency between the schema and instances.

```

<xsd:complexType abstract="true" name="SpatialStructureElement">
  <xsd:complexContent>
    <xsd:extension base="Product">
      <xsd:sequence>
        <xsd:element type="Identifier" minOccurs="0" name="longName" />
        <xsd:element type="ElementCompositionEnum" name="compositionType" />
        <xsd:element name="containsElements" minOccurs="0" maxOccurs="unbounded">
          <xsd:complexType>
            <xsd:attribute ref="Inverse" />
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="SlabOfBridge">
  <xsd:complexContent>
    <xsd:extension base="SpatialStructureElement">
      <xsd:sequence>
        <xsd:element type="SlabTypeEnum" name="slabType" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

**Figure 6** A part of the product model schema in ifcXML

```

<ElementType name="SlabOfBridge" content="empty" model="open">
  <attribute type="XMLID" required="yes" />
  <attribute type="GlobalId" required="yes" />
  <attribute type="OwnerHistory" required="yes" />
  <attribute type="Name" required="no" />
  <attribute type="Description" required="no" />
  <attribute type="ObjectType" required="no" />
  <attribute type="IsDefinedBy" required="no" />
  <attribute type="HasAssociations" required="no" />
  <attribute type="HasAssignments" required="no" />
  <attribute type="Decomposes" required="no" />
  <attribute type="IsDecomposedBy" required="no" />
  <attribute type="ObjectPlacement" required="no" />
  <attribute type="Representation" required="no" />
  <attribute type="ReferencedBy" required="no" />
  <attribute type="LongName" required="no" />
  <attribute type="CompositionType" required="yes" />
  <attribute type="ContainsElements" required="no" />
  <attribute type="ElevationOfRefHeight" required="no" />
  <attribute type="ElevationOfTerrain" required="no" />
  <attribute type="BuildingAddress" required="no" />
  <attribute type="ServicedBySystems" required="no" />
</ElementType>

```

**Figure 7** A part of the product model schema in BLIS-XML

```

<SlabOfBridge id="_1001">
  <globalId|.CY[{}]+Hw=#rH0,xel</globalId>
  <OwnerHistory>
  <name>Original Bridge</name>
  <description />
  <objectType />
  <isDefinedBy inverse="_101" />
  <isDecomposedBy />
  <decomposes />
  <objectPlacement>
  <representation>
  <longName />
  <compositionType>COMPLEX</compositionType>
  <slabType>PCHollowSlabBridge</slabType>
  <containsElements inverse="_1000" />
</SlabOfBridge>
<CartesianPoint id="_1004">
  <coordinates>0,0,0</coordinates>
</CartesianPoint>
<CartesianPoint id="_1009">

```

**Figure 8** A part of the instance file representing SlabOfBridge

```

<RelDefinesByProperties id="_101">
  <relatedObjects>
    <SlabOfBridge href="_1001" />
  </relatedObjects>
  <relatingPropertyDefinition>
    <ConcreteProperties id="_202" />
  </relatingPropertyDefinition>
</RelDefinesByProperties>
<RelContainedInSpatialStructure id="_1000">
  <relatingStructure>
    <SlabOfBridge href="_1001" />
  </relatingStructure>
  <relatedElements>
    <Rebar href="_3001" />
    <Rebar href="_3002" />
    <Rebar href="_3003" />
    <Rebar href="_3004" />
    <Rebar href="_3005" />
    <Rebar href="_3006" />
    <Rebar href="_3007" />
  </relatedElements>

```

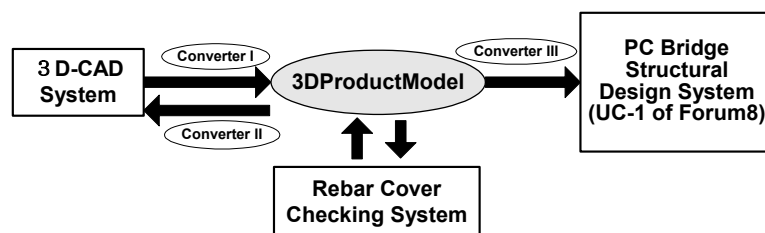
**Figure 9** A part of the instance representing the relationship

## INTEGRATION OF THE PRODUCT MODEL WITH APPLICATION SYSTEMS

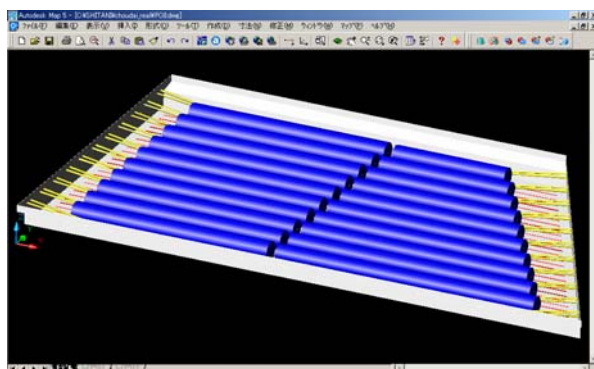
In order to check the validity and practicality of the developed product model, the product model was integrated with three application systems, i.e., 3D-CAD (AutoCAD 2002), a PC bridge structural design system (UC-1 of Forum8), and a rebar cover checking system by developing prototype data conversion programs (**Figure 10**), and they were applied to a design case.

The authors developed Converter Program I, which semi-automatically generates an instance file of the developed product model in ifcXML from an AutoCAD 3D product data, by using Visual Basic for Application (VBA). They also developed Converter Program II, which retrieves data from an instance file of the developed product model by using an XML parser and which renders the 3D model in AutoCAD automatically. Converter Program III bridges the product model and the design software (UC-1). Furthermore, they developed a rebar cover checking system by using Java Servlet and XML for Java Parser. This system directly reads an instance file of the product model and computes the minimum distance between each rebar and each surface of the concrete slab. Then, the minimum distance is smaller than the required cover, the system adds the information to the violated rebar in the product model.

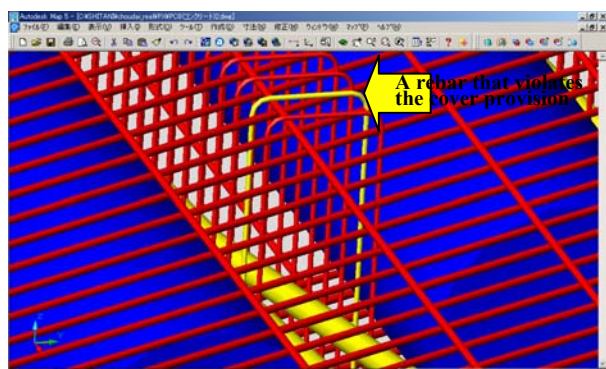
The application case is described in the following. First, a designer constructs a preliminary 3D CAD model of the PC hollow slab bridge (**Figure 11**). At this stage, each rebar is not modelled yet but only the data such as diameters and pitches of the rebars are assumed. Then, the instance file of the bridge is generated by executing Converter Program I, and the input data for the design checking system (UC-1) is generated by executing Converter Program III. The user executes UC-1 and checks the conformance of the design with design codes. As the design satisfies the codes, the user performs detailed design including selection and layout of rebars, and updates the product model instance data. Then, the user executes the rebar cover checking system and finds a violated rebar in the 3D-CAD system as shown in **Figure 12**. The user modifies the rebar and updates the product model data. This application case indicates the validity and practicality of the developed product model.



**Figure 10** An integrated system by the product model



**Figure 11** A PC hollow slab bridge



**Figure 12** Detection of a violated rebar

## CONCLUSION

In this research, the authors developed a product model for RC or PC slab bridges on the basis of IFC, and implemented the product model schema and instance by ifcXML. Then, the product model was integrated with three application systems. The contribution of this research is as follows.

- The authors have expanded the realm of IFC from buildings to RC and PC bridges.
- New classes for properly representing a slab and contained members such as rebars, prestressing strands, voids, etc., have been defined.
- A modern model developing technique, i.e., separating property sets from object classes rather than representing all attributes in product classes, was employed, which makes the model more flexible.
- The pros and cons of various XML were discussed and ifcXML has been evaluated as a most suitable XML schema for implementation of product models based on IFC.
- This research showed the validity and practicality of the product model by integrating three application systems using the developed product model and data conversion programs.

For future work, the authors intend to develop a process model and integrate it with the product model to apply them to a construction management system based on a progress payment method.

## ACKNOWLEDGMENTS

The authors would like to acknowledge the support and advices by Research Center for Land and Construction Management of the National Institute for Land and Infrastructure Management, Japan Prestressed Concrete Contractors Association, ST-4 Committee of Japan Chapter of IAI, and Forum8 Co., Ltd.

## REFERENCES

aecXML (2002) <http://www.iai-na.com/aecxml/mission.php>

BLIS-XML (2002) <http://www.blis-project.org/>

Crowley, A.J. and Watson, A.S. (2000) *CIMSteel Integration Standards*, The Steel Construction Institute.

Faraj, I., Alshawi, M., Aouad, G., Child, T., and Underwood, J. (1999) *Distributed Object Environment: Using International Standards for Data Exchange*, Journal of Computer-Aided Civil and Infrastructure Engineering, Vol. 14, pp.395-405.

IAI (2002) [http://www.iai-international.org/iai\\_international/](http://www.iai-international.org/iai_international/)

IFC2x (2000) [http://cic.vtt.fi/niai/technical/ifc\\_2x/](http://cic.vtt.fi/niai/technical/ifc_2x/)

ifcXML (2001) [http://www.iai-international.org/iai\\_international/Technical\\_Documents/documentation/IFCXML/ifcXML\\_language\\_binding\\_V1-02.pdf](http://www.iai-international.org/iai_international/Technical_Documents/documentation/IFCXML/ifcXML_language_binding_V1-02.pdf)

ISO 10303-1 (1994) *Industrial Automation Systems and Integration -- Product Data Representation and Exchange, Part 1: Overview and Fundamental Principles*, ISO

XDR (1998) <http://www.ltg.ed.ac.uk/~ht/XMLData-Reduced.htm>

XML Schema (2001) <http://www.w3.org/TR/xmlschema-0/>

Yabuki, N., Shitani, T., et al (2001a) *An Integrated Design System for Steel Connections*, Journal of Civil Engineering Information Systems, Japan Society of Civil Engineers. (In Japanese)

Yabuki, N. and Saito, D. (2001b) *An Inspection Information System for Penstocks Using a 3D Product Model and RFIDs*, Journal of Civil Engineering Information Systems, Japan Society of Civil Engineers. (In Japanese)

Yabuki, N. and Furukawa, M. (2002) *An Integrated Design and Project Management System for Prestressed Concrete Hollow Slab Bridges*, Proceedings of the Ninth International Conference on Computing in Civil and Building Engineering, Taipei, Taiwan, pp.1207-1212.