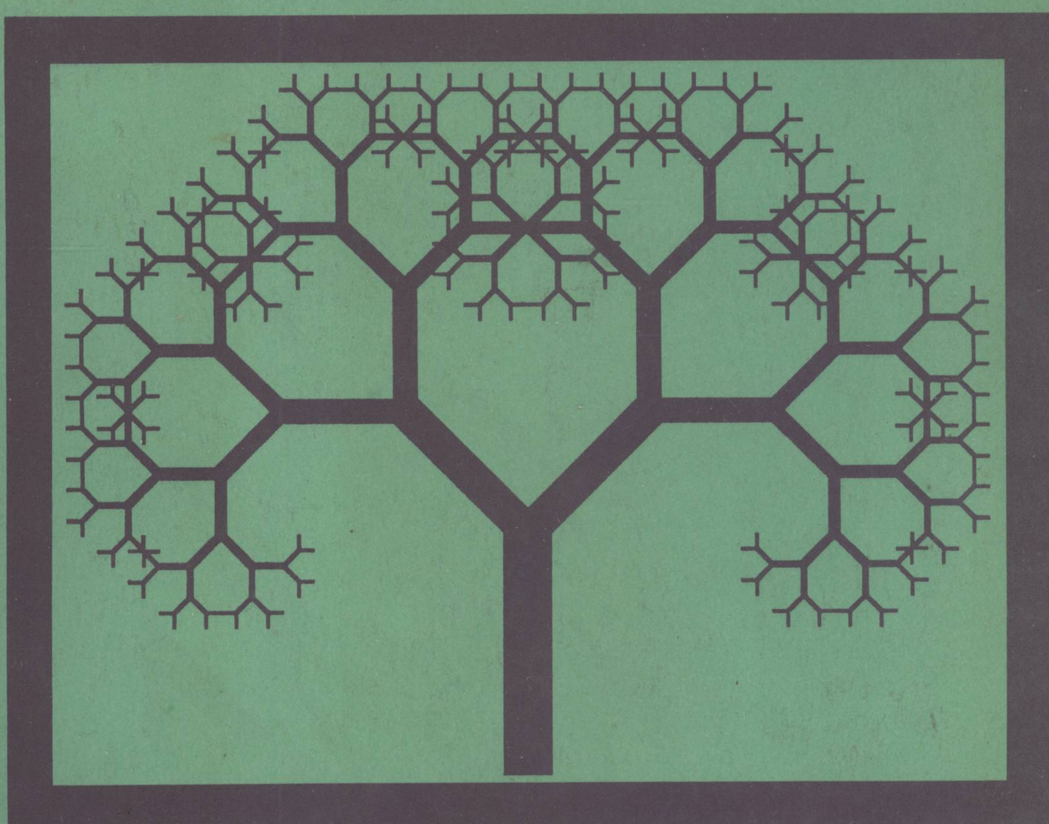
ARTIFICIAL INTELLIGENCE AND STRUCTURAL ENGINEERING



ARTIFICIAL INTELLIGENCE and STRUCTURAL ENGINEERING

Edited by
B. H. V. TOPPING

CIVIL-COMP PRESS

CIVIL-COMP PRESS

10 Saxe-Coburg Place
Edinburgh

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

ISBN 0-948749-15-6

Printed in Scotland
by
MEIGLE PRINTERS
Market Street, Galashiels

ARTIFICIAL INTELLIGENCE AND STRUCTURAL ENGINEERING

PREFACE

The papers included in this volume and its companion "Artificial Intelligence and Civil Engineering" (Civil-Comp Press, ISBN 0-948749-14-8, 1991) were presented at the Second International Conference on the Application of Artificial Intelligence to Civil and Structural Engineering, held at Exeter College, Oxford from 3rd to 5th September 1991.

The proceedings of the first conference in this series were published as "Artificial Intelligence Techniques and Applications for Civil and Structural Engineers" (Civil-Comp Press, ISBN 0-948749-13-X, 1989) and this volume included thirty six papers. Growing interest in the application of artificial intelligence, particularly neural networks and machine learning, to an increasing number of domains is reflected in the large number of papers in the two volumes.

I should like to thank all authors for their contributions and in particular those who travelled to Oxford to present their papers at the Conference.

I should also like to thank the members of the Conference Advisory Board for their help and assistance. The Board Members were: Professor H Adeli, Ohio State University, Columbus, Ohio, United States of America; Dr M Alshawi, University of Salford, United Kingdom; Dr T Arciszewski, Wayne State University, Detroit, Michigan, United States of America; Dr B Balachandran, University of Sydney, Australia; Dr E Balagurusamy, National Centre for Expert Systems, Osmania University Campus, Hyderabad, India; Professor D Blockley, Bristol University, United Kingdom; Dr G Cameron, University of Waterloo, Waterloo, Ontario, Canada; Dr W T Chan, National University of Singapore; Professor J Christian, University of New Brunswick, Fredericton, Canada; Dr I E G Davey-Wilson, Oxford Polytechnic, Headington, Oxford; E William East, US Army Construction Engineering Research Laboratory, Champaign, United States of America; Dr M Eisenberger, Technion, Israel Institute of Technology, Technion City, Haifa, Israel; Professor K Fazio, Center for Building Studies, Concordia University, Montreal, Quebec, Canada; Professor S J Fenves, Carnegie-Mellon University, Pittsburgh, Pennsylvania, United States of America; Dr Renate Fruchter,Stanford University, United States of America; Dr H Furuta, Kyoto University, Japan; Professor James Garrett, Carnegie-Mellon University, Pittsburgh, Pennsylvania, United States of America; Professor J S Gero, Department of Architectural Science, University of Sydney, Australia; Professor D E Grierson, University of Waterloo, Waterloo, Ontario, Canada; Professor P Hajela, Rensselaer Polytechnic Institute, Troy, New York, United States of America; Professor K C Hover, Cornell University, Ithaca, United States of America; A T Humphrey, GEC Marconi Research Centre, Chelmsford, Essex, United Kingdom; Professor C W Ibbs, University of California, Berkeley, United States of America; Dr D G Jamieson, Thames Water, Reading, Berkshire, United Kingdom; Professor P W Jowitt, Heriot-Watt University, Riccarton, Edinburgh, United Kingdom; Dr S Jozwiak, Institute for Fundamental Technological Research, Polish Academy of Sciences, Warsaw, Poland; K Kahkonen, Technical Research Centre of Finland, The Laboratory of Urban Planning and Building Design, Espoo, Finland; Dr Krishnaamoorthy, Indian Institute of Technology, Madras, India; Dr V K Koumousis, National Technical University, Athens, Greece; Dr B Kumar, Strathclyde University, Glasgow, United Kingdom; John Lansdown, Centre for Advanced Studies in Computer Aided Art and Design, Middlesex Polytechnic, Barnet, United Kingdom; Professor K. Lawrence, University of Texas at Arlington, Arlington, United States of America; Professor Liu Xihui, Institute of Computer Science and Knowledge Engineering, China Academy of Electronics and Information Technology, Beijing, PR China; Professor I MacLeod, Strathclyde University, Glasgow, United Kingdom; Dr I May, University of Bradford, Bradford, United Kingdom; Dr J.C. Miles, School of Engineering, University of Wales, Cardiff, United Kingdom; Dr J Oliphant, Heriot-Watt University, Riccarton, Edinburgh, United Kingdom; Dr G Powell, University of California, Berkeley, California, United States of America; Dr D Rehak, Carnegie-Mellon University, Pittsburgh, Pennsylvania, United States of America; Dr S G Ritchie, University of California,

Irvine, United States of America; Dr M A Rosenman, Department of Architectural Science, University of Sydney, Australia; Dr S Rowlinson, University of Hong Kong, Hong Kong; Professor G Roy, The University of Western Australia, Nedlands, Australia; Professor H C Shah, Stanford University, California, United States of America; Dr W J Spencer, Swimburne Institute of Technology, Hawthorn, Victoria, Australia; Professor D Sriram, Massachusetts Institute of Technology, Cambridge, Massachusetts, United States of America; Dr Chee-Kiong Soh, Department of Civil Engineering, Nanyang Technological Institute, Singapore; D Taffs, Ove Arup & Partners, London, United Kingdom; Dr A Watson, Leeds University, Leeds, United Kingdom; and Professor G Yagawa, University of Tokyo, Japan.

Finally I should like to acknowledge the contributions made by my research associates and students during the preparation of this volume. In particular, the contributions made by Bimal Kumar, Mansour Jadid, Asad Khan, Nick Bitoulas and János Sziveri are gratefully acknowledged.

Barry H.V. Topping
Heriot-Watt University
Edinburgh

September 1991

1 STRUCTURAL MODELLING, REPRESENTATION AND ANALYSIS

◦ CONTENTS ◦

FORM, FUNCTION AND BEHAVIOR IN STRUCTURAL ENGINEERING KNOWLEDGE REPRESENTATION

J.A. Abdalla, D.H.D. Phan and H.C. Howard
Center for Integrated Facility Engineering,
Stanford University, Stanford, United States of America

Typical structural engineering objects are complex entities that include large, yet closely related data and knowledge. A great deal of attention should be given to the representation of the elements that make up these complex objects. In order to reduce their inherent complexity, these objects can be decomposed into several simpler primitive objects which are related to themselves and also to their composite object by various relationships. A natural way of decomposing these objects is along the fundamental aspects of *form*, *function* and *behavior* of the design object which also correspond to the sources for data and knowledge of the object. *Form* (or *structure*) describes the physical characteristics of the object, *function* depicts the object role and its intended use, and *behavior* describes the object's manifestation of its functions, when subject to the environment stimuli. The structural engineer reasons about these fundamental aspects during the design process and across various stages of the object lifecycle. Consequently, form, function and behavior should be incorporated into the representation of the design object.

This paper addresses form, function, and behavior representation of structural engineering design objects. It then presents several sample hierarchies of primitive object classes that are characterized according to specific form, function, or behavior criteria. To illustrate the concept, the paper shows a detailed example of a base plate for a transmission pole. This work is part of a current effort to develop the Primitive-Composite data model for structural engineering that will also be briefly presented.

Key Words

Form, Function, Behavior, Knowledge Representation, Primitive Object, Composite Object, Characterization Hierarchy, Relationship, Data Model, Product Model.

INTRODUCTION

The concepts of *form*, *function*, and *behavior* were introduced, with some variations, in various fields of applications. These fields include: qualitative physics [Ref. 2, Ref. 9], function-based representation [Ref. 15, Ref. 16, Ref. 10], engineering design [Ref. 3, Ref. 11, Ref. 17, Ref. 12], design knowledge capture [Ref. 7, Ref. 4], etc. In this research, we are exploring the use of form, function and behavior in Data Modelling and Knowledge Representation. Form, function and behavior are used here as criteria for characterizing primitive object classes into several class hierarchies (or *primitive characterization hierarchies*), each of which relates to a single concept. This will result in the representation of simpler primitive object classes and in more homogeneous primitive characterization hierarchies. Composite object classes can then be defined in terms of these primitive classes from several characterization hierarchies. This will enhance the atomicity, modularity and cleanness desired in the representation of complex structural engineering objects. To accurately represent a structural engineering object, it is essential to represent the ingredient primitive objects and their relationships. The composite object can then be formed by combining these primitive objects using the appropriate relationships. Consequently, much attention should be given to the representation of the elements that make up the composite objects, namely the primitive objects and relationship types.

This paper outlines the major characterization hierarchies of primitive object classes where form, function and behavior are clearly separated. The paper also describes how primitive objects from these hierarchies can be used to define composite objects. First, an overview of the Primitive-Composite data model (which is the main goal of this work) is presented. Form, function and behavior are then introduced along with several primitive characterization hierarchies. To illustrate the concept, an integrated example of a base plate for a transmission pole is presented. Finally, the benefits of this modelling approach are stated.

OVERVIEW OF THE PRIMITIVE-COMPOSITE DATA MODEL

This research is part of an ongoing effort at the Center for Integrated Facility Engineering (CIFE) at Stanford University to develop the *Primitive-Composite (P-C) data model* [Ref. 8]. The long-term objectives of the data model are to help integrating structural engineering activities and to contribute to the integration of facility engineering. The P-C data model is based on the object-oriented data model, and therefore incorporates key object-oriented concepts of *class* (object class), *instance* (or object), *attribute*, and *methods*. In addition, the data model incorporates a number of extensions to the core object-oriented data model. The motivation for these extensions is three-fold: to represent complex structural engineering objects, to offer maximum flexibility in modeling structural engineering data and knowledge, and to facilitate data exchange among different applications.

The following items are extensions to the object-oriented data model in the Primitive-Composite data model. Items 1 to 4 are extensions to describe how objects are organized to represent independent concepts and how these objects can be combined to create complex abstractions that suit the user views. Items 5 to 8 are extensions to support the data exchange using the Primitive-Composite data model.

1. **Primitive Class and Primitive Instance** — A primitive class represents an atomic entity that deals with one simple concept (e.g., shape, material, or function). A primitive class is non-decomposable with respect to the particular domain of modeling. A primitive instance is an instance of a primitive class.

2. **Primitive Class Hierarchy** — A primitive class hierarchy groups primitive classes that represent increasing specializations of a single concept of form, function, or behavior. As an example, two-dimensional shapes can be specialized into 3-sided shapes, 4-sided shapes, 5-sided shapes, while 4-sided shapes can be further specialized into squares, rectangles, trapezoids.

3. **Composite Class** — A composite class consists of a heterogeneous set of related primitive classes that represent a complex concept in the modeling domain or in an application. It is used to describe a common combination of domain concepts that may be found in many structural engineering objects. For example, a "Beams" composite class describes the concept of beam as a combination of specific forms, functions, and behaviors, and provides users with the abstraction of the beam object that they are accustomed to.

4. **Composite Instance** — A composite instance is an instance of a composite class. For example, a "beam23" combines physical form objects that represent its location and shape, a function object that describes its load resisting role, and a behavior object that defines the beam as a flexural analysis element.

5. **Primitive data base schema** — A primitive data base schema is a set of primitive characterization hierarchies that define the basic concepts of a domain. These concepts can be used directly or indirectly by the domain specialists for different applications sharing the same primitive schema. The latter may include some composite classes that are commonly seen by the domain specialist, but not defined specifically for any particular application.

6. **Primitive data base** — A primitive data base contains instances of primitive classes from a primitive schema. As described later, a primitive data base can be used as a medium of data exchange between different application systems.

7. **Composite data base schema** — A composite data base schema includes a subset of the primitive schema, and a set of composite classes that defines an application view of the domain data and suits the particular needs of specialized domain tasks. Composite classes, in this case, provide a convenient framework for formalizing higher level abstractions of complex structural engineering objects, in particular in applications that make use of the primitive schema.

8. **Composite data base** — A composite data base is a data base that contains instances of composite and primitive classes from a composite schema.

FORM, FUNCTION AND BEHAVIOR

Form, function, and behavior are the basic conceptual building blocks for defining characterization hierarchies of primitive object classes. Using the Primitive-Composite approach, we can define structural design objects as composite objects in terms of form, function, and behavior primitives. These primitive objects can be selected from different characterization hierarchies. This section identifies the elements of form, function, and behavior that is needed to represent structural design objects. The detailed development of the P-C data model that incorporates form, function, and behavior representation of structural engineering objects is the subject of our ongoing research [Ref. 1, Ref. 13].

Form

Form describes the physical characteristics of an object. There are many types of form description: spatial, geometric, topological, material, fabrication, etc. This section covers those aspects of form necessary to define structural engineering design objects.

- **Spatial Form:** This form describes where and how a physical design object is located, oriented, and realized in three-dimensional space. Such a description includes the spatial envelope of the object, and its location and orientation with respect to certain global reference data or relative to other objects in its environment. The spatial envelope of a physical object can be defined in terms of a local coordinate system and the dimensions (length, width, and height) of its spatial enclosure. For example, a beam object in a typical structure can be located and oriented in terms of its local coordinate system or relative to its floor or frame objects. Figure 1 below illustrates two characterization hierarchies for coordinate system and spatial enclosure forms.

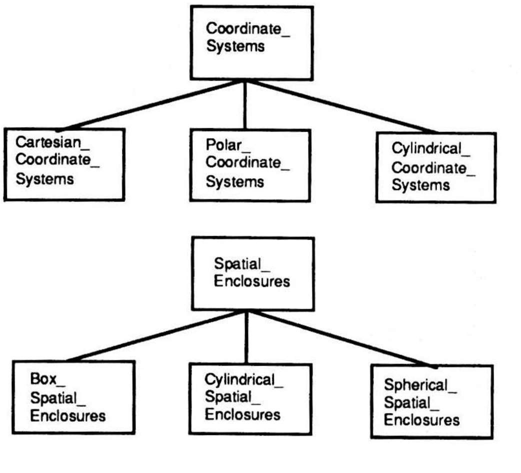


Figure 1: Sample Spatial Form Characterization Hierarchies

- **Geometric Form:** This form defines the geometric shape (and dimensioning) of the object in terms of geometric elements such as points, lines, curves, surfaces, etc. Physical objects are three-dimensional, but their shapes can be represented in many ways by different geometric forms. For example, a rectangular column object can be represented as three 1-Dimensional (1-D) geometric forms that are its width, depth and height. The same column can be represented as one 1-D geometric form (height) and one 2-D geometric form (section-shape), or it can be represented as a solid 3-D geometric form (parallelopiped). Many geometric entities are defined in the PDES/STEP Integrated Product Information Model (IPIM) [Ref. 18]. Figure 2 below illustrates a characterization hierarchy for two-dimensional rectilinear shapes.

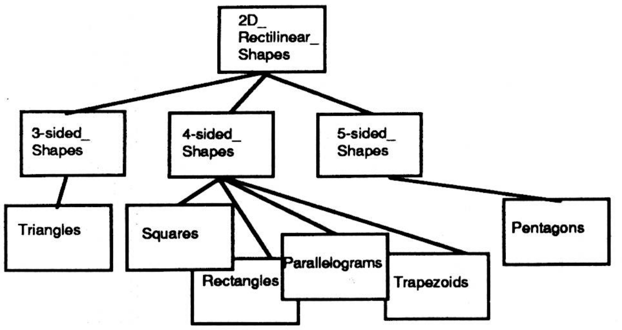


Figure 2: A Sample Geometry Form Characterization Hierarchy

- **Topological Form:** This form defines the connectivity of the object, in the constructed environment, in terms of topological elements. In structural engineering, a wire frame model of the structure is commonly constructed to define the topology of the structure. Such a wire frame model is analogous to a finite element model used for structural analysis purpose. As shown in Figure 3 below, *vertices* (of dimensionality 0), *edges* (of dimensionality 1), *faces* (of dimensionality 2), and *volumes* (of dimensionality 3) are topological primitive entities defined in the PDES/STEP IPIM [Ref. 18] and the GARM model [Ref. 6].

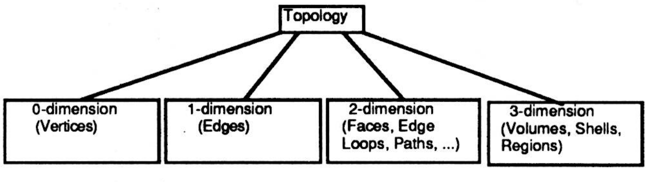
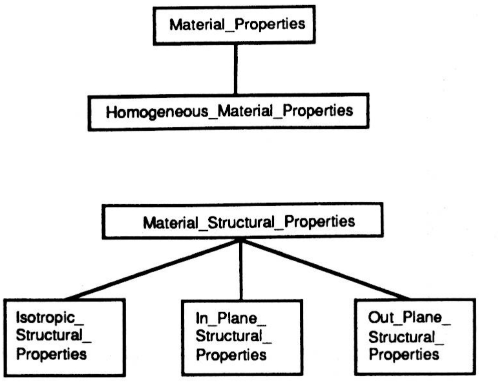


Figure 3: A Sample Topology Characterization Hierarchy

- **Cross-Section Properties Form:** This form includes the cross-sectional properties of the object such as section dimensions, section modulus, area, moments of inertia, torsional rigidity, etc. These properties are derived from the section shape forms of the object. There are several AISC standard shape primitives for structural steel members that are categorized as Rolled Sections, Plates, Bars, etc.

- **Material Form:** This form describes the material type and material properties of the object. The material types used in civil engineering include steel, reinforced concrete, asphalt, mortar, wood, etc. According to the PDES IPMP [Ref. 18], material properties can be classified into groups such as physical, structural, thermal, and thermal expansion. In [Ref. 14] details of material representations for finite element applications have been provided. Figure 4 shows characterization hierarchies for homogeneous material properties and material structural properties. The primitive classes in these hierarchies can be used to represent common material in structural engineering such as steel, aluminum, wood, concrete, etc.



(According to PDES IPMP [Wilson 88])

Figure 4: Sample Material Form Characterization Hierarchies

- **Hierarchical Aggregation Form:** This form is used to describe the hierarchical framework of structural design systems and their component objects. Structural members such as beam, column, girder, etc., are designed not independently of one another, but from some preconceived load resisting systems to which they belong. Common entities of this form include "Systems", "Assemblies", "Arrangements", "Members", "Parts", and "Segments".

- **(Part) Fabrication Form:** This form includes fabrication features of an engineering part prescribed by the designer. There is a large set of standard fabrication features such as taper, bend, cut out hole, edge clipping, edge preparation, NC mark, etc. Standard fabrication features are defined in the PDES Integrated Product Information Model [Ref. 18] and the NIDDESC Ship Structural Model [Ref. 5].

- **Load Form:** This form includes loading description features such load intensity, loading shape, and other factors that describe a load. Figure 5 shows a simple applied load form characterization hierarchy.

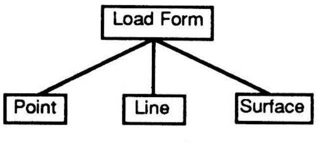


Figure 5: A Sample Load Form Characterization Hierarchy

Function

The function description of an object reflects the intended role or purpose of the object in its constructed environment. An object may serve several functions. For example, structural functions of the building elements relate to some aspects of load carrying, load transferring, part connecting, member supporting roles, etc. These functions come from the specialized relationships among the building elements, or between the building elements and other entities such as loads, load cases, etc. In the Primitive-Composite data model, a number of key structural engineering function objects (primitive) are identified. As illustrated in Figure 6, these primitive objects include: load resisting function, load transmitting function, object supporting function, object connecting function, and object bracing function.

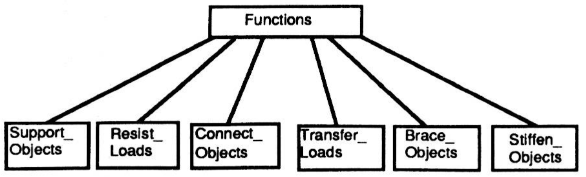


Figure 6: A Sample Function Characterization Hierarchy

EXAMPLE Figure 7 below provides an example of a flat slab, and a column with dropped panel and column capital, and a footing. This example shows the key functions of structural engineering objects such as load resisting function, object supporting function, object connecting function, etc. The *load resisting function* of an object is to resist loads that are applied directly or are transmitted to it from another object. The column object in Figure 7 resists its own weight, the weight of the column-capital, the weight of the dropped-panel, the weight of the flat-slab, the concentrated load directly applied to it, and the distributed loads transmitted to it from the flat slab. The *load transmitting function* of an object is to transfer the loading it carries to other objects in the load path. For example, the flat slab transmits the load it carries to the column through the dropped panel and the column capital. The column object then transmits the load to the footing. The *object supporting function* is to support another object. This function enables the transferring of loads from the supported object, to the supporting object, down to the next object in the load path. The column, for instance, supports the column capital. The column capital supports the dropped panel, which in turn supports the flat slab. The *object connecting function* connects two or more objects together. For example, a column-to-footing connection object connects the column object to the footing object. The *object bracing function* is to brace another object.

In addition, there are several other structural engineering functions that can be included. For instance, the dropped-panel reduces the shear stress in the slab within the area of the dropped-panel to avoid punching shear. The column-capital has a primary function to increase the stiffness of the column object, and thus to reduce the bending moment carried by the flat slab object.

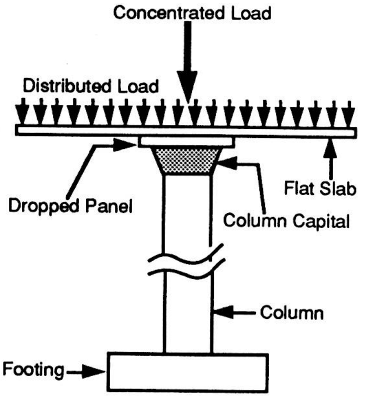


Figure 7: Example of a Column and Flat Slab

Behavior

The behavior of a design object is the way the object responds to the environmental stimuli in carrying out a certain function. Since an object may perform several functions, it follows that it may exhibit different behaviors, each of which corresponds to a particular function or combination of functions. For example, in resisting gravity loads, a column object develops internal axial stresses; in resisting lateral loads, it exhibits shear and bending stresses. In structural engineering, the behavior of a structural component under the influence of loading is manifested in terms of internal forces, stresses, deflection, deformation, vibration, etc. There are also well defined behavior models of structural elements such as flexural bending, torsion, axial buckling, etc.

The design of structural elements includes criteria that impose certain limits on their behavior. These design criteria ensure the acceptable performance of the design object according to professional standards with regard to the following aspects: strength (stresses and internal forces), serviceability (deflection, vibration, cracking, etc.), ductility, stability, and reliability. As shown in Figure 8, these design aspects are used as criteria to characterize the behavior primitive classes.

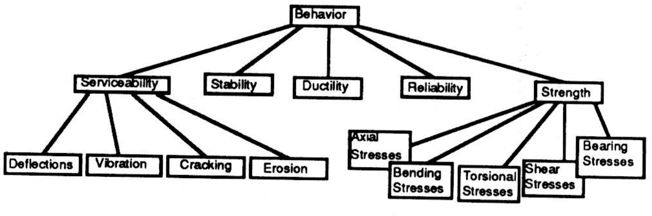


Figure 8: A Sample Behavior Hierarchy

Summary of Form, Function and Behavior

The major aspects of form, function, and behavior are summarized in Figure 9 below. They represent three orthogonal planes. A composite object class can be located within the region defined by these three planes, where it makes use of certain primitive classes from each of these planes. Although not shown in the figure, there are some underlying mappings and relationships among primitive objects of the different planes.

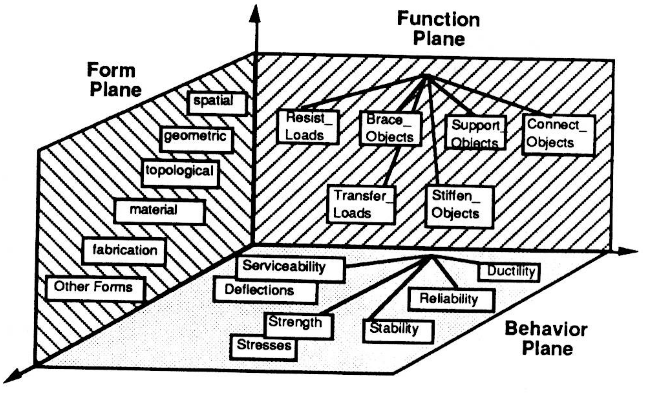


Figure 9: Form, Function, and Behavior in Structural Engineering

AN INTEGRATED EXAMPLE: TRANSMISSION POLE BASE PLATE

In this section, we show an integrated example that demonstrates the use of form, function, and behavior primitive objects in defining a composite object of a base plate for a transmission pole. The example also shows the effects of such a representation on different tasks in the design process.

Typical transmission steel poles consist of steel shafts of various lengths (up to forty feet). These shafts are tapered, hollow, and cylindrical; their cross section can be hexagonal, octagonal, or twelve-sided. The size at the pole base shaft is determined from structural analysis/design of the pole structure for various loading conditions. The reaction loads at the base of the pole are used to design the base plate and anchor bolts assembly (as shown in Figure 10) whose functions are to carry and to transmit these loads to the foundation. Each load case of the pole structure yields a combined axial load P, a resultant shear force V, and an overturning moment M as the applied loads on the base plate.

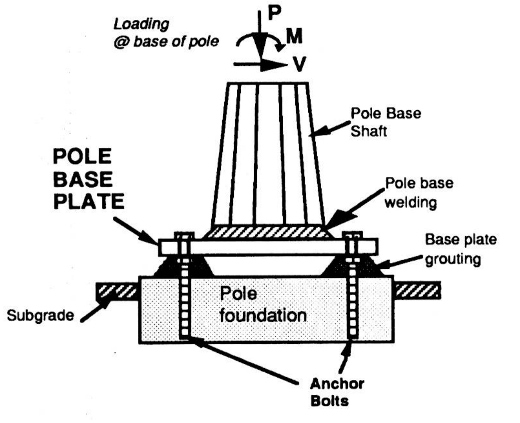


Figure 10: Transmission Pole Base Plate and Anchor Bolts Assembly

A sample base plate with a 4-bolt-hole pattern is illustrated in Figure 11 below. In general, the pole base plate is made of normal strength (grade A-36) or high strength (grade A-572 or other) steel material.

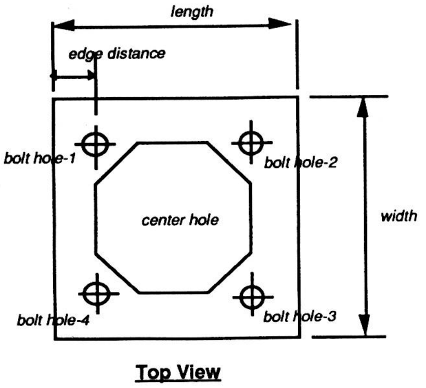


Figure 11: A Sample Base Plate with 4-Bolt-Hole pattern

FORM REPRESENTATION OF THE BASE PLATE COMPOSITE OBJECT The physical description of the base plate object (or instance) used in this example are represented in terms of its form primitive objects. The latter are instances of object classes from disjoint characterization hierarchies that correspond to the form characterization criteria as presented earlier in Section 3.1. Figures 12, 13, 14, 15, 16, and 17 below illustrate the base plate composite object and its spatial, geometric, topological, cross section properties, material, and fabrication form primitives respectively. These figures also show the corresponding primitive object classes from which these primitives are instantiated. Some of these classes are based on entities defined in PDES/STEP IPMP [Ref. 18] and in NIDDESC [Ref. 5].

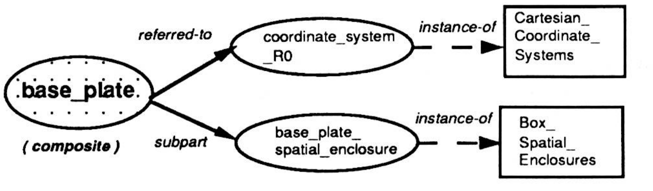


Figure 12: The Base Plate Composite Object and its Spatial Form Primitives

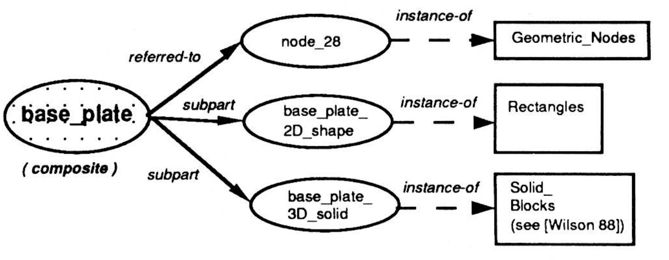


Figure 13: The Base Plate Composite Object and its Geometric Form Primitives

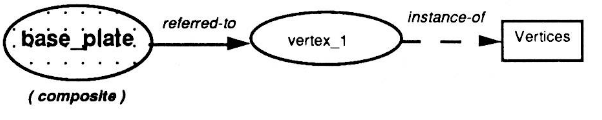


Figure 14: The Base Plate Composite Object and its Topological Form Primitives

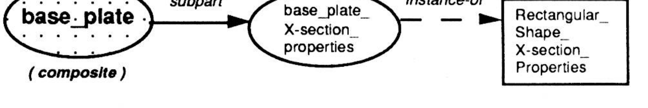


Figure 15: The Base Plate Composite Object and its Cross Section Properties Form Primitives

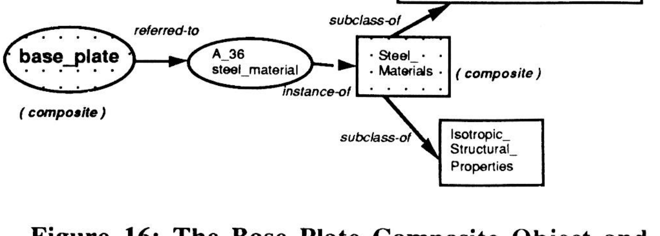


Figure 16: The Base Plate Composite Object and its Material Properties Form Primitives

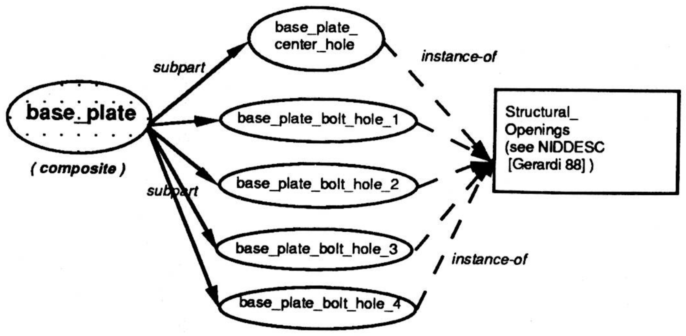


Figure 17: The Base Plate Composite Object and its Fabrication Form Primitives

FUNCTION AND BEHAVIOR REPRESENTATION OF THE BASE PLATE COMPOSITE OBJECT The first function of the base plate is to transfer the reaction loads P, V, and M at the base of the pole to the anchor bolts (which in turn transfer them to the foundation). These external loads transform into axial loads (Tb on the tension side and Cb on the compression side), and shear loads (Vb) of the anchor bolts. The combined loading effect on the set of anchor bolts, however, applies the axial bolt loads (i.e., Tb and Cb) on the base plate whose second function is to resist these loads. Figure 18 illustrates the analysis model used to analyze and design the base plate. In this model, the base plate is subject to both shear and bending stresses.

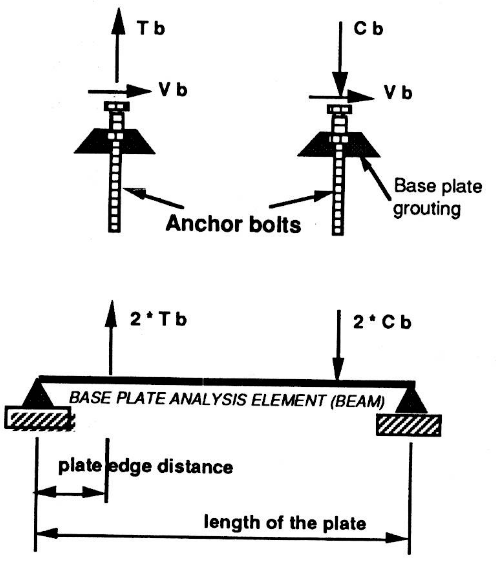


Figure 18: Base Plate Analysis Model

Figures 19 and 20 shows the base plate composite object and its function and behavior primitive objects respectively. Again, these primitive objects are instances of object classes from different characterization hierarchies that correspond to specific function or behavior characterization criteria as presented earlier in Sections 3.2 and 3.3. These figures also show the corresponding primitive object classes from which these primitive objects are instantiated.

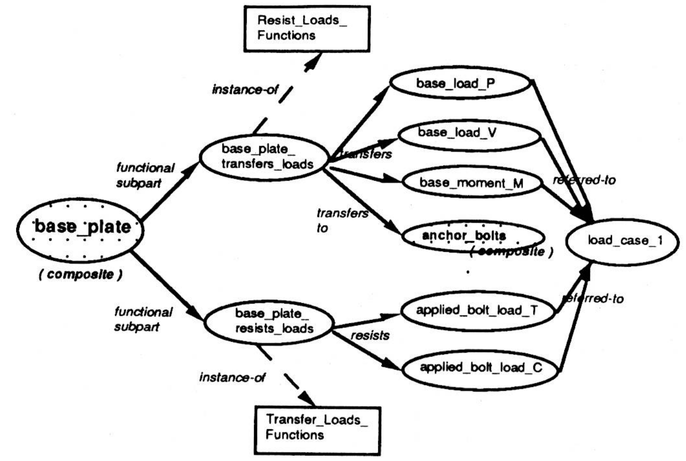


Figure 19: The Base Plate Composite Object and its Function Primitives

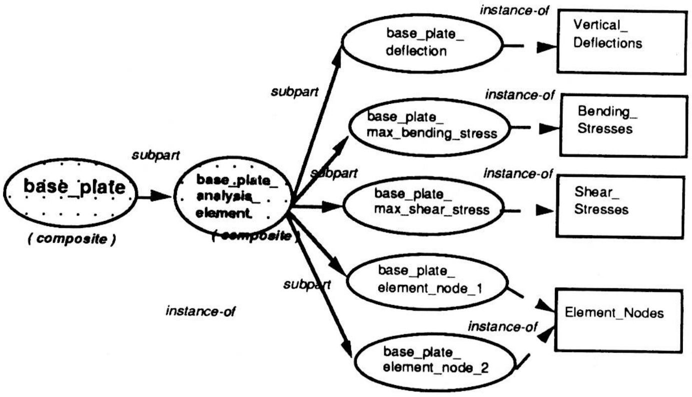


Figure 20: Base Plate Instance and its Behavior Form Primitives

THE BASE PLATE COMPOSITE OBJECT As shown previously, the base plate composite object is represented in terms of a number of form, function, and behavior primitive objects. The disjoint primitive characterization hierarchies provide the primitive object classes from which these primitive objects are instantiated. Each of these hierarchies corresponds to a specific characterization criterion, and therefore describes a single concept of form, function, and behavior. As illustrated in Figure 21, the complete base plate composite object definition is done by selecting the proper primitive classes from various characterization hierarchies (all of which are not shown here due to the limited scope of the paper), and thereby creating the appropriate primitive instances.