NIST-GCR-97-738

EVALUATION OF THE AP230 APPLICATION PROTOCOL

Prepared for:
U.S. DEPARTMENT OF COMMERCE
National Institute of Standards and Technology
Building and Fire Research Laboratory
Computer Integration Construction Group
Building Environment Division
Gaithersburg, MD 20899

NIST

U.S. DEPARTMENT OF COMMERCE
William M. Daley, Secretary

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
Raymond Kammer, Director

NIST-GCR-97-738

EVALUATION OF THE AP230 APPLICATION PROTOCOL

James H. Garrett, Jr.
Steven J. Fenves
Han Kiliccote

Carnegie Mellon University
Department of Civil and
Environmental Engineering

Prepared for:
U.S. DEPARTMENT OF COMMERCE
National Institute of Standards and Technology
Building and Fire Research Laboratory
Computer Integration Construction Group
Building Environment Division
Gaithersburg, MD 20899

September 1997

DEPARTMENT OF COMMERCE
William M. Daley, Secretary

NATIONAL INSTITUTE OF STANDARDS AND
TECHNOLOGY
Raymond Kammer, Director

NOTICE

This report was prepared for the Building and Fire Research Laboratory, Computer Integrated Construction Group of the National Institute of Standards and Technology under grant number 60NANB6D0143. The statement and conclusions contained in this report are those of the authors and do not necessarily reflect the views of the National Institute of Standards and Technology, the Building and Fire Research Laboratory, or the Computer Integrated Construction Group.

TABLE OF CONTENTS

LIST OF FIGURES

1. Introduction

AP230 is a proposed international standard for the representation and exchange of electronic data relating to structural steel framed buildings. It is still currently under development and not yet accepted as an international standard.

Our major objectives in doing this research are to evaluate the AP230 Application Protocol with respect to US practice and extend it for use in early conceptual stages of the facility design process.

In this report, we present the problems we identified in AP230 and propose extensions to solve these problems.

This report is organized as follows. In the next section, an overview of AP230 is presented. The third section presents the problems we have identified in AP230. The forth section presents an alternate model to solve the problems. The fifth section presents how this new model can be integrated with AP230. The last section is the summary of this research.

2. Overview of AP230[1]

The aim of AP230 is to provide an international standard for the representation and exchange of electronic data relating to structural steel framed buildings.

Currently in the steel construction industry, data exchange occurs in an ad hoc manner using translators between software products to exchange and share data. However, to provide data exchange between *n* systems, this approach may (potentially) require $n(n-1)$translators.

The aim of AP230 is to reduce the number of translators needed to exchange and share data between different software products in the steel construction industry by creating and standardizing a data format which is internationally agreed upon, open, and vendor–neutral. This standard data format will enable each user to exchange data across the steel industry; each user will require a single translator in order to achieve this exchange. Moreover, creation of a data format that is product–based, as opposed to being based simply on geometrical data, will enable members of the steel construction industry to exchange a full range of information related to their activities.

2.1. AP230 as an International Standard

The International Organization for Standardization (ISO) is a world–wide federation of national standards bodies (ISO member bodies). The work of preparing international standards is carried out through ISO technical committees. ISO 10303 is an international standard for the computer–interpretable representation and exchange of product data and is organized as a series of parts, each published separately. The parts of ISO 10303 fall into one of the following series: description methods, integrated resources, application integrated constructs, abstract test suites, implementation methods, conformance testing, and application protocols. AP230 (ISO 10303 part 230) is an application

[1] The information presented in this section is derived from draft AP230 documents. These documents can be currently accessed from http://www.leeds.ac.uk/civil/research/cae/step/ap230/ap230.htm

protocol for the representation and exchange of information relating to structural steel framed buildings. The computer applications to which it relates are those providing structural analysis, member design, connection design, and detailing functions for the designers and constructors of steel structural systems in buildings.

2.2. CIMsteel Integration Standards

The product model underlying AP230 is based closely on the product model underlying the CIMsteel Integration Standards (CIS). The CIMsteel project was aimed at improving the efficiency and effectiveness of the European constructional steelwork industry through the introduction of computer integrated manufacturing (CIM). The CIMsteel Integration Standards were formally launched in 1995 and quickly adopted by software vendors.

The CIS are a set of specifications that provide standards which the developers of engineering software applications can develop and implement translators by which users of such applications are able to export engineering data from one application and import it into another.

The CIS has been designed to be applicable to any application that involves steel structures. The information related to an application is exported from an application in the form of a data exchange file. Within this file, the engineering information is structured in a logical and standardized way. Other applications that are based on CIS standards can extract the information that is relevant to the application.

2.3. Scope of AP230

In AP230, the life–cycle of a building is decomposed into five separate stages: (1) plan, (2) design, (3) construct, (4) use, and (5) demolish. AP230 supports the exchange of data during the first three of these stages. To describe activities in these three stages, AP230 uses application activity models (AAM's), presented in a graphical format, to represent the processes and information flows that relate to these three stages of a building life cycle.

Among various AAM's that constitute the first three stages of a building life cycle – plan, design, and construct AP230 currently concentrates on structural design, loading assessment, structural scheme modelling, structural analysis, member design, connection design, and steelwork detailing. In AP230, these activities are grouped into two consecutive activities: *produce overall design* and *produce detail design.* In AP230, the information flow between these activities and sub–activities (i.e., smaller activities that constitute a larger activity) is shown in a graphical format. The sub–activities that occur in these activities and the information exchange that occurs are shown in Figures 1 and 2.

As shown in Figure 1, the activity "produce overall design" consists of five activities: manage overall design; produce concept design; produce overall architectural design; produce overall service design; and produce overall structural design. The activity "produce overall design" has two major outputs with which AP230 is concerned: overall structural design and analysis results. In ISO terminology, the outputs of an activity, which may become input to another activity, are called Units of Functionality (UoF).

Similarly, the activities that are performed to produce a detail design are shown in Figure 2. These activities are: manage detail design; design members; design connections; and detail steelwork.

Figures 1 and 2 also show the Units of Functionality (data flows in the activity model) with which AP230 is concerned. These UoF's are:

- Project Brief;
- Analysis Results;
- Overall Structural Design;
- Structural Scheme;
- Design Loading;
- Load Case Combinations;
- Basic Load Cases;
- Structural Models;
- Member Design Information;
- Connection Design Information; and
- Detailed Designs.

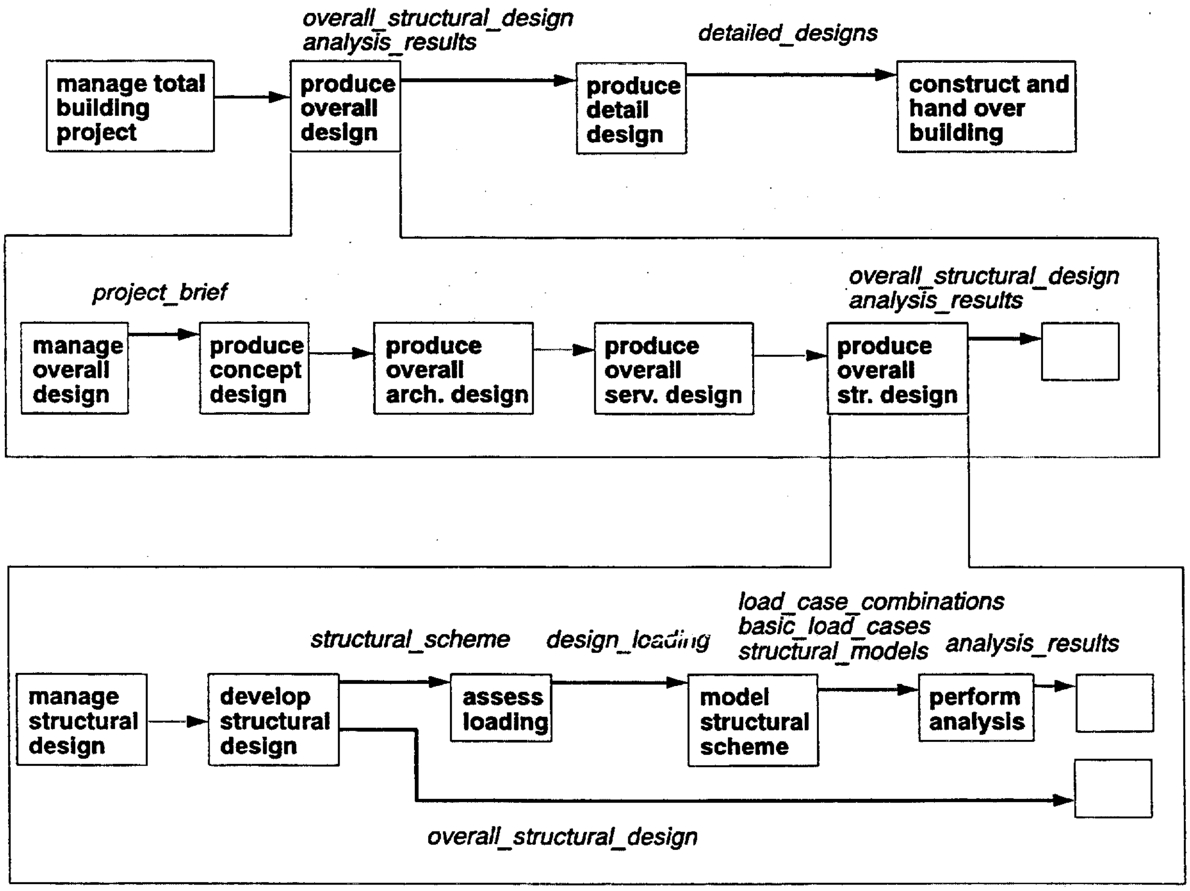

Figure 1. Activities of which produce overall design consist.

Figure 3 shows the simplified graphical representation of one of these UoF's: overall structural design.

The products addressed by AP230 – steelwork building frames and their components – are employed in low, medium and high rise construction in domestic, commercial and industrial applications. The AP is applicable to a variety of structures ranging from simple, single–story portal frame industrial "sheds" to multi–story office blocks. The main structural steelwork is covered by this AP, as is secondary steelwork such as purlins, siderails, cleats and cladding. The frames considered may be braced or unbraced. Connections can be pinned, rigid, or semi–rigid with rigid and semi–rigid being full or partial strength.

The data model underlying this AP treats structural steel frames as being fabricated from manufacturing assemblies, and, in turn, views manufacturing assemblies as being made up of parts and joint systems. The AP includes support for rolled, welded, cast, or cold–formed parts (although only limited information is held on cast and cold–formed parts).

AP230 also covers welded and bolted joint systems, and bolted joint systems may involve ordinary and pre–loaded bolts.

In general terms, the data supported by AP230 include: geometric and geographic data; data related to physical and material characteristics; data related to structural behavior, data related to unique identification; logical grouping data; and temporal data. AP230 specifies persons and organizations, projects, sites, analytical models, analysis results, structures (locations, connectivity, and characteristics of parts, joints and sub–structures), and design assemblies (connectivity and descriptions of members, connections, and frames).

The following are outside the scope of AP230: complex parts; complex features; complex joint systems; curved prismatic parts; non–isotropic materials; non–standard fabricated beams; crane rails; compound beams; second order elastic analysis; hold–down bolts, studs (used in joint systems); threaded rods; pins; cambered

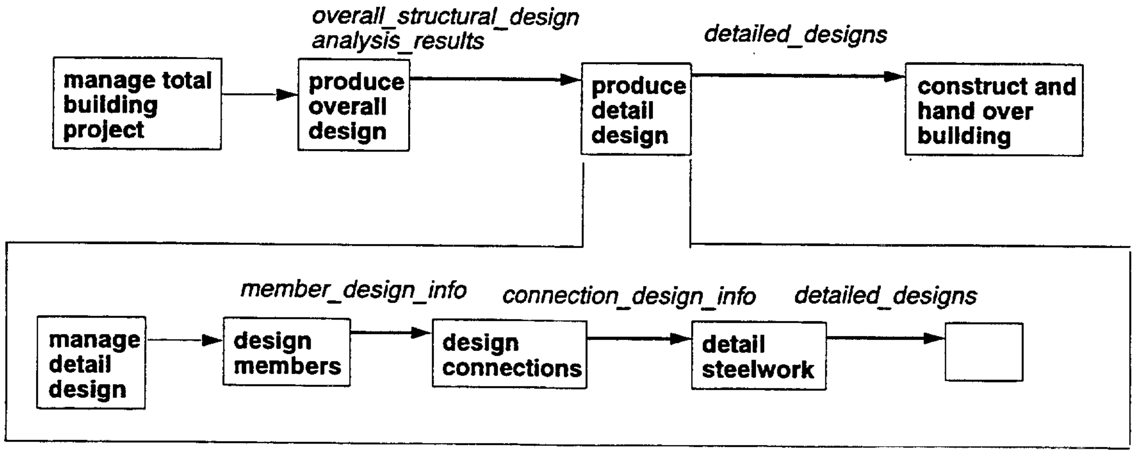

Figure 2. Activities of which produce detail design consist

beams; dimensional tolerances; bearing surfaces; elastic bearings; expansion joints; dynamic / cyclic loading; moving loads; 3D solid modelling and detailed FEA; cost issues; organizational issues; contractual arrangements; and composite construction.

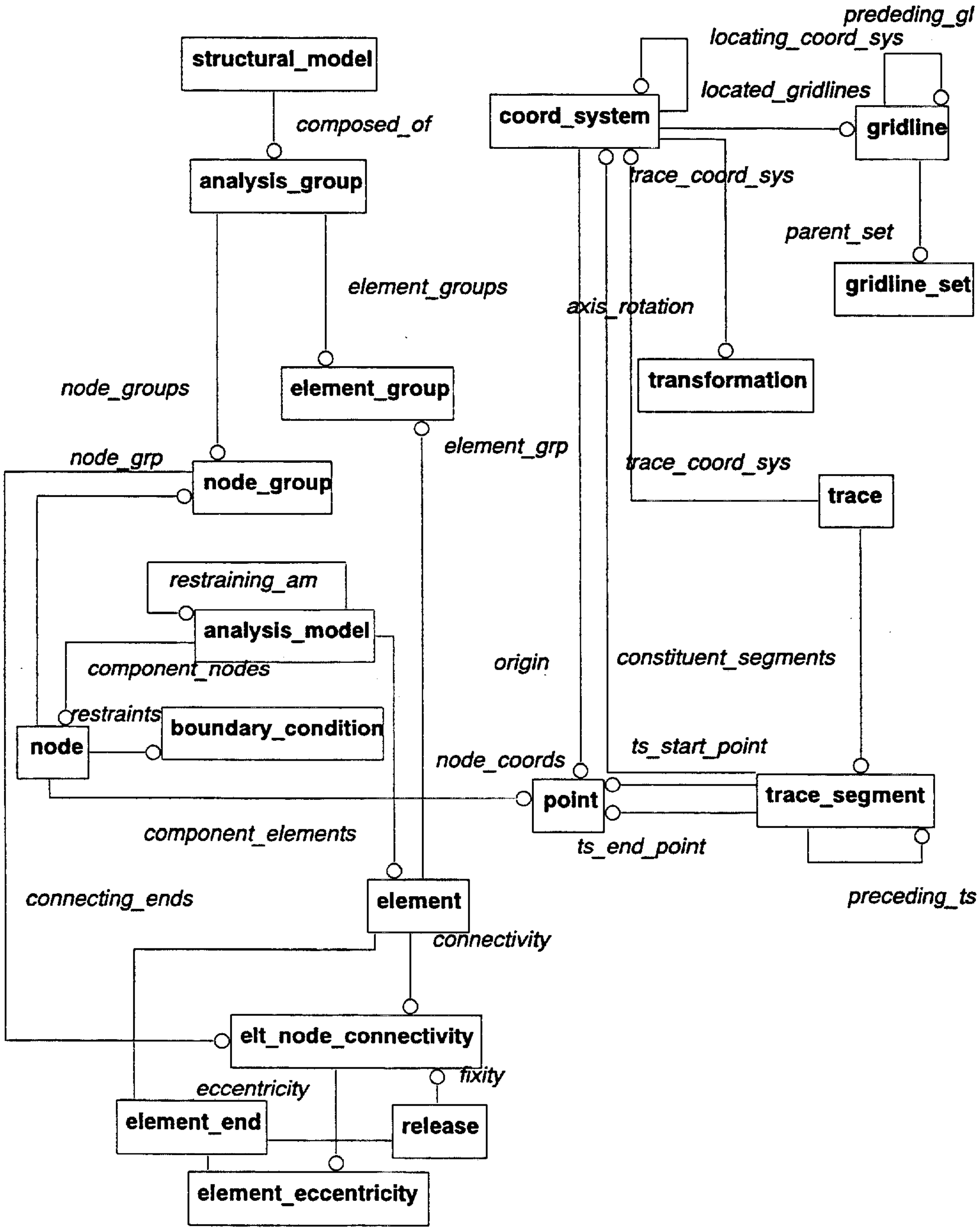

Figure 3. Overall structural design Unit of Functionality

3. Evaluation of AP230

To be able to evaluate AP230 from US a perspective we decided to:

- perform an informal analysis of AP230;
- develop an analysis module using AP230 as the data model;
- interface AP230 with the SEED project at CMU; and
- represent portions of the LRFD Specification using AP230 as a data model.

In the next sections, the detailed description of these tasks is presented.

3.1. Informal analysis

The first task we performed was to become extremely familiar with AP230. To be able to do this, we downloaded the current version (N528) of AP230 from the AP230 Web site (http://www.leeds.ac.uk/civil/research/cae/step/ap230/ap230.htm). From the same URL, we also obtained documents that contain comments of various authors on the current version AP230. We also obtained from the CIMsteel web site information related to the CIMSteel project, which provided the basis for the development of AP230.

To be able to discuss and clarify the issues we discovered during informal analysis, we also participated in two meetings regarding AP230. The first one of these meetings was held in Chester, UK and the second one was held in Chicago, IL.

In Chester, an international ISO 10303 STEP meeting was held between March 3–7, 1997. This meeting was hosted by the British Standards Institution. As part of this meeting, Work Group 3, Task Group 12, which deals with AEC and Building Construction, met. In the Task Group meetings, we met the authors of AP230, Dr. Alastair S. Watson, Dr. Michael A. Ward, and Dr. Andrew J. Crowley. We had several meetings with Dr. Ward and Dr. Crowley about AP230. During these meetings, we were able to present our questions, concerns, and recommendations. During these meetings, the authors of AP230 claimed that:

- with the completion of Clause 4 of the AP230 document and completion of the ARM validation clause of the AP230 Validation report, Group One documentation for AP230 had been completed;
- the entire set of definitions accompanying the CIMsteel Integration Standard (CIS) data–model will be rewritten in line with STEP guidelines using AP230 as the base model;
- even though the AP230 project of CIMsteel has severe financial problems, the authors still had funding until the end of 1997 and they believed that the ARM stage of AP230 could be finished by that date; and
- the resources have not been found to fund Interpretation and Qualification for AP230 and that until this problem was resolved, AP230 could not progress to CD ballot.

The second meeting in which we participated was held in Chicago, IL on May 7, 1997 as part of 1997 National Steel Construction Conference (NSCC 1997). During this meeting, we had the opportunity to see the live CIMsteel demonstration on which AP230 is based. This demonstration featured four software applications exchanging information according to the most recent CIS specification on which the AP230 data model is based. The software packages involved were Frameworks from Intergraph, Space from QSE, StruCAD from Ace-Cad, and SSC from Godata. All four vendors participated in the downstream exchange of data between steel layout, design, detailing and cost estimating activities. Feedback was demonstrated, involving a late design change being fed upstream. The presentations before the demonstrations provided background information and further detail. After the demonstration, the members of the AISC Task Committee met. During that meeting, we presented our preliminary findings of the analysis of AP 230 we did as part of this project. After a long discussion following our presentation, the Task Committee, to be able to recommend an appropriate strategy for data exchange to AISC, formally requested complete AP230 and CIS documentation from its authors. The full documentation on AP230 and CIMSteel had not been available prior to that point in time. Subsequent to this meeting, we were given access to the CIMsteel specification and AP230 details.

While performing an informal analysis of the current version of AP230 (N528), we discovered that the authors of AP230 had developed a very large and extensive product and process model for the exchange of computer–interpretable information relating to structural steel building frames. Although we were impressed with the amount of the work they had undertaken, we found the following problems with the current version of AP230.

- The documentation of AP230 is not well organized, contains editorial errors, is incomplete and difficult to browse.
- The naming convention used in the current version of AP230 is internally inconsistent in many places. Abbreviations of the same term may appear in many different forms in different terms, or the same term may not be abbreviated at all in other places. We also found that the abbreviation of some terms are misleading.
- The application reference model (i.e., the product data model expressed in EXPRESS–G) mainly supports overall structural design and contains little information on other activities including conceptual design.
- The application activity model (i.e., the process model expressed in IDEF0) is much more general than the activities covered in the application reference model. It is not clear whether the authors plan to cover some of these activities in the future or whether they consider all of these activities out of the scope of AP230.
- We found that the activities that are covered by the application reference model are arbitrarily selected and do not form a single consecutive subset of the activities described in the application activity model.
- We were unable to find a logical explanation of why some of the outputs of the activities in the application activity model appear as controls to the consecutive activities, but not as inputs.
- The Units of Functionality (i.e., the logical link between the application activity model and application reference model) are not

very informative and do not convey the content of the information that will be carried in these units. The major problem is that it is not possible to deduce if an application object used in a Unit of Functionality is an object created by the activity that produces that unit of functionality or used as input by that activity.

- The application objects listed under the *Units of Functionality* and *Conformance Requirements* sections are, in most cases, incomplete and inconsistent with the description of the functionalities and conformance classes.
- The application reference model is only available in EXPRESS–G and not in (ASCII) EXPRESS.

We also identified the following technical problems with the current version of the application reference model.

- The application reference model contains classes that can be used to describe three logical views of the same structure: *design_assembly, manufact_asbly* and *analysis_group*. However, the explicit links between the classes in these views do not exist.
- The immediate subconcepts of the top level concept, *steel_structural_frame_entity,* are arbitrarily chosen and the selection is very inconsistent and misleading. For example, even though *weld_mechanism* is a subconcept of *steel_structural_frame_entity, bolt_mechanism* is not.

3.2. Development of an analysis module

In the early phase of our analysis, one of the tasks we were planning to perform was to develop an analysis module that uses the data model used in AP230. We were planning to use this module to test various aspects of AP230 pertaining to structural analysis, such as the adequacy of support for the representation of structures, loadings, load case combinations, basic load cases, and analysis results.

However, during the two meetings regarding AP230 in which we participated (Chester, UK and Chicago, IL), various software vendors presented various analysis programs that were integrated with CIMsteel international standards. Based on these presentations and our informal analysis, we concluded that since AP230 is based on CIMsteel, AP230 provides adequate support for performing structural analysis and any additional testing of support for analysis would be redundant

3.3. Interfacing of AP230 with SEED

The SEED project intends to develop a software environment that supports the early phases of building design (Flemming et al., 1993). The goal of the project is to provide support for the various aspects of preliminary design of buildings that can gain from computer support, including the rapid generation of design representations.

SEED consist of three major modules: SEED–Pro, which provides support for architectural programming, SEED–Layout, which provides support for schematic layout design and SEED–

Config, which provides support for schematic configuration and design.

Figure 4 shows the various modules of SEED and how these modules interact with each other. As shown in the figure, the output of the modules in SEED (i.e., architectural program, 2D Layout design, and 3D Building Design) are stored in a database. The data model used in the database is specific to SEED.

We identified two major approaches that can be used to achieve the integration of AP230 with SEED.

- extend or change the data model used in the database to support the concepts and relations provided in AP230; or
- develop a translator that can translate the data model used in the database to the data model of AP230.

Since we do not have control over the modules of SEED, we concluded that the easiest way to integrate AP230 with SEED was to develop a translator between the data models used in SEED and AP230.

To simplify the task of developing a translator, we concentrated on SEED–Config, in which the main task addressed is to generate and evaluate a schematic three–dimensional configuration of spaces and physical building components.

While working on the translator, we realized that the data model used by SEED–Config is substantially different from the data model of AP230. The major differences that we identified are as follows:

- The class hierarchy used in SEED–Config is based on the architectural view of a structure and not a civil engineering view. Due to this difference, a structure represented in SEED–Config may be partitioned into components

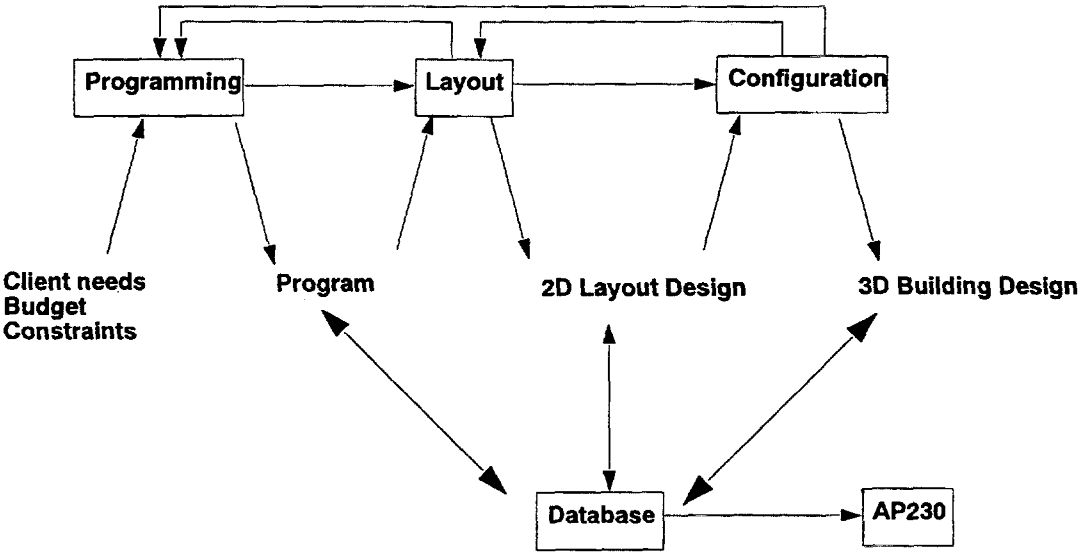

Figure 4. Integration of AP230 with SEED

that do not exist in AP230, such as columns, wall elements, slab elements, etc.

- SEED–Config provides a conceptual view of the structure. A beam running through multiple frames and physically cut into or built–up from multiple pieces may still be represented as a single entity. In AP230, such a beam cannot be represented as an entity but must be represented using the pieces of which the beam consists.
- Since SEED–Config is used during the process of design, it incorporates the assumption that some design information may be missing (e.g., in SEED–Config it is possible to represent a beam for which the cross section is not known). AP230 requires a much more complete view of the structure.

Because of these differences and the problems we encountered during the representation of portions of LRFD (see Section 3.4.), we decided that before we start working on the details of a translator, we should extend AP230 to eliminate these differences. The data models that eliminate some of these differences are presented in Sections 4. and 5.

3.4. Representation of portions of LRFD

The major task we performed while analyzing AP230 from a US perspective was to represent portions of AISC LRFD (AISC 1993) specification using AP230 as the data model. This task allowed us to check whether AP230 can be used as a product model to express portions of LRFD, thus to check whether the data model used in AP230 supports to US practice. To be able to do this, we focused on three parts of LRFD.

- Section A4, which deals with loads and load combinations;
- Sections B1, B2, and B3, which deal with gross, net and effective net area of members; and
- Section D1, which deals with design tensile strength of tension members.

While expressing these portions, we found that:

- LRFD and AP230 use different terminology to express the same entities and attributes (e.g., the term *member* in LRFD corresponds to the term *element* in AP230 and the term *element* in LRFD has no equivalent in AP230, because AP230 does not support built–up members);
- the Units of Functionality defined in AP230 assume an order for the design process (e.g., AP230 assumes that member design is done before connection design, so it is not possible to establish or express a net area or effective net area during the member selection activity using AP230);
- AP230 does not support built–up members, composite design and earthquake loads;
- even though most of the information required for evaluating a design against LRFD can be expressed using AP230, deducing some of the information is a very difficult task (e.g., calculating whether a tension load is transmitted directly to each of the cross–sectional elements requires reasoning with the geometry);
- because AP230 can only be used to exchange information regarding the geometry of members, but not their functionality (e.g., whether a plate is used as a stiffener cannot

be expressed using AP230; only its geometry can be expressed using AP230), it is impossible to check if a member satisfies some provisions of LRFD; and

- AP230 does not support design evolution. In AP230, different entities (design assembly, analysis assembly and manufacturing assembly) are used to represent different views of the same structure at different phases of the design. In each assembly, different entities are used to represent the structure (e.g., *design_part* is used in a design assembly and *located_part* is used in a manufacturing assembly). Thus, AP230 requires the creation of new instances to represent the same entity at different phases of design. This makes conformance checking very difficult because AP230 does not always provide explicit links between different views of the same entity.

Based on these difficulties, we decided to propose modifications to AP230 that would support support conformance checking using LRFD. This new model is presented in Sections 4. and 5.

4. An Alternate Model for Steel Structures

As stated in Section 3.4., AP230 does not support design evolution. The entities that are used to represent different views of the same structure at different phases of the design (design assembly, analysis assembly and manufacturing assembly) are independent of each other, i.e., in each assembly, different entities are used to represent the structure. For example, while design assemblies require the usage of an entity called *connector* to describe a connection, manufacturing assemblies require the usage of an entity called *located_joint_system* for the same connection. Thus, to represent a connection in a manufacturing assembly, which is already represented in design assembly, a new instance (an instance of *located_joint_system*) must be created. In fact, AP230 requires the creation of different instances to represent the same entity at every phase of design. In AP230, there are two major sets of entities that are duplicated:

- entities that represent members; and
- entities that represent connections.

Entities that represent members. AP230 uses five distinct entities to represent members:

- part_joint;
- struc_member;
- element;
- design_part (and s_part); and
- located_part.

The information stored in these entities is often duplicated. For example, the length of a member is available (or can be derived from the end coordinates of the member) in most of these entities. Also, AP230 preselects which information should be made available in each entity. For example, information related to the material of the member is only available in the entities *element* and *s_part.*

Entities that represent connections. AP230 uses five distinct entities to represent information related to connections:

- struc_connection;
- connector;
- assembly connection,

- located_joint_system; and
- node.

Similar to entities that describe members, some of the attributes stored in these entities are usually duplicates of each other.

To eliminate most of the duplicate entities and provide design evolution, we first developed an alternative data model to the one used in AP230. Later, as described in Section 5., we integrated this model into the data model used in AP230.

4.1. Requirements of the new model

In this section, the major requirements we considered while developing the alternate model are described. The new model should provide support for:

- evolutionary design information;
- multiple views of the same entities;
- a conceptual view of the structure;
- missing information;
- derivable information;
- composition;
- functional decomposition;
- strong connectivity between three views (analysis, design, and manufacturing) of the structure;
- decomposition of connections;
- decomposition of members;
- built–up members; and
- virtual segments.

These requirements are described in the next twelve sections.

4.1.1. Evolutionary design information.

As described in Section 4., AP230 requires the creation of new instances to represent the same entity at different phases of design. We have a different perspective for design activities. We perceive a design activity as one that adds more details to the same entity, not as one that creats new entities with additional information.

4.1.2. Multiple views of the same entities

In steel structures, an entity such as a member, can have multiple views. For example, a built–up member can be perceived as a single entity or can be broken into several elements for analysis purposes. These views do not affect the physical reality that exists. They are merely reclassifications of some entities for different purposes. For example, consider the structure shown in Figure 5 .

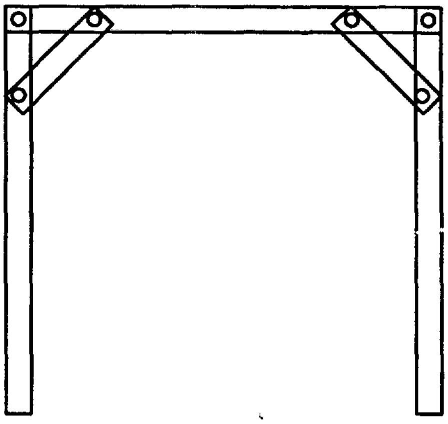

Figure 5. A simple structure

This structure can be described in at least three different ways. The first representation (view) is shown in Figure 6, which has five members and six connections. .

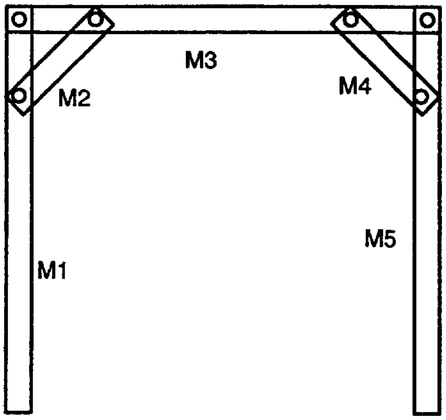

Figure 6. First view of the simple structure

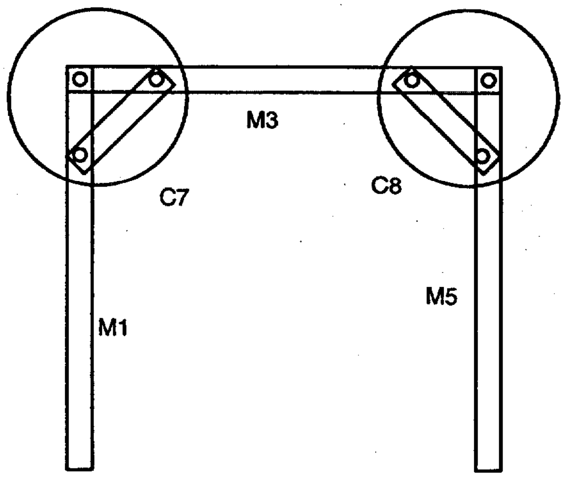

Figure 7. Second view of the simple structure

As shown in Figure 7, a second view has three members and two connections. The connections have secondary members and three sub-connections.

As shown in Figure 8, a third view has three members, two design assemblies and four connections. Each design assembly has three members and three connections. Notice that in this

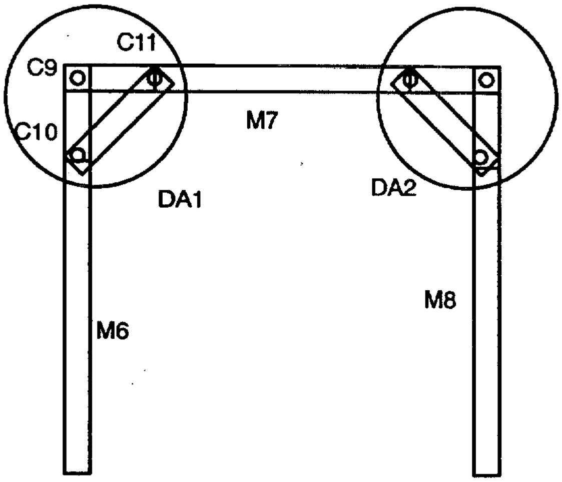

Figure 8. Third view of the simple structure

view, members have been assumed to be physically divided into two members.

4.1.3. Conceptual view of the structure

It may be desirable to represent a beam running through multiple frames and physically divided into multiple pieces as a single entity. In AP230, such a beam cannot be represented as a single entity, but must be represented using the pieces of the beam.

4.1.4. Missing information

During the process of design, even though some design information may be missing (e.g., as discussed in 3.3., in SEED–Config it is possible to represent a beam for which the cross section is not known), it may desirable to exchange the incomplete representation of the structure with another application.

4.1.5. Derivable information

Even though some information may be derivable from other pieces of information when the design is complete, in the early phases of the design, it may be desirable to store an assumed value for a piece of information that will be calculated later. For example, the net area of a beam can be calculated by using the area of the beam and information related to the connections at the ends of the beam. However, it may be desirable to store the assumed net area of a beam that was used before the connection is designed.

4.1.6. Composition

AP230 only provides decomposition hierarchies, i.e., a design assembly can be decomposed into design assemblies, members (struc_member), or connections (struc_connection). However, AP230 does not allow composition hierarchies, i.e., it is not possible to add a secondary member to a member with a connection between them. The only way to do this is to create a design assembly composed of the primary member, secondary member and connection between them. For example consider the column with a seat angle attached shown in Figure 9.

To describe the column shown in Figure 9 using AP230, an artificial assembly must be created. This assembly will consist of two members and a connection between these members. Another way to describe this column is to simply state that a secondary member (the seat angle) is attached to the column. This description does not require the usage of an artificial assembly. We believe that since the latter description does not require the usage of an artificial entity, it is simpler to use and is more desirable in most circumstances.

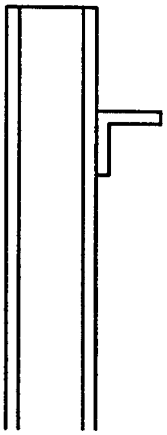

Figure 9. A simple column

4.1.7. Functional decomposition

Even though AP230 supports decomposition, functional relations between the entities cannot be represented. For example, the only way to distinguish between a primary member and a secondary element used for a connection is to check the geometry of the members. Another common case occurs when more than two secondary members are attached to a single primary member. Such a case is shown in Figure 10., where two seat angles are attached to a column.

The column in Figure 10 can be described using an assembly that consist of three members and two connections. Even though the two secondary members and connections are functionally independent of each other (e.g., analysis of one connection does not require knowledge about the other), they are not grouped as independent of each other. The only way to analyze this decomposition is to analyze the geometry and connectivity its members.

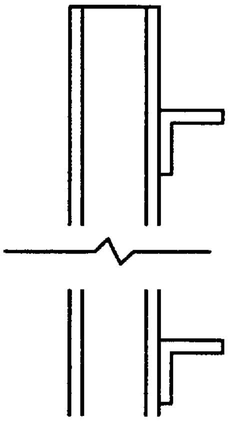

Figure 10. A column with two seats

4.1.8. Strong connectivity between views

Even though AP230 provides some links between the entities in three views (analysis, design, and manufacturing) of the structure, these links are not complete. For example, there are no explicit links between the nodes in an analysis model and their corresponding connections in the design assembly.

4.1.9. Decomposition of connections

Sometimes it is desirable to view a connection as made up of multiple connections and parts. For example, the connection C7 in Figure 7 can be viewed as made up of a secondary member and three subconnections.

4.1.10. Decomposition of members

Similar to connections, a member can be decomposed into sub–members. In AP230, only assemblies can be decomposed into smaller entities.

4.1.11. Built–up members

Currently AP230 does not support built–up members and we believe built–up members are very important for US practice. Built–up members are special cases of members that can be decomposed into sub–members: the sub–members in built–up members constitute a decomposition of the built–up member. For example, a double angle member can be decomposed into two sub–members, each consisting of one angle.

4.1.12. Virtual segments

Even where there is no physical division of a member into sub–members, it may be desirable to divide a member into segments for analysis purposes. For example, the net area of a beam may be different at each end of the beam. In such cases, segments provide a convenient way to represent this information. Also, segments can be used during the calculation of effective buckling length by considering portions of a beam between braceing points as segments.

4.2. Proposed model

Based on these requirements, we first developed an alternative data model to the one is used in AP230. This model is described in this section.

The major entities and relations between these entities in the model are shown in Figure 11. There are eleven major entities in this model. These entities are:

- Object;
- AndDescription;

- OrDescription;
- LocatedObject;
- Location;
- Connection;
- Component;
- Assembly;
- Structure;
- Member; and
- Properties.

In the next eleven sections, these entities are described.

4.2.1. Object

An object is the supertype of the entities Component and Connection. An Object may be decomposed into multiple objects. In AP230, the decomposition is only allowed for assemblies, i.e., members and connections cannot be decomposed into other members and connections. An example of a relation that describes decomposition in AP230 is shown in Figure 12.

As shown in Figure 12, a design assembly may be decomposed into other design assemblies. However, such a relation does not exist for members and connections. In our model, since

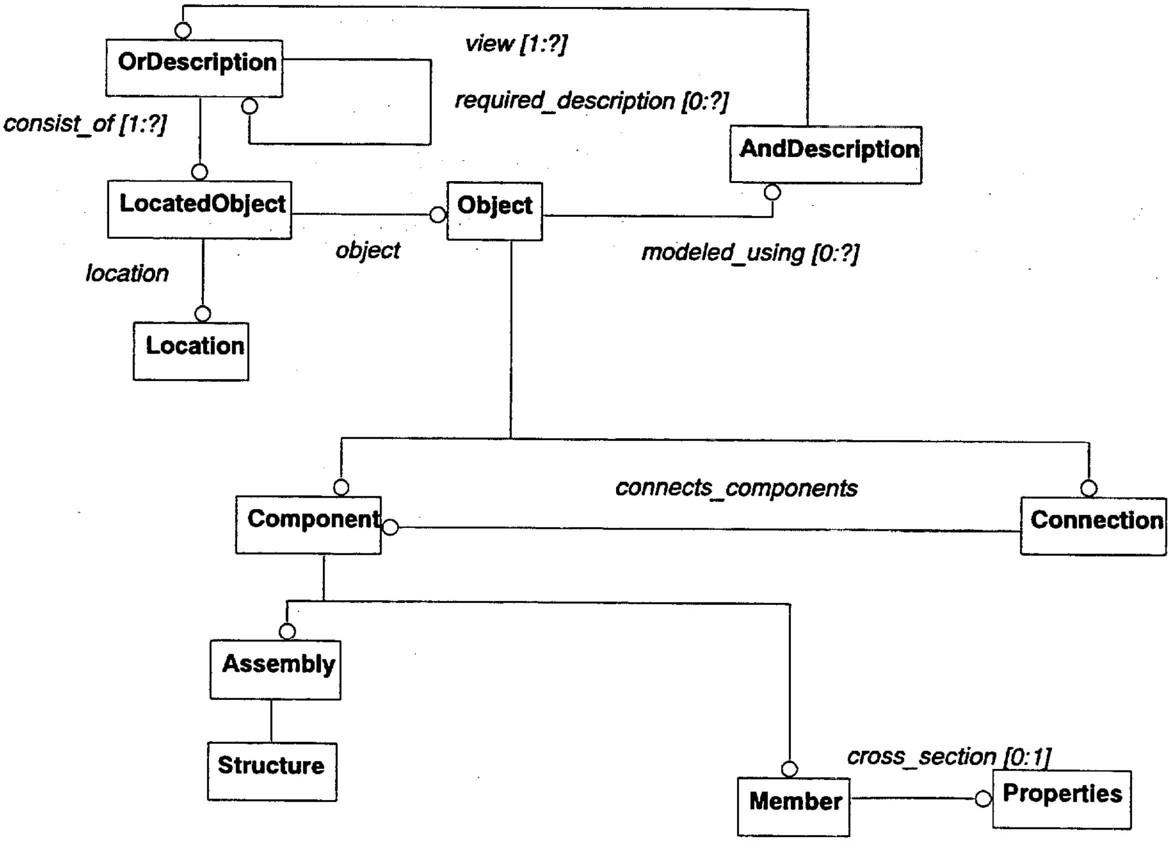

Figure 11. The Alternate Data Model

members and connections are subtypes of the entity Object, they can also decomposed into other members and connections. The decompositions are described using the entity *AndDescription.*

4.2.2. AndDescription

As shown in Figure 12, the decomposition of assemblies in AP230 is defined using simple relations, i.e., using relations such as *decomposes_into_da* that provide a link from an assembly to another assembly. In the alternate model, the decomposition of an object into multiple objects is described using the entity *AndDescription.* There are two major reasons why we used this entity to describe decomposition rather than using a simple relation as it is in AP230:

- The usage of *AndDescription* provides a convenient way to store information about composition as well as decomposition. For example, the column shown in Figure 9 can be described by stating that a secondary member is attached to the column. The type of AndDescription in this case becomes a composition. An example of how this can be done is described in Section 4.3.
- The usage of *AndDescription* provides a convenient way to store functional information about the decomposition. For example, the column shown in Figure 10 can be described by stating that the column has two *AndDescriptions.* Using these *AndDescriptions,* it is possible to state that the column has two independent attachments.

The alternate model supports four types of *AndDescriptions*:

- Composition;
- TransverseDecomposition;
- LongitudinalDecomposition; and
- MixedDecomposition.

Compositions are used to describe the objects that are attached to other objects. The difference between a decomposition and composition is that a decomposition describes an entity by defining the entities of which the entity is made; whereas a composition defines the entities that are attached to that entity.

TransverseDecompositions are used to describe members that are decomposed into transverse members, e.g., a built–up member made of two double angles can be described using a transverse decomposition.

LongitudinalDecompositions are used to describe members that are decomposed into longi-

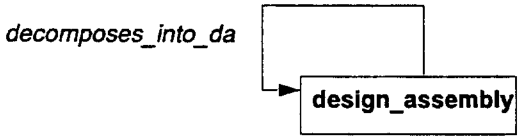

Figure 12. Decomposition in AP230

tudinal members, e.g., a beam that is made from multiple members linked end–to–endcan be described using a longitudinal decomposition.

MixedDecompositions are used to describe decompositions that are neither transverse nor longitudinal. Assemblies and connections are usually described using such decompositions.

Rather than classifying decompositions that are neither transverse nor longitudinal, these four types can be easily extended to cover other commonly occurring decompositions.

4.2.3. OrDescription

OrDescription provides support for multiple views of the same object. As stated in Section 4.1., in steel structures, an entity such as a member can have multiple views. For example, a built–up member can be perceived as a single entity or as an entity broken into transverse elements for analysis purposes. OrDescriptions allow the coexistence of these multiple views.

4.2.4. LocatedObject

In AP230, the location and the function of an object is an intrinsic property of that object, i.e., the location and the function of an element (such as a member) is fixed. It is not possible to look at the same object from multiple views. Since the new model supports multiple views, the location and the function of an object is dependent on the object of which it is part. For example, as shown in Figure 6, a structure may have multiple views. The location of the top left connection (C9) is dependent on the view of which it is part. In the first view, the location of this connection must be defined relative to the structure. In the second view, the location of this connection must be defined relative to C7. Similarly, a member can be perceived as primary member in one view and secondary in another. LocatedObject provides a convenient way to describe information of an object dependent on the view. Information that does not change, such as the cross section of a beam, is attached to the object itself.

4.2.5. Location

This entity defines the location of an object in a coordinate system.

4.2.6. Connection

In the new model, a connection is a subtype of an object. This means that a connection may be decomposed into multiple objects. For example, consider the connection shown in Figure 13.

The connection shown in Figure 13 connects three primary members. Also, the same connection may be decomposed into multiple smaller connections and secondary members used in the connections. Some of these connections can also be decomposed into smaller connections and members. A connection that is not decomposed into smaller connections becomes a joint system that uses a bolt system or weld system.

Components that a connection connects together are described using the relation *connects_components*.

4.2.7. Component

Another subtype of object is a component. Since a component is a subtype of object, it can be decomposed into multiple objects. The new model supports two types of components: assemblies, and members.

4.2.8. Assembly

An assembly is a subtype of component. The new model supports five types of assemblies: design assemblies, manufacturing assemblies, fabrication assemblies, erection assemblies, analysis assemblies.

4.2.9. Structure

A structure is a subtype of assembly. A structure can be decomposed into other structures. The difference between a structure and assembly is that a structure is a self–supporting assembly.

4.2.10. Member

Member is a subtype of component. Members can be prismatic or nonprismatic. Members can be divided longitudinally into other members. Such members are called *segments*. There are two types of segments:

- Ordinary segments. Ordinary segments occur when a member is physically divided into smaller segments. For example, the ordinary segments shown Figure 14 are formed because Member1, which is represented as a single continuous member in a conceptual design program such as SEED–Config, is physically divided into two smaller pieces.
- Continuous Segments. Continuous segments occur when a member is divided into smaller components for analysis or design purposes. In such segments, there are no physical divisions of the member into multiple pieces. An example of such a segment is

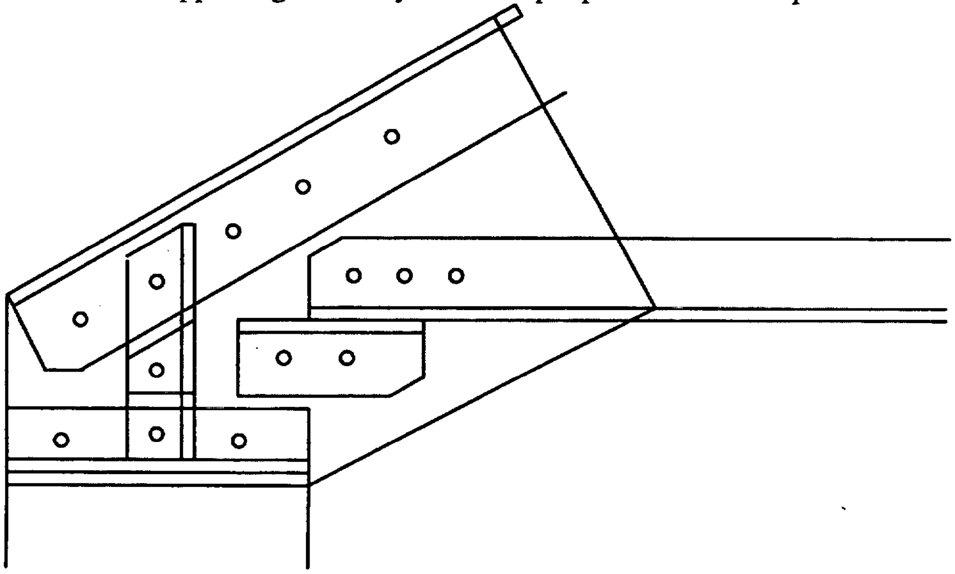

Figure 13. A Complex Connection

shown in Figure 15. Such segments can also be used to define properties that change along a member even though there is no connection. For example consider the beam shown in Figure 16. As shown in this figure, even though there is no connection, a member can be divided into smaller continuous segments to represent different net areas of the member.

A member can also be divided into transverse segments for analysis or design purposes. Transverse segments have the same length as the original member. Transverse segments are usually used for built–up members. In LRFD segment components are referred as elements.

4.2.11. Properties.

This entity is used to store cross sectional properties of members.

4.3. Example

In this section, an example will be used to illustrate the model we developed. In this example,

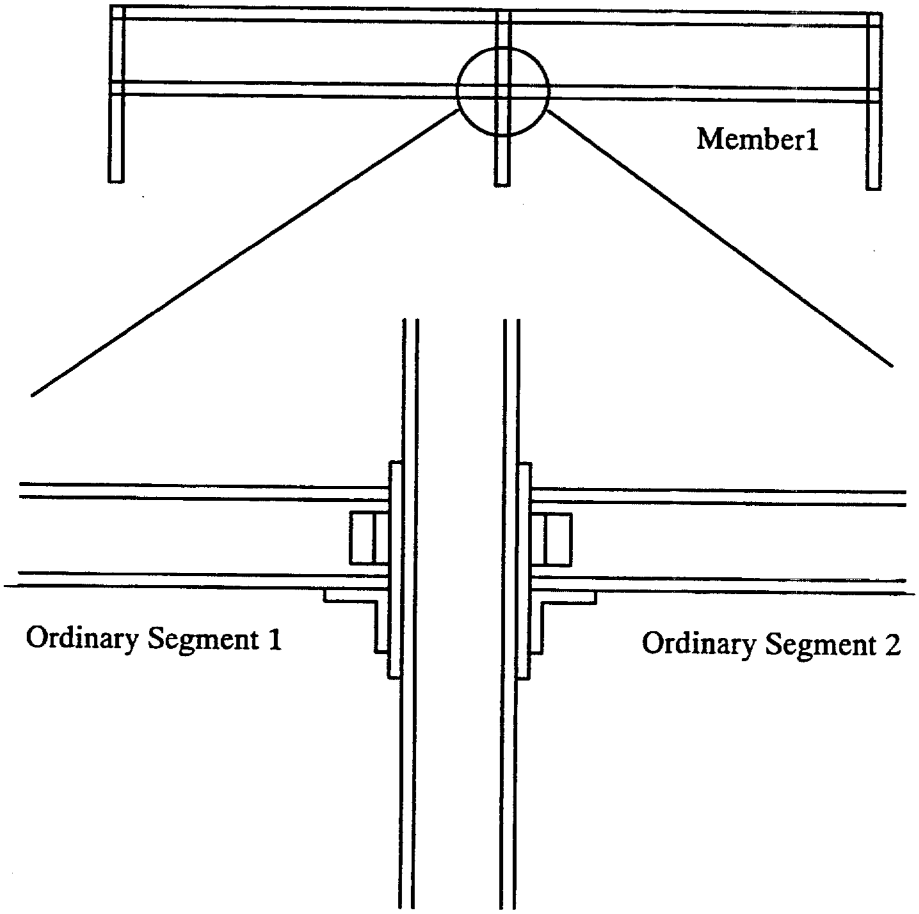

Figure 14. Ordinary Segment

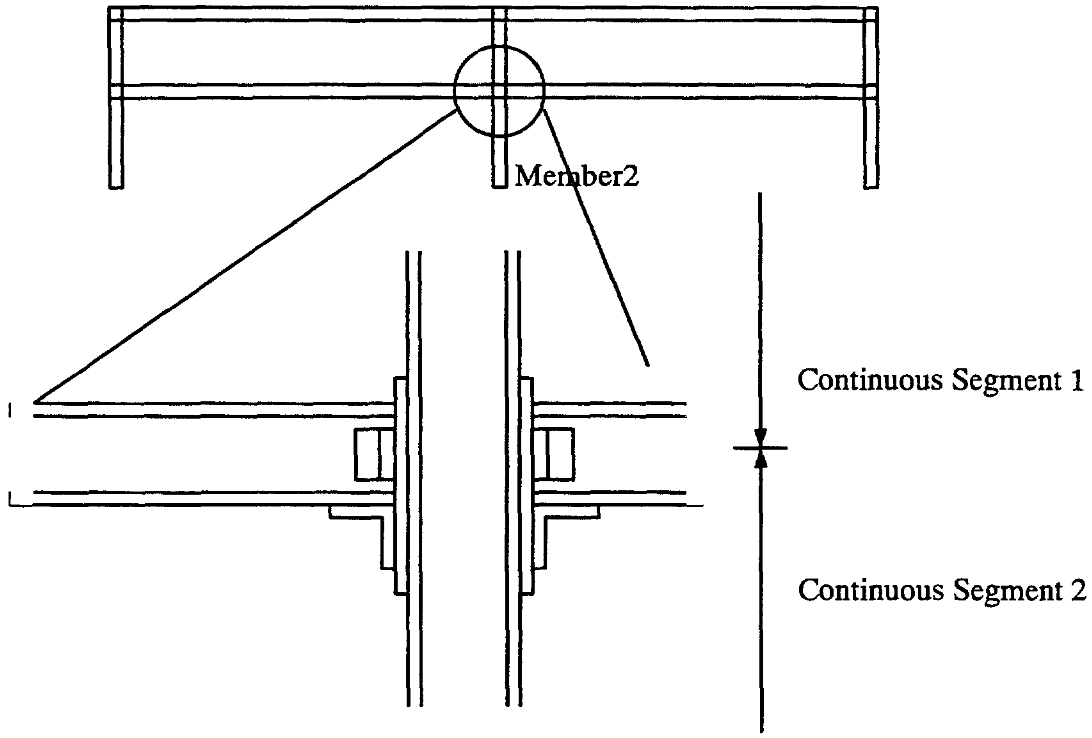

Figure 15. Continuous Segment

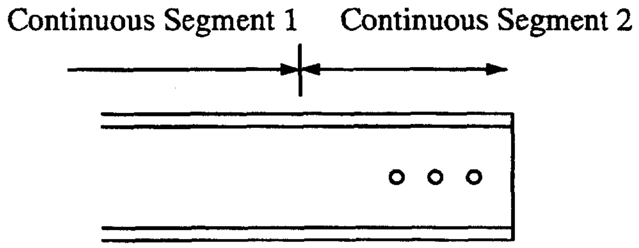

Figure 16. Continuous Segments

the simple structure shown in Figure 5 will be described using the model we developed.

As described in Section 4.1.2., this structure has at least three logical representations: the first one, shown in Figure 6, has five members and six connections. The second one, shown in Figure 7, has three members and two connections. The third one, shown in Figure 8, has three members and two design assemblies and four connections.

In this example, we will assume that the designer, at the conceptual stage of the structure, does not know whether the structure will require the secondary members (M2 and M4) for the required stability of the structure. Thus, the designer develops a structure with three members and two connections that do not contain any secondary members. This view corresponds to the view shown in Figure 17.

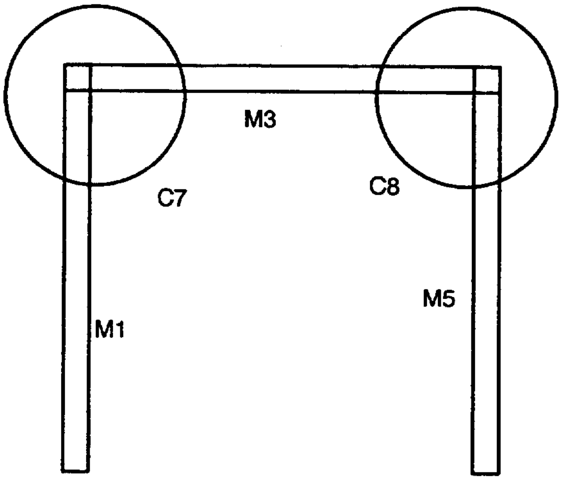

Figure 17. Initial view the simple structure

As shown in Figure 17, the structure contains three members and two connections. At this stage the details of the connections and members are not defined. The partial graphical description of the model of this structure is shown in Figure 18.

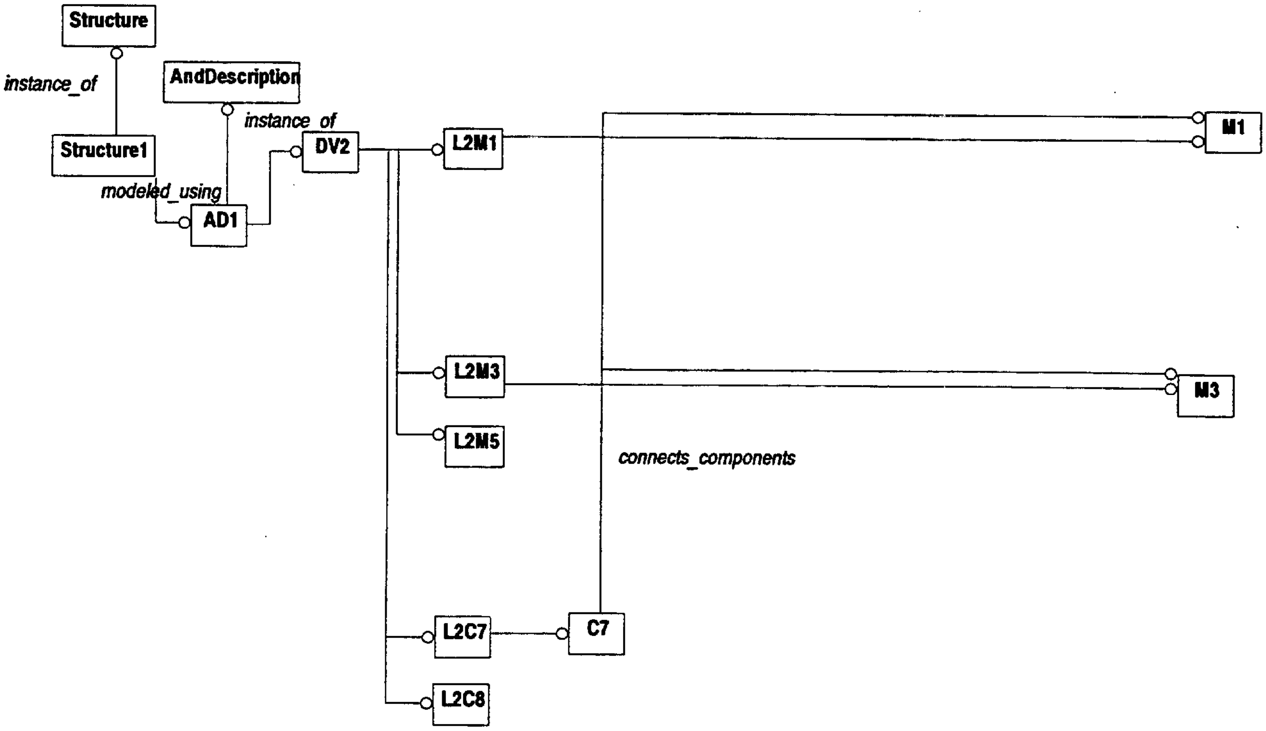

Figure 18. Initial description of the structure

As shown in Figure 18, the structure, identified as Structure1, is modeled using a single And-Description (AD1), which describes a decomposition. In this stage, AD1 has a single view

(DV2) (others will be added later). In this view, the structure is decomposed into three members and two connections. The locations and functions of these objects are defined using five LocatedObjects (L2M1, L2M3, L2M5, L2C7, and L2C8). Members M1, M3 and M5 (to simplify the figure, M5 is not shown in the figure) and Connections C7 and C8 (to simplify the figure, C8 is not shown in the figure) are the actual objects that are used to define the properties of the members and connections that do not change from view to view. For example, The information that C7 is used to connect M1 and M3 is independent of the view.

At this stage, let us assume that the designer decides that the connections will probably require secondary support members and decides to design them.

One of the possible connection designs is shown in Figure 19.

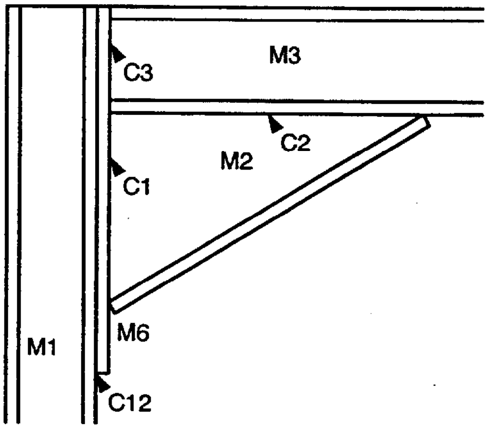

Figure 19. Partially designed connection

As shown in Figure 19, this connection requires two secondary members (M6, which is a plate and M2, which is cut from a T beam) and four sub–connections (C1, which connects M2 to M6, C2, which connects M2 to M3, C3, which connects M3 to M6, and C12, which connects M6 to M1).

As described in 4.1.2., entities may have multiple views. Six possible views of this connection are shown in Figure 20 – Figure 25.

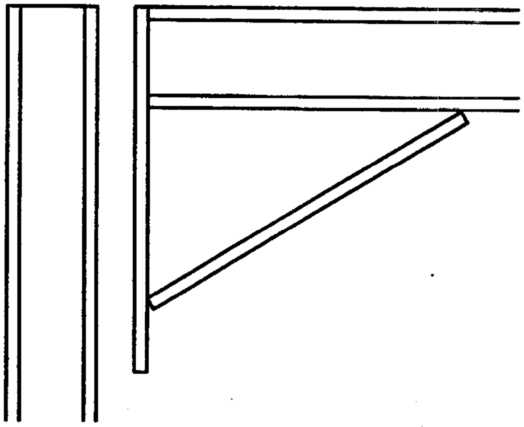

Figure 20. First view of the connection

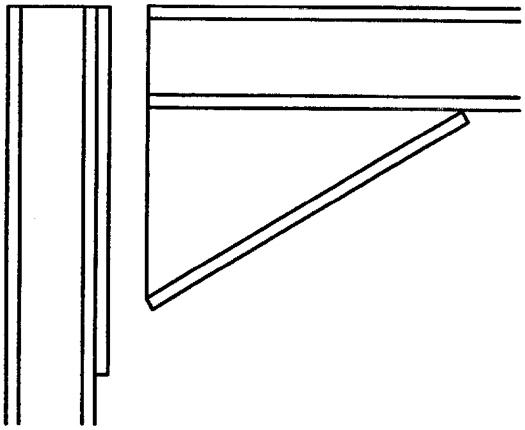

Figure 21. Second view of the connection

Let us assume that the designer decides the view shown in Figure 22 (the third view) makes more

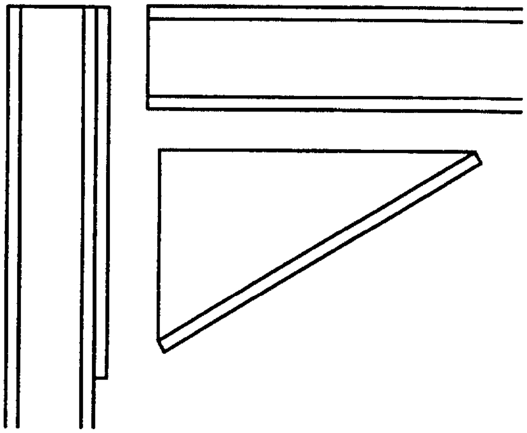

Figure 22. Third view of the connection

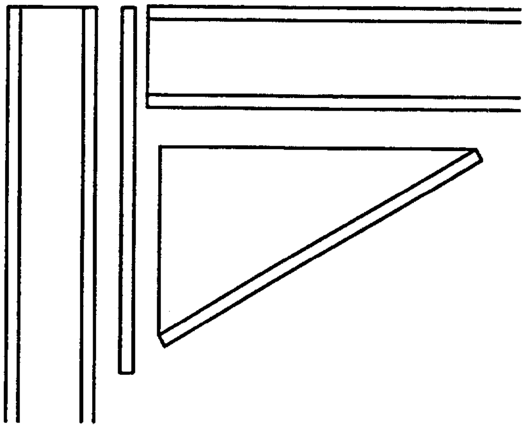

Figure 23. Fourth view of the connection

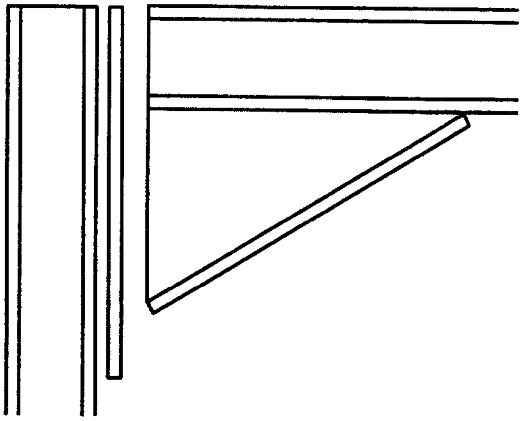

Figure 24. Fifth view of the connection

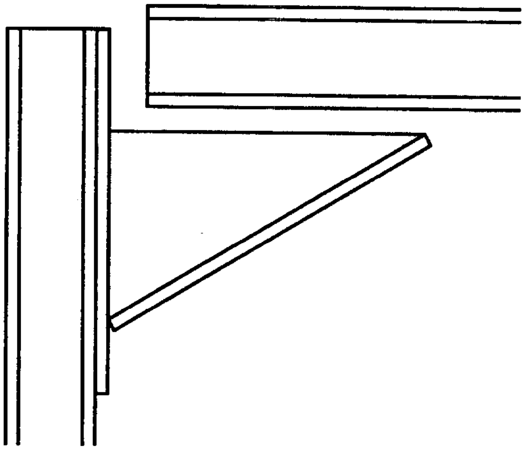

Figure 25. Sixth view of the connection

sense for her purposes. The description of the structure with this view of the connection is shown in Figure 26.

As shown in Figure 26, the design of this connection only requires changes in affected entities, i.e., in the members M1 and M3 and the connection C7. As shown in Figure 26, there are two sets of entities that are created. The entities in the first set are attached to the member M1. The entities in the second set are attached to the connection C7. The names of the entities used to define this connection are shown in Figure 27.

The entities in the first set (AD3, DV6, L4C12, L4M6, C12, and M6) are used to describe the member M1 with the plate attached to it. To be able to do that, the description uses an AndDescription (AD3), which defines a composition. Notice that AD3 is not a decomposition because the plate will be considered attached to M1. The usage of composition eliminates the need of creating an object representing an artificial de-

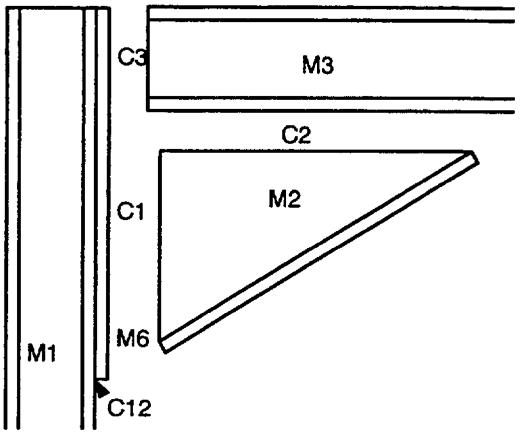

Figure 27. Third view of the connection with entities identified

sign assembly that will be decomposed into the member M1 and the plate (M6). AD3 has a single view DV6, which contains two located objects (L4C12 and L4M6). L4M6 defines the location of the plate attached to M1 and L4C12 defines the location and function of the connection C12, which connects the member M6 to M1.

The entities in the second set (AD2, DV4, L3M2, L3C1, L3C2, L3C3, M2, C1, C2, and C3) are used to describe the secondary member M2, which is connected to M1 and M3, and the connections between these members. This description uses an AndDescription (AD2), which is a decomposition. This decomposition has a single view (DV4). DV4 is decomposed into four located objects: L3M2, the LocatedObject that defines the secondary member M2, and three connections, C1, C2, and C3, which connect M3 to M1, M2 to M3, and M2 to M1, respectively.

At a later stage, let us assume that the fabricator decides to use the second view of the connection (the one shown in Figure 21). The addition of this view to the description of the structure does not require changes in previous view. The de-

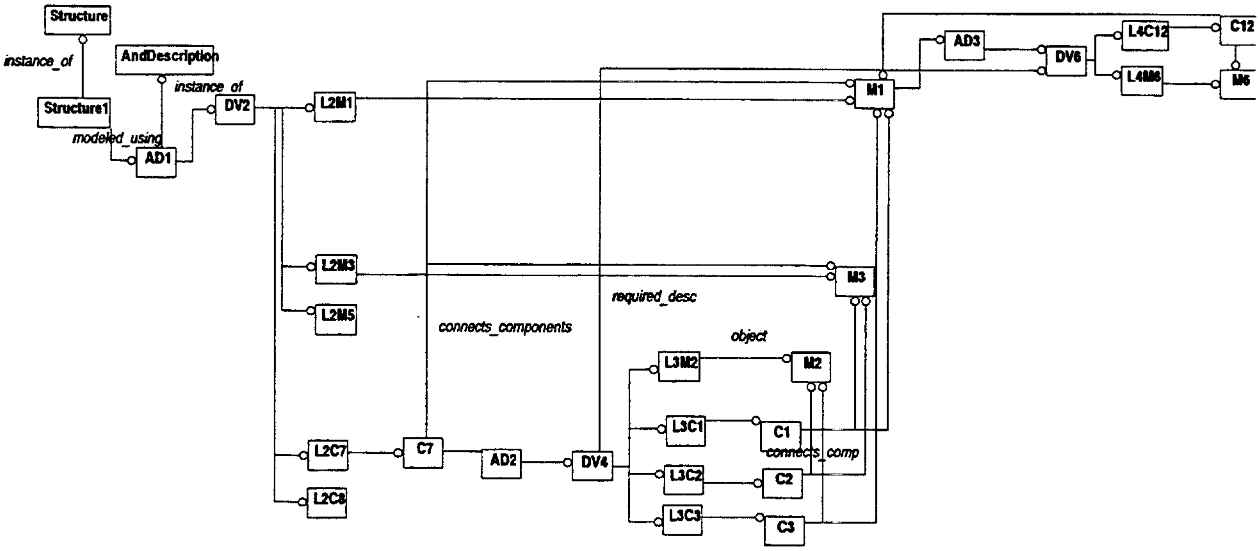

Figure 26. Description of the structure with the third view of the connection

scription of the structure with this view is shown in Figure 28.

As shown in Figure 28, AD2, which is used to describe C7, is attached to another view (DV5). This view has a single connection (C14) that connects M1 to M3. This connection can be decomposed into two sub–connections C1 and C3. To be able to describe this view, M2 is also considered as attached to M1.

Now let us assume that the designer wants to perform an analysis of the structure. For analysis purposes, she would like the member M2 (the secondary member in the connection) to be considered as a primary member. This view corresponds to the representation of the structure shown in Figure 6 (the first view of the structure). To be able to represent this view, the designer has to create another view and attach it to AD1. This representation is shown in Figure 29. In this figure only connections at the left side of the structure are shown. Since the connections and secondary members at the right side are very similar to the ones on the left side, they are not shown so as to simplify the figure.

As shown in Figure 29, the addition of this view does not require changes in previous views of the structure. In this view, five new entities (L1M2, L1M4, L1C1, L1C2, L1C3) are created. These entities are LocatedObjects. They are used to define the location of the secondary members and connections in the coordinate system of the structure. Notice that the addition of this view does not require the creation of new members and connections. For example, the connection C1 in both views is de-

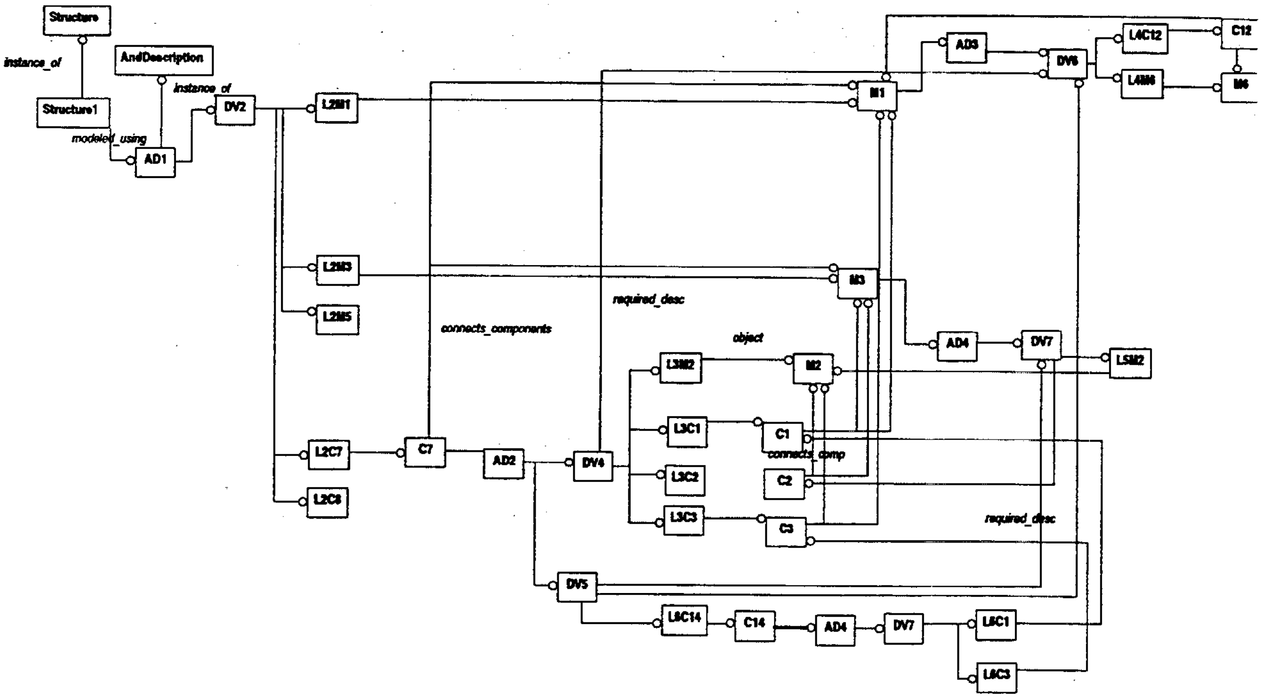

Figure 28. Description of the structure with the second and third view of the connection

fined in the same way except that the location of the connection is dependent on the view. In the first view, the location of the connection C1 was defined in the coordinate system of the connection C7. In the second view the location of the connect C1 is defined in the coordinate system of the structure. Due to this reason, the new located objects rather than duplicating the old objects, can directly point to the old objects.

This view of the structure is however incompatible with most of the analysis programs available to perform structure analysis. Most of these programs can only deal with connections that occur at the ends of members, i.e., they cannot deal with connections that occur within the members.

To satisfy the need of these programs, the structure can be represented as shown in Figure 30.

As shown in Figure 30, this representation is created by adding a third view to AD1. This view has eight members and six connections (To simplify the figure only five of them are shown in the figure). In this view, each primary member (M1, M3, and M5) is divided into continuos segments and virtual connections are created between these segments. For example, the member M1 is divided into two continuos

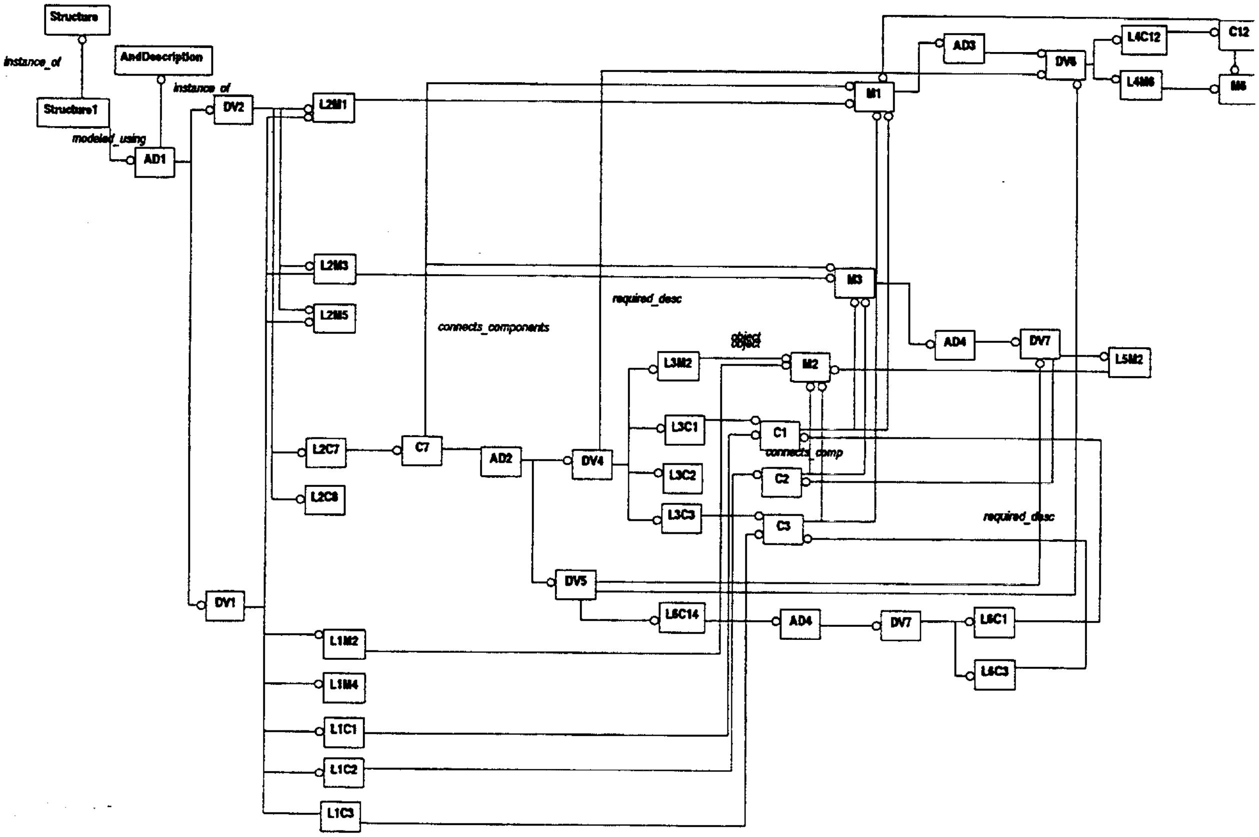

Figure 29. Description of the first view of the structure

segments (M1a and M1b). This decomposition is added as an AndDescription to M1.

The description of the connections in this view can be described using the connections already defined in other views with minor additions. For example, the connection C3x (in this view, C3x is a connection between M1a, M1b, and M2) can be described using a decomposition (AD9) that has a single view (DV11) that uses the connection C3.

Let us assume that the designer using the analysis results wants to design M1 as a built–up

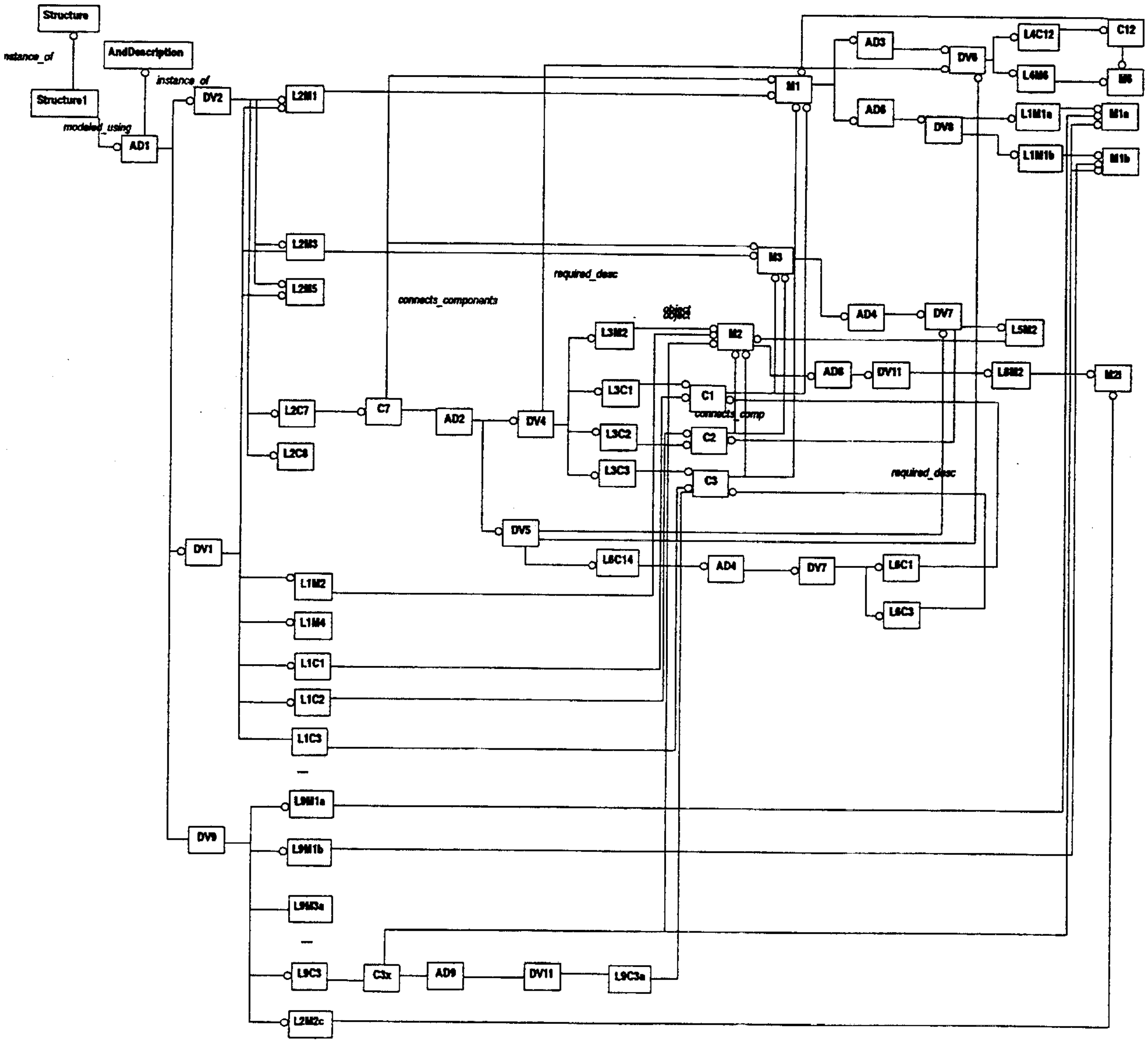

Figure 30. Alternative description of the structure for analysis purposes

member. This information can be directly added to M1 as an AndDescription as shown in Figure 31 .

In this view, M1 is divided into three transverse members. These members are the two flanges and the web (M1w1, M1f1, and M1f2) of the built–up member.

As this example shows, entities in this model can have multiple views in which members and connections be decomposed into smaller elements.

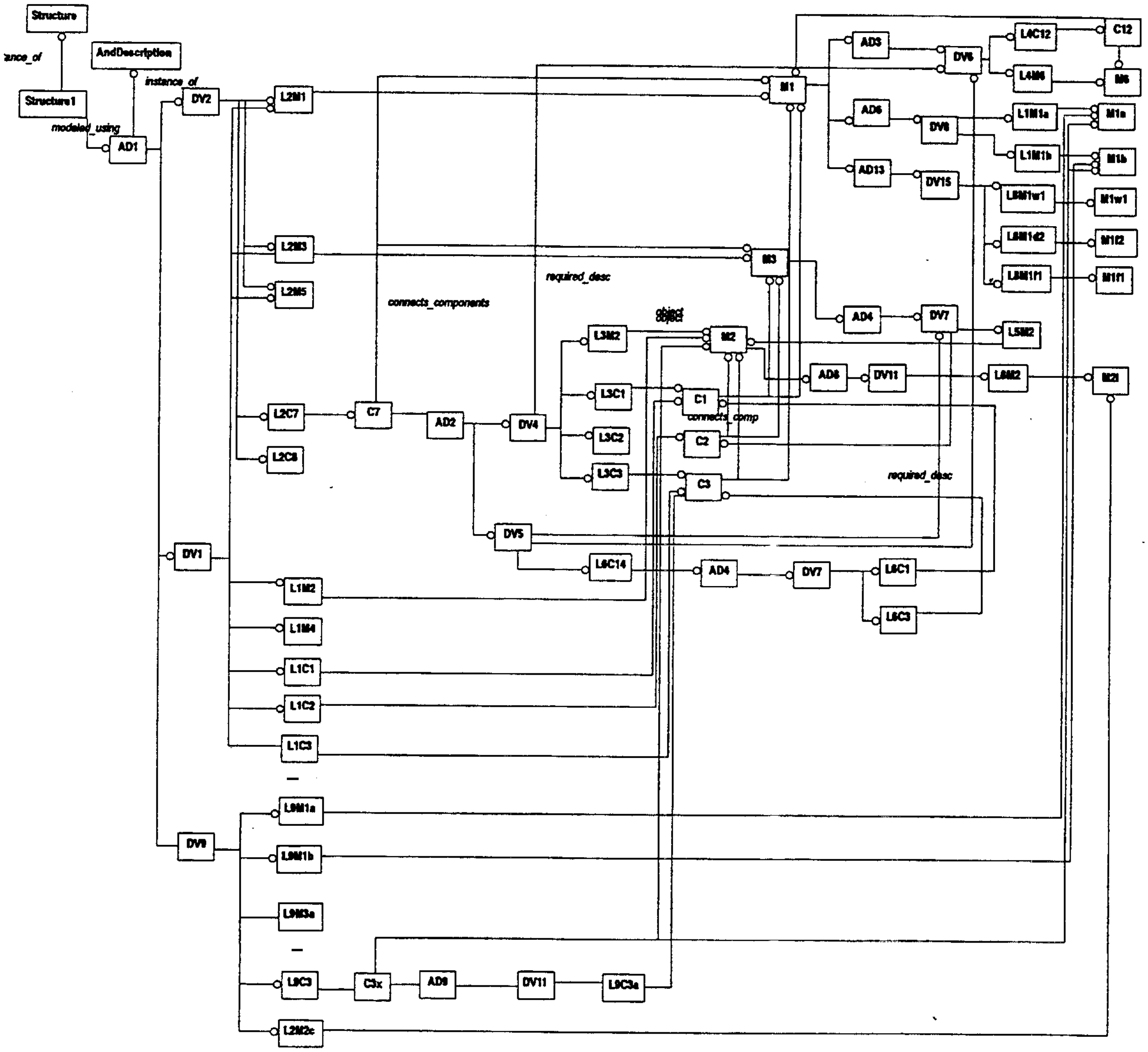

Figure 31. Decomposition of M1 into transverse elements

5. Integration of the Alternate Model into AP230

We realize this model is not a complete model for steel structures. For example, information about loading and analysis results is not yet defined in the model. We never intended to propose a new model to replace AP230. The purpose of this model was first to illustrate the deficiencies of AP230 and then extend AP230 to eliminate some of these deficiencies. In this section, we present how this model could be integrated into AP230.

5.1. Overview of the model

In this section, an overview of the integrated model is presented by comparing it to the model used in current version of AP230 shown in Figure 32.

The model obtained after the integration of the alternate model into AP230 is shown in Figure 33 .

A comparison of these two models reveals that the differences can be classified in four categories:

- entities copied from the alternate model described in Section 4.2.;
- entities present in the alternate model described in Section 4.2. but removed in the integrated model;
- entities removed from AP230; and
- entities present in AP230 but renamed in the integrated model.

In the following four sections, these categories are described:

5.1.1. Entities copied from the alternate model

As shown in Figure 33, the integrated model contains some of the entities described in Section 4.2. These entities are:

- Object;
- LocatedObject;
- Location;
- Connection;
- Component;
- Assembly;
- Structure; and
- Member.

In the next eight sections, these entities are described.

5.1.2. Object

An object is still the supertype of the entities Component and Connection. An Object may be decomposed into multiple objects. In AP230, the decomposition is only allowed for assemblies, i.e., members and connections cannot be decomposed into other members and connections. In this model, since members and connections are subtypes of the entity Object, they can also decomposed into other members and connections. The decompositions are described using the relation of *decomposes_into.*

5.1.3. LocatedObject

In AP230, the location and the function of an object are intrinsic properties of the object, i.e., the location and the function of an element (such as a member) is fixed. Since the alternate

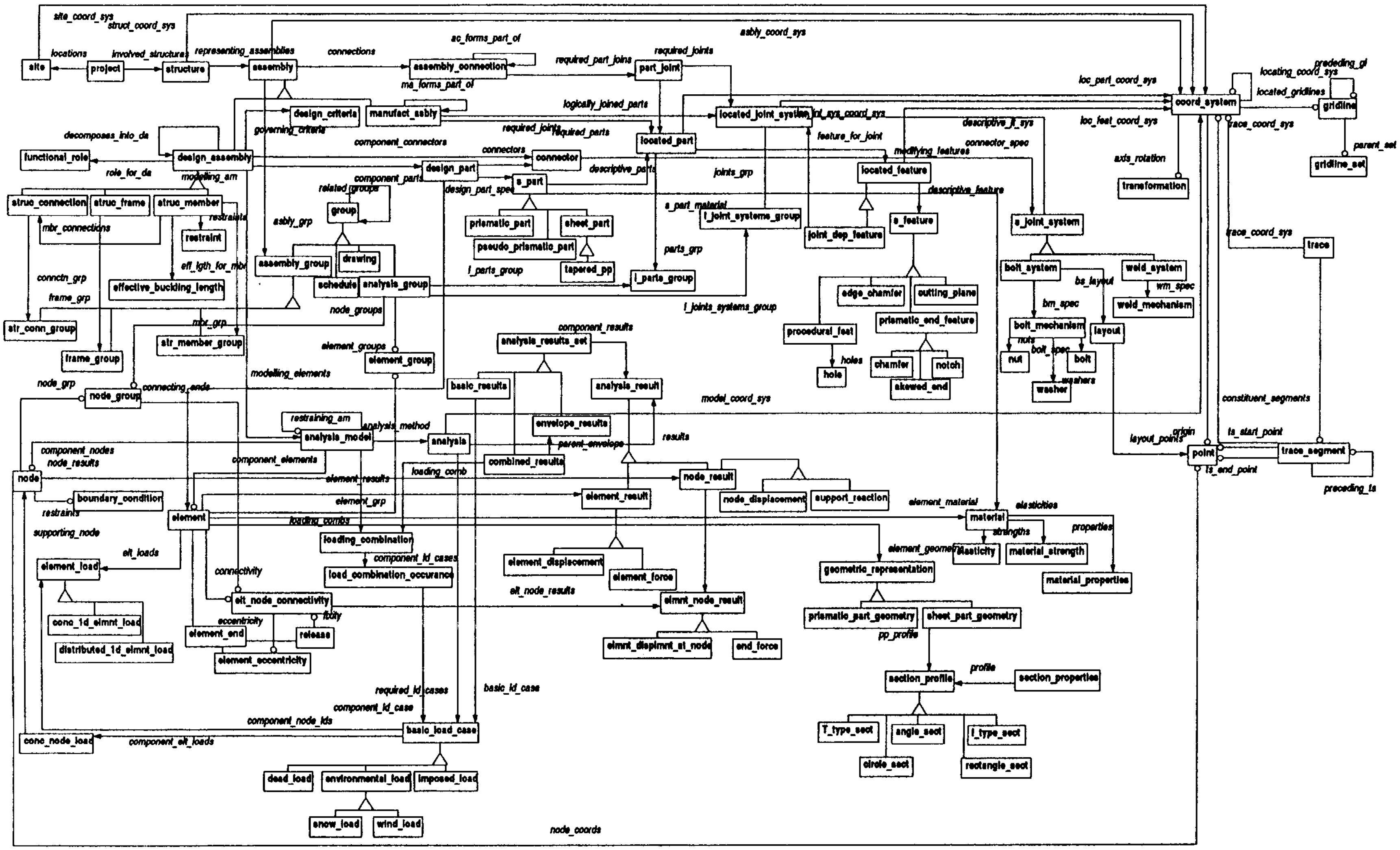

Figure 32. The building model used in AP230

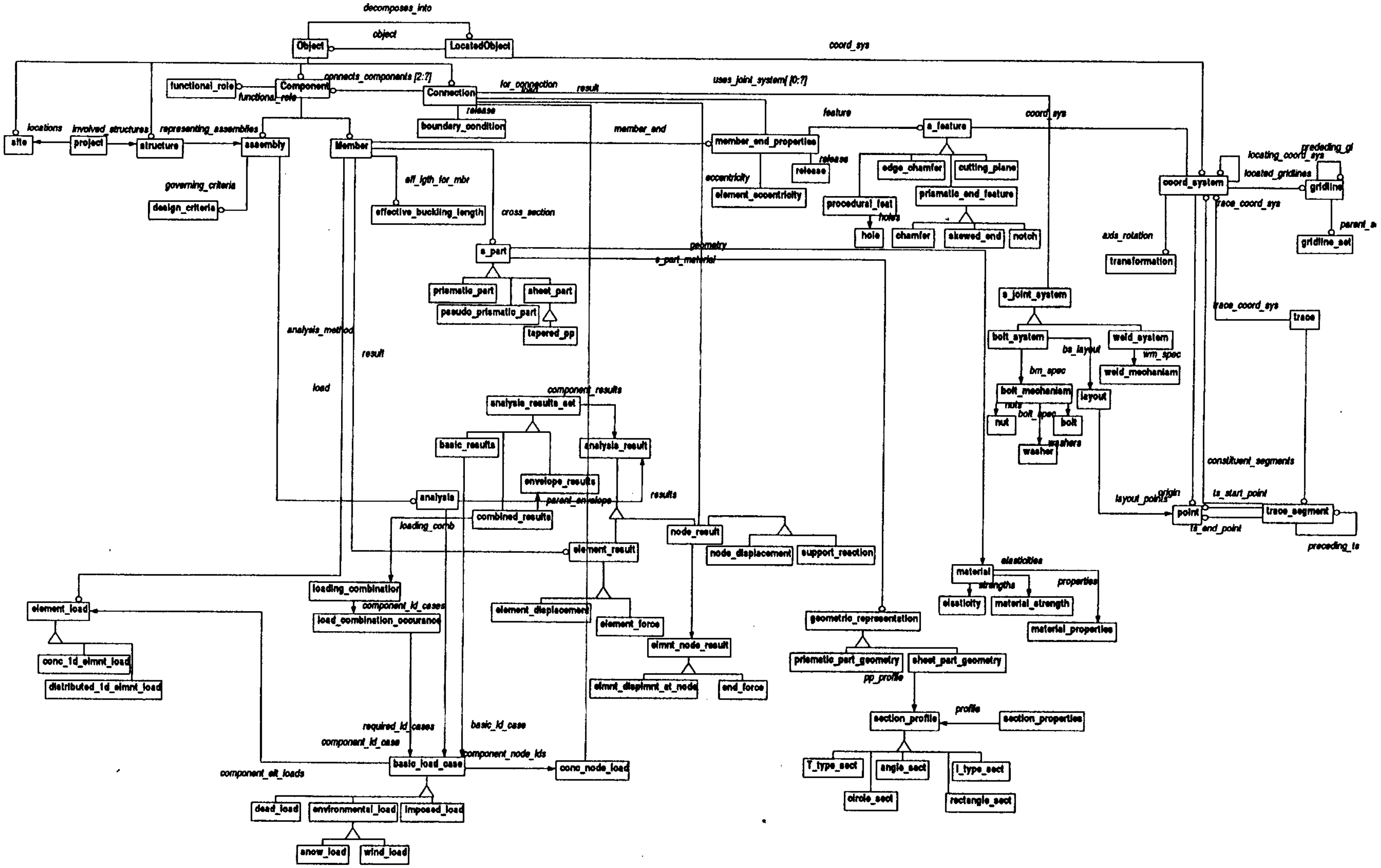

Figure 33. Integration of new model into AP230

model supports multiple views, this entity was created to describe the location and the function of an object that is dependent on the view. We believe it is better to separate the information that describes intrinsic properties of the objects (such as the cross section of a beam) from its location.

5.1.4. Location

This entity defines the location of an object in a coordinate system.

5.1.5. Connection

In the integrated model, a connection is a subtype of an object. This means that a connection may be decomposed into multiple objects.

A connection that is not decomposed into smaller connections becomes a joint system that uses a bolt system or weld system.

Components that a connection connects together are described using the relation *connects_components*.

5.1.6. Component

Another subtype of object is a component. Since a component is a subtype of object, it can be decomposed into multiple objects. The integrated model supports two types of components: assemblies and members. The only restriction on the decomposition of a component is that an assembly can only be the decomposition of another assembly (e.g., a beam cannot be decomposed into an assembly).

5.1.7. Assembly

An assembly is a subtype of component. The integrated model supports five types of assemblies: design assemblies; manufacturing assemblies; fabrication assemblies; erection assemblies; analysis assemblies.

5.1.8. Structure

In the integrated model, to facilitate the integration of the alternate model into AP230, a structure is defined as an independent entity. In the alternate model a structure was a subtype of assembly. The decomposition of a structure into other structures is described using the relation *representing_assemblies*.

5.1.9. Member

In the integrated model, a member retains the properties described in the alternate model. It is a subtype of component. Members can be prismatic or nonprismatic. Members can be divided longitudinally into segments (either ordinary or continuous). A member can also be divided into transverse segment components for analysis or design purposes. Transverse segment components have the same length as the original member.

5.2. Entities removed from the alternate Model

Integration of the alternate model into AP230 was a difficult process because the two models were very different from each other. To allow the integration of these models, we had to sacrifice some of the advantages of the new model. As shown in Figure 33, some entities present in the alternate model described in Section 4.2. are removed in the integrated model. These entities are:

- AndDescription;
- OrDescription; and
- Properties.

In the following sections, the reasons of removing these entities are presented.

5.2.1. AndDescription

In the integrated model, the entity AndDescription is replaced with a simple relation. Usage of a relation rather than AndDescription and OrDescription eliminates some benefits of the alternate model but facilitates the integration. The benefits that were eliminated by this simplification are:

- The usage of *AndDescription* provided a convenient way to store information about composition as well as decomposition. The usage of a simple relation allows only decomposition.
- The usage of *AndDescription* provided a convenient way to store functional information about the decomposition. Again the usage of a simple relation does not allow a way to describe the function of the decomposition.

5.2.2. OrDescription

The *OrDescription* was created to provide support for multiple views of the same object. As stated in Section 4.1., in steel structures, an entity such as a member can have multiple views. For example, a built–up member can be perceived as a single entity or as an entity broken into transverse elements for analysis purposes. OrDescriptions allowed the coexistence of these multiple views. By removing OrDescription, it is no longer possible to describe multiple views of the same entity. However, by using the relation *representing_assemblies,* it is still possible to describe different views of the structure as a whole.

5.2.3. Properties

AP230 contains a large set of classes to describe the properties of members and connections. Some of these classes are: *s_part,* used to describe cross sectional properties of members, *s_feature,* used to describe features attached or removed from members, *weld_system* and *bolt_system,* used to describe connections; and *material,* used to describe properties related to the material used in members and connections. AP230 contains very detailed explanations for these classes. To facilitate the integration of the alternate model into AP230, we decided that the members and connections should directly reference these classes rather than using the intermediate class, Properties.

5.3. Entities removed from AP230

In the next table, the classes that are removed from AP230 in the integrated model and the classes in the integrated model that can be used instead of these classes are shown.

Class in AP230	Class in the Integrated Model
assembly_connection	connection
part_joint	connection
manufact_asbly	member or assembly

design_assembly	member or assembly
struc_connection	connection
struc_frame	assembly
struc_member	member
node	connection
analysis_model	assembly
element	member
design_part	member
connector	connection
part_joint	member
located_part	located_object and member
located_joint_system	located_object and connection
group (and its subclasses)	assembly

5.4. Renamed Entities

In the integrated model, some entities are renamed. In the next table, the old name and the new names of these entities are shown.

Class in AP230	Class in the Integrated Model
element_load	member_load
conc_1d_elmnt_load	conc_1d_member_load
distributed_1d_elmnt_load	distributed_1d_member_load
element_end	member_end
element_eccentricity	member_eccentricity
elt_node_connectivity	member_end_properties
elmnt_node_result	member_node_result
elmnt_displmnt_at_node	member_displmnt_at_connection
elemnt_result	member_result
element_displacement	member_displacement
element_force	member_force
node_result	connection_force
node_displacement	connection_displacement
conc_node_load	conc_connection_load

5.5. Example

In this section, an example will be used to illustrate the integrated model. In this example, the simple structure shown in Figure 34 will be described using the integrated model.

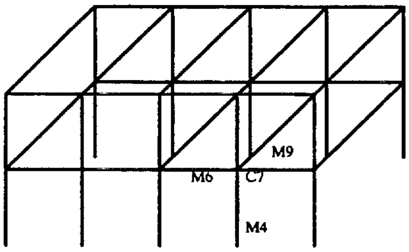

Figure 34. A simple structure

At the conceptual stage, this structure has twenty–three members (in the conceptual stage, the members that span multiple bays and columns that span multiple stories are considered single members). For example, the connection C7 connects M6 (the member that spans four bays), M4 (the column) and M9 (the horizontal member). At this stage of the design, the members are not divided into smaller pieces. The partial representation of this structure using the integrated model is shown in Figure 35.

In Figure 35, only members relevant to the connection C7 are shown. These members are M6, M4 and M9. In this view of the structure, the structure (S1) has a single representing assembly (DA1). DA1 can be decomposed into twenty four members (only four are shown) and twenty connections (only two are shown). One of these connections is C7, which is shown in Figure 36.

As shown in Figure 36, C7 connects connects three members M6, M4, and M9. Also, as shown in the figure, at this stage, these members are not broken into smaller pieces.

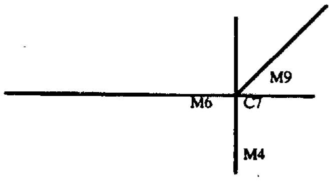

Figure 36. Members connected by C7

For analysis purposes, these members can be broken into smaller pieces. For example, the members connected at the connection C7 can be broken into smaller pieces as shown in Figure 37.

As shown in Figure 37, M6 can be broken into four smaller members (only two are shown) and M4 can be broken into two smaller members. The representation of the same structure for analysis purposes is shown in Figure 38.

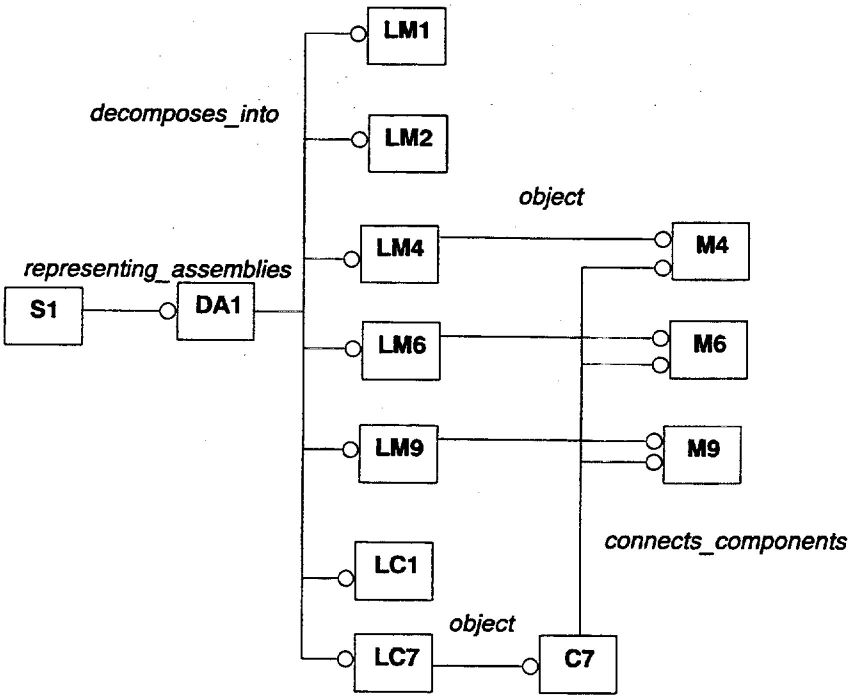

Figure 35. Initial description of the structure

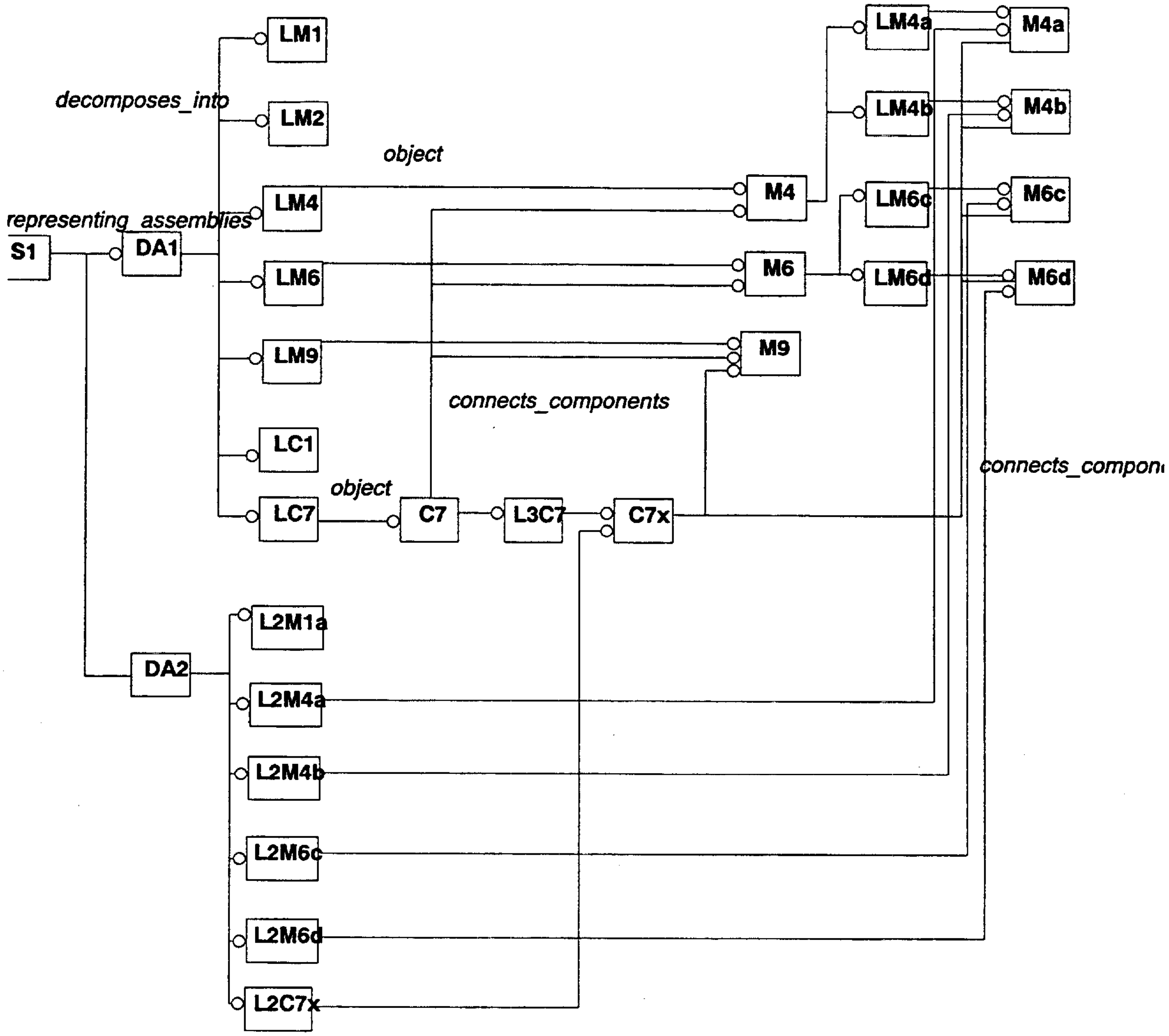

Figure 38. Description of the structure for analysis purposes

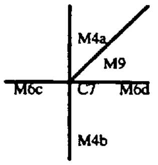

Figure 37. Members broken into smaller pieces

As shown in Figure 38, the structure (S1) has been attached to another assembly (DA2). This assembly has forty six members (only five are shown) and twenty connections (only one is shown).

Unlike the alternate model described in Section 4.2., the addition of a view in the integrated model may require the addition of artificial entities to the representation of the structure. This is mainly because the integrated model does not support multiple views of the same entity. For example, now, the connection C7, which was described as connecting three members in DA1 (the assembly that corresponds to the conceptual stage) must be represented as connecting five members. To be able to do this, first M4 and M6 are divided into smaller members. M4 is divided into two members (M4a and M4b) and M6 is divided into four members (M6a, M6b, M6c, and M6d). In Figure 38, M6a and M6b are not shown. Then, C7 is decomposed into another connection that connects these five members. The new connection has an artificial id (C7x) even though C7 and C7x are actually the same connections at different views because the integrated model does not support multiple views for members and connections.

At the manufacturing stage, the same connection can be designed as shown in Figure 39. The same connection is shown in Figure 40 from a 90 degree rotated view.

The partial description of the structure at this stage is shown in Figure 41.

As shown in Figure 41, the manufacturing assembly is represented using an assembly (DA3), which is attached to the structure (S1) as another representing assembly. In this example, we assumed that all primary members are assembled at site to form the structure. In Figure 41, only one of the members (the column M4) is shown. This member and the secondary members attached at the connections (such as the plates and legs attached at the connection

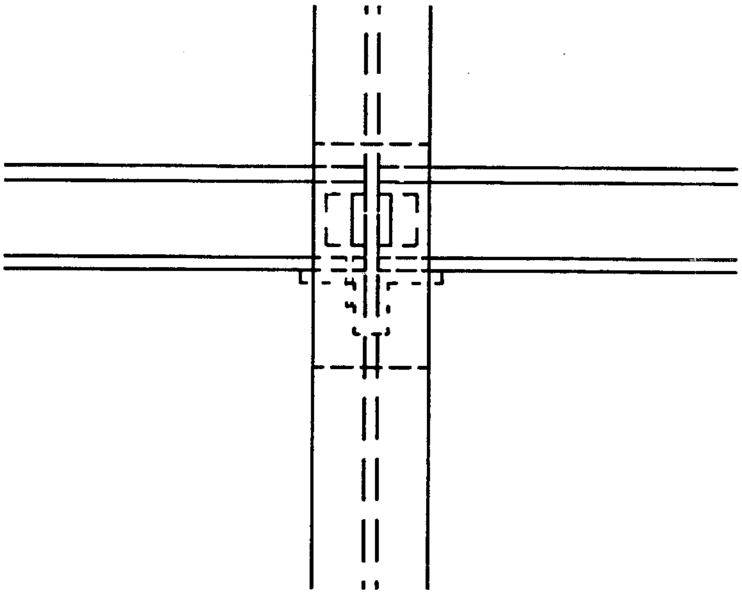

Figure 39. Description of the connection for manufacturing purposes

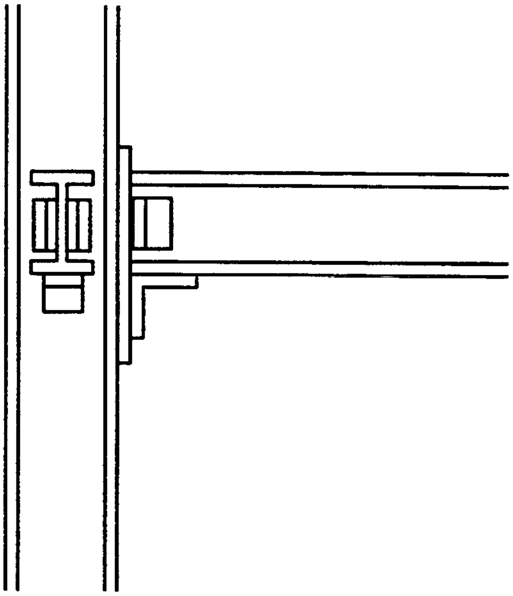

Figure 40. Rotated view of the connection

C7) are represented as a member (M4F). Figure 42 shows this member .

As shown in Figure 42. M4F can be decomposed into column M4, three design assemblies (DA4t1, DA4t2, and DA4t3 (not shown)) and three connections (DA4C1, DA3C2 (not shown), and DA4C3 (not shown)). The design assembly DA4t1 can be decomposed into two secondary members (P1 and L1) and a connection between them (DA4t1C1). This design assembly is attached to the column M4 using the connection (DA4C1).

As shown in this example, a member and a connection can still be decomposed into smaller members and connections. However, to be able to represent this decomposition, artificial assemblies or members must be created. Also, as shown in this example, similar to AP230, different views of the structure can be represented using multiple assemblies.

6. Summary

AP230 is a proposed international standard for the representation and exchange of electronic data relating to structural steel framed buildings. It is still currently under development and not yet accepted as an international standard.

Our major objectives in doing this research were to evaluate AP230 Application Protocol with respect to US practice and extend it for use in early conceptual stages of the facility design process. To be able to evaluate AP230 from US a perspective we decided to:

- perform an informal analysis of AP230;
- develop an analysis module using AP230 as the data model;
- interface AP230 with the SEED project at CMU; and

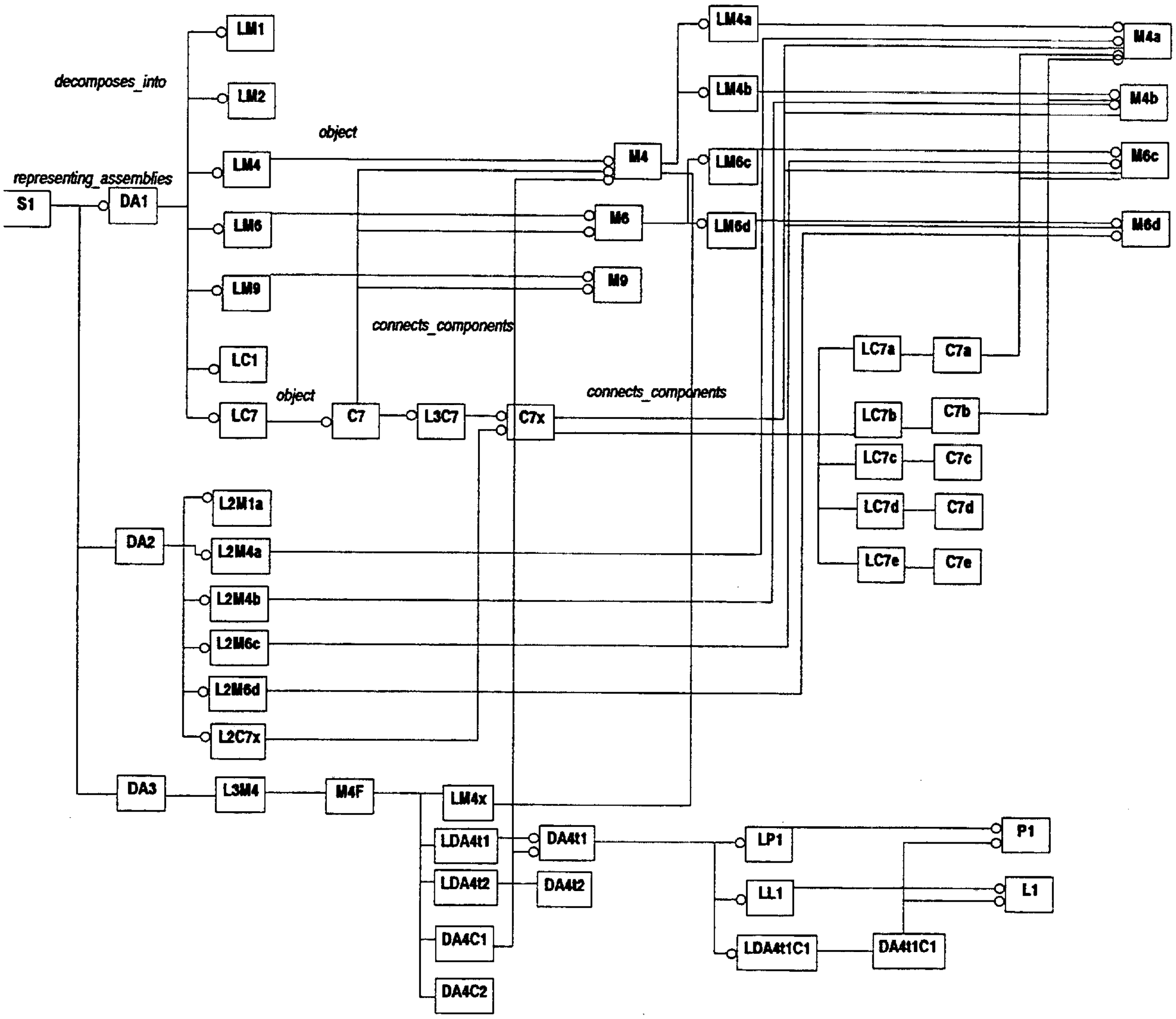

Figure 41. Description of the structure for manufacturing purposes

- represent portions of the LRFD Specification using AP230 as a data model.

While performing these tasks, we discovered that the authors of AP230 had developed a very large and extensive product and process model for the exchange of computer–interpretable information relating to structural steel building frames. Although we were impressed with the amount of the work they had undertaken, we found the some problems with the current version of AP230, which are described in Section 3. Briefly, these problems are:

- The documentation of AP230 is not well organized, contains editorial errors, is incomplete and difficult to browse.

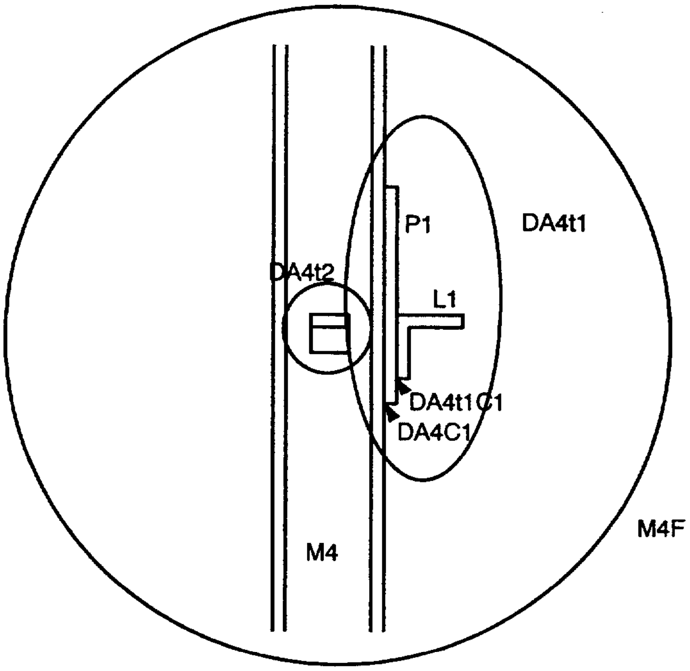

Figure 42. Column M4F in manufacturing stage

- The naming convention used in the current version of AP230 is internally inconsistent in many places.

- The application reference model mainly supports overall structural design and contains little information on other activities including conceptual design.

- The application activity model is much more general than the activities covered in the application reference model.

- We found that the activities that are covered by the application reference model are arbitrarily selected and do not form a single consecutive subset of the activities described in the application activity model.

- We were unable to find a logical explanation why some of the outputs of the activities in the application activity model appear as controls to the consecutive activities, but not as inputs.

- The Units of Functionality are not very informative and do not convey the content of the information that will be carried in these units.

- The application objects listed under Units of Functionality and conformance requirements sections are, in most cases, incomplete and inconsistent with the description of the functionalities and conformance classes.

- The application reference model is only available in EXPRESS–G and not in (ASCII) EXPRESS.

- The application reference model contains classes that can be used to describe three logical views of the same structure: *design_assembly, manufact_asbly* and *analy-*

sis_group. However, the explicit links between the classes in these views do not exist.

- The immediate subconcepts of the top level concept, *steel_structural_frame_entity,* are arbitrarily chosen and the selection is very inconsistent and misleading.
- US Practice (based on LRFD) and AP230 use different terminology to express the same entities and attributes.
- AP230 assumes an order for the design process (e.g., AP230 assumes that member selection will be done before connections are designed,). Due to this assumption, some information is not possible to be established or expressed (e.g., net area or effective net area during the member selection activity using AP230).
- AP230 does not support built–up members, composite design and earthquake loads.
- Even though most of the information required for evaluating a design against LRFD can be expressed using AP230, deducing some of the information is a very difficult task.
- Because AP230 can only be used to exchange information regarding the geometry of members, but not their functionality, it is impossible to check if a member satisfies some provisions of LRFD.
- AP230 does not support design evolution. In AP230, different entities are used to represent different views of the same structure at different phases of the design.
- AP230 does not support multiple views of the same component.
- AP230 only allows decomposition hierarchies and does not allow composition hierarchies.
- AP230 only allows decomposition for assemblies and not for members and connections.
- AP230 provides limited support for the representation of the conceptual view of a structure.
- AP230 provides limited support for the representation of members and connections with missing information.
- AP230 does not support expression of derivable information.
- Virtual segments (segments that occur when there is no physical division of a member into sub–members) are not allowed in AP230.

We decided to extend AP230 to solve these problems. To be able to extend AP230, we first developed an alternative data model to the one used in AP230. Later, we integrated this model into the data model used in AP230.

As described in Section 4., the alternative data model was developed without concern about integrating it with AP230. We believe the new model is an improvement to AP230 because it supports:

- evolutionary design information;
- multiple views of the same entities;
- a conceptual view of the structure;
- missing information;
- derivable information;
- composition;

- functional decomposition;
- strong connectivity between three views (analysis, design, and manufacturing) of the structure;
- decomposition of connections;
- decomposition of members;
- built–up members; and
- virtual segments.

However, because this model is not complete and completing it would require a major effort, we decided to integrate this model with AP230. The integrated model retains some of the advantages of the alternative model and uses most of the entities that are defined in AP230.

7. Closure

AP230 is a very large and extensive product and process model. Overall it is a very good start and for the most part, it is applicable to US practice. However, some major and minor problems must be corrected before it can be successfully used in the US as a standard for the representation and exchange of electronic data relating to structural steel framed buildings. The model we developed is a starting point for correcting these problems.

8. Acknowledgements

This research is based on work supported by the National Institute of Standards and Technology (Computer Integrated Construction Program of Building Fire Research Laboratory) Project No. 060NANB6D0143.

NIST-114
(REV. 11-94)
ADMAN 4.09

U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY

MANUSCRIPT REVIEW AND APPROVAL

(ERB USE ONLY)	
ERB CONTROL NUMBER	DIVISION
PUBLICATION REPORT NUMBER NIST-GCR-97-738	CATEGORY CODE
PUBLICATION DATE September 1997	NUMBER PRINTED PAGES

INSTRUCTIONS: ATTACH ORIGINAL OF THIS FORM TO ONE (1) COPY OF MANUSCRIPT AND SEND TO THE SECRETARY, APPROPRIATE EDITORIAL REVIEW BOARD.

TITLE AND SUBTITLE (CITE IN FULL)

GCR-97-738: Evaluation of the AP230 Application Protocol

CONTRACT OR GRANT NUMBER
60NANB6D0143

TYPE OF REPORT AND/OR PERIOD COVERED

AUTHOR(S) (LAST NAME, FIRST INITIAL, SECOND INITIAL)
Garrett, J.H. Jr.; Fenves, S.J.; Kilicotte, H.

PERFORMING ORGANIZATION (CHECK (X) ONE BLOCK)
☒ NIST/GAITHERSBURG
□ NIST/BOULDER
□ JILA/BOULDER

LABORATORY AND DIVISION NAMES (FIRST NIST AUTHOR ONLY)

SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (STREET, CITY, STATE, ZIP)
U.S. Department of Commerce
National Institute of Standards and Technology
Gaithersburg, MD 20899

PROPOSED FOR NIST PUBLICATION
□ JOURNAL OF RESEARCH (NIST JRES)
□ J. PHYS. & CHEM. REF. DATA (JPCRD)
□ HANDBOOK (NIST HB)
□ SPECIAL PUBLICATION (NIST SP)
□ TECHNICAL NOTE (NIST TN)
□ MONOGRAPH (NIST MN)
□ NATL. STD. REF. DATA SERIES (NIST NSRDS)
□ FEDERAL INF. PROCESS. STDS. (NIST FIPS)
□ LIST OF PUBLICATIONS (NIST LP)
□ NIST INTERAGENCY/INTERNAL REPORT (NISTIR)
□ LETTER CIRCULAR
□ BUILDING SCIENCE SERIES
□ PRODUCT STANDARDS
☒ OTHER NIST-GCR

PROPOSED FOR NON-NIST PUBLICATION (CITE FULLY) □ U.S. □ FOREIGN

PUBLISHING MEDIUM
☒ PAPER □ CD-ROM
□ DISKETTE (SPECIFY)
□ OTHER (SPECIFY)

SUPPLEMENTARY NOTES

ABSTRACT (A 2000-CHARACTER OR LESS FACTUAL SUMMARY OF MOST SIGNIFICANT INFORMATION. IF DOCUMENT INCLUDES A SIGNIFICANT BIBLIOGRAPHY OR LITERATURE SURVEY, CITE IT HERE. SPELL OUT ACRONYMS ON FIRST REFERENCE.) (CONTINUE ON SEPARATE PAGE, IF NECESSARY.)

AP230 is a proposed international standard for the representation and exchange of electronic data relating to structural steel framed buildings. It is still currently under development and not yet accepted as an international standard.

Our major objectives in doing this research are to evaluate the AP230 Application Protocol with respect to US practice and extend it for use in early conceptual stages of the facility design process.

In this report, we present the problems we identified in AP230 and propose extensions to solve these problems.

This report is organized as follows: an overview of AP230 is presented, problems we have identified in AP230, alternate model to solve the problems, how new model can be integrated with AP230 and summary of this research.

KEY WORDS (MAXIMUM OF 9; 28 CHARACTERS AND SPACES EACH; SEPARATE WITH SEMICOLONS; ALPHABETIC ORDER; CAPITALIZE ONLY PROPER NAMES)

application protocol; building frame; information models; ISO 10303; STEP; structural design; and structural steel

AVAILABILITY
☒ UNLIMITED □ FOR OFFICIAL DISTRIBUTION - DO NOT RELEASE TO NTIS
□ ORDER FROM SUPERINTENDENT OF DOCUMENTS, U.S. GPO, WASHINGTON, DC 20402
☒ ORDER FROM NTIS, SPRINGFIELD, VA 22161

NOTE TO AUTHOR(S): IF YOU DO NOT WISH THIS MANUSCRIPT ANNOUNCED BEFORE PUBLICATION, PLEASE CHECK HERE. □