User and Functional Requirements for 3D Parametric Modeling of Cast-in-place Reinforced Concrete Structures

November 19, 2007

3D Parametric Modeling of Cast-In-Place Reinforced Concrete Structures

Reinforced Concrete BIM Consortium

User and Functional Requirements for 3D Parametric Modeling of Cast-in-place Reinforced Concrete Structures

A DRAFT Specification

R. Sacks, C.M. Eastman, R. Barak and Y.-S. Jeong

On behalf of the members of the

Reinforced Concrete BIM Consortium[1]

November 19, 2007

[1] Accu-Crete, Atomic Energy of Canada Limited, Barton Malow, Bechtel, Georgia Institute of Technology, Ghafari, Grand River Construction, M. A. Mortenson, SOM, Technion, Tekla, Thornton Tomasetti Group, Walter P. Moore

Acknowledgement

Numerous members of the RC BIM consortium drafted significant portions of content for this report, including contributions of both text and figures. Although their companies are listed as formal consortium members, the authors specifically acknowledge the personal contributions of the following representatives:

Christopher Brown
Peter Carrato (consortium chair)
Ben Cheplak
Andy Dickey
Alexander Ivanov
John Manuel
David McLean
Rick Ricciuti
Tim Webster

We also acknowledge the generous support of Tekla Inc. in funding this effort, and in particular the leadership role played by Stacy Scopano in initiating the consortium.

Table of Contents

List of Tables

List of Figures

1. Introduction and Overview

1.1. Overview

The consortium’s primary objective is to formulate a specification of user-based functional requirements for effectively and efficiently modeling and detailing cast-in-place (CIP) reinforced concrete (RC) structures in a parametric 3D computer model.

This specification describes functional requirements for satisfying end-users’ requirements such as the parametric behaviors, shape and property editing, specialized functional operations, numbering and quantity take-off requirements, and other aspects that are peculiar to reinforced concrete. The functional requirements are derived through three stages (see Figure 1).

The first stage has been to compile a process model to describe the work activity and information flows in design and construction of CIP RC structures. The process model served as a check-list to ensure that the user requirements of all aspects of the process were considered. It also defined the scope of the process to be considered. The next stage has involved bringing together a “list of issues” regarding the specific engineering needs of CIP RC projects. This list was then translated into functional requirements regarding the parametric tool in order to support the various design, analysis, engineering and construction management activities of all actors throughout the process. As a final stage, the functional requirements of parametric 3D modeling tools would be implemented to provide the CIP industry, designers and contractors alike, with the ability to model its structures from the early stage of conceptual design through detailing and up to and including construction and maintenance.

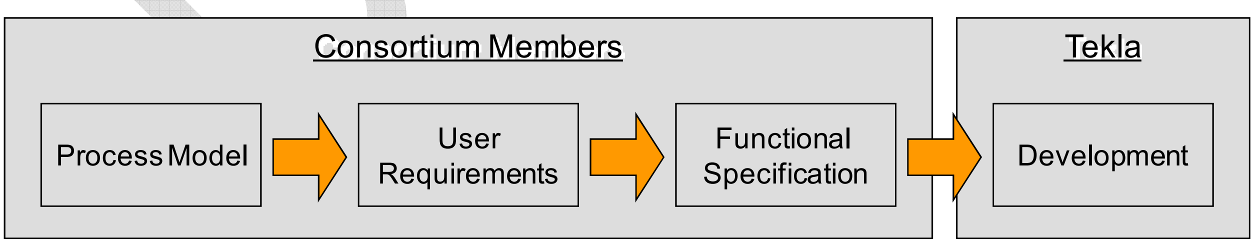


Figure 1. Development processes of BIM tool for CIP RC structures

1.2. Members of Consortium

This research was performed through a collaborative effort among four groups: North American cast-in-place concrete contractors and construction management services, structural engineers, building product modeling experts, and software developers.

- Concrete Contractors and Construction Managers
  Accu-Crete – Chrisotpher Sansone
  Atomic Energy of Canada Limited – Dejan Dikic, Medhat Elgohary, Alexander Ivanov, Azhar Khan, Rick Ricciuti
  Barton Malow – Ronald Sinopoli, Tim Webster
  Bechtel – Peter Carrato, John Manuel, Khaldoon Sakkal
  Grand River Construction – Matthew Ray
  M. A. Mortenson – Ken Bracken, Derek Cunz, Greg Knutson, David Schroeder

- Structural Engineers
  Ghafari – Samir Emdanat
  SOM – Christopher Brown
  Thornton Tomasetti Group – David McLean
  Walter P. Moore – Ben Cheplak

- Building Product Modeling Experts
  Georgia Institute of Technology – Chuck Eastman, Yeon-Suk Jeong, Donghoon Yang
  Technion – Israel Institute of Technology – Rafael Sacks and Ronen Barak

- Software Developers

Tekla Inc. – Andy Dickey, Hans Ehrnrooth, Michael Gustafson, Heikki Haikonen, Petri Heinonen, Ari Kohonen, Elja Kontturi, Charles Pool, Ville Rousu, Stacy Scopano, Jukka Suomi, Rhett Thomson, Laura Virros, Alistair Wells, Ragnar Wessman,

Tekes – Reijo Kangas

2. Process Model and Scope

The consortium compiled a process model to describe the work and information flows prevalent in reinforced concrete construction. The model affords a clear and definitive medium of communication between the many participating members, allowing rigorous definition of the aspects of the process from early conceptual design through construction and maintenance.

The model also served as a check-list to ensure that the user requirements of all aspects of the process were considered. It also identifies the information exchanges needed with outside software or actors. The process covers the activities and information needs of all the professional groups concerned with reinforced concrete structures. It also defines the boundaries of the scope of RC engineering, for the purpose of this project.

The model was compiled using the GT-PPM modeling tool (Lee, 2007). The models were first compiled for three specific sub-tasks - engineering, contracting and interface, on the basis of a draft proposed by the researchers, and then the three sub-models were compiled into a comprehensive model.

2.1. Construction Project of Cast-in-place RC Structures

The engineering requirements for a CIP RC structures were derived through modeling the process - activities and information flows - prevalent in cast-in-place reinforced concrete construction. The process models were prepared using the GT-PPM modeling tool. The model covered the stages of engineering, contracting, pre-construction, installation and construction management, as shown in **Figure 2**. Each stage was classified into more detailed sub-processes. This model especially handled eight sub-processes corresponding to engineering fields – “structural concept and layout”, “structural analysis & design”, “construction planning”, “detailing”, “production planning”, “fabrication”, “form, rebar, pour and strip”, “production control”.

Note that no ‘actors’ are defined in the process maps. Naturally, many professions and organizations are participants in the process - design engineers, contractors, suppliers, fabricators, and others. Many of them can be performed by professionals belonging to different organizations, depending on the contractual arrangements of any particular project. For example, a detailer may belong to the engineer of record’s office, to an engineering office hired by a contractor, or even to a rebar fabricator’s organization. In each case, the basic information flows and requirements are the same.

2.2. Structural Concept and Layout

Structural concept and layout is based on the conceptual design of the facility as a whole and of other building systems. Developing a structural concept needs fundamental data such as seismic data, building code, operation and maintenance requirements. Structural layout is determined by structural concept and other system layout. Since structural layout includes determination of structural types, materials, applied design method, location and size of structural members, feasibility analysis and stability analysis can be already performed at the conceptual design stage (see Figure 3). In order to analyze feasibility of CIP RC construction projects, cost, material availability and expected construction schedules should be considered. If structural stability and feasibility are not satisfied, feedback for rebuilding the structural concept or structural layout is required. The results that are generated from this process are transferred to the structural analysis and design stage for construction of CIP RC structures.

2.3. Structural Analysis & Design

The structural analysis and design process is a core process within the overall engineering process for cast-in-place reinforced concrete structures. User interfaces for performing this process require various kinds of information from different sources. As shown in Figure 4, design codes, standard details and information about the other building systems are all necessary.

In order to perform preliminary design, engineers must decide on the concrete type, set structural member size and identify structural connections, determine load transfers on the basis of the structural concept and layout (which is determined in the previous process). Also, engineers should consider external environmental conditions such as soil, drainage, and other site conditions. Structural analysis can be performed using loads and load combinations defined based on the results of preliminary design

2.4. Preconstruction and Construction Detailing

2.4.1. Construction Planning

Construction planning begins with consideration of the construction methods Figure 4. Although in reality at this stage engineers might consider alternative structural methods, the scope of this discussion is limited to cast-in-place concrete. Thus the range of alternatives considered will extend primarily to various formwork alternatives, as well as to the equipment used for pouring the concrete. Decisions may also be made as to the type of concrete and the methods of curing it. Numerous methods may be compared through each of the steps detailed in Figure 5. After review of the construction method, detailed planning commences. This includes a loop of refinement that extends to cost estimating of the proposed approach, providing feedback to inform steps to improve the plan. Once a construction plan is finalized, scheduling of the production steps can be undertaken. The schedule determines the master plan for execution of engineering detailing and production planning, as well as the basis for overall project control.

2.4.2. Detailing (Construction Drawings)

Detailing provides the fine-grained engineering information needed for fabrication of the rebars and embeds, for designing and planning the formwork, and for actual fabrication on site of the finished products. It is commonly performed in the cascading steps of setting final dimensions for the concrete members, detailing the reinforcement bars and meshes and finally detailing the embeds, as shown in Figure 6. As soon as the overall geometry has been proposed, the design is checked for clashes or interference with members of other building systems, such as HVAC ducts, electrical and water conduits. Where possible, these are resolved through dimensional change, by moving parts of one or both interfering systems, or by approving penetration of the other system's parts (such as a pipe) into the concrete volume. Once the full detail has been set with candidate values, the internal parts are also checked for clashes or interference and for constructability, and clashes are remedied by going back and adjusting the details until all clashes are considered resolved. The same steps must be performed to ensure that there are no clashes between internal parts of the concrete and parts of any other systems that are indeed embedded in the concrete. Another aspect of detailing is final definition of the surface finishes that are desired on the exterior of the members.

2.5. Installation

2.5.1. Production Planning

The production planning process shown in Figure 7 specifies detailed activities of the pre-construction stage. Production tasks for CIP RC structures are based on approved design information that is defined through preceding processes. Production planning defines work packages comprised of formwork, rebar work, and embed sets and pours. Quantities of materials for purchasing, generated through these tasks, are linked with and transferred to the fabrication process. In production planning, engineers determine pour size, delivery systems, pour batching, labor requirements, and schedule. Since the work packages are dependent on a variety of requirements, production models for CIP RC structures should support flexible definition.

2.5.2. Fabrication

Fabrication refers to the final processes that must be completed before execution of a CIP work package at the construction site; fabrication of the formwork system, fabrication of the reinforcing steel, fabrication of any embeds required that are not off-the-shelf items, and finally procurement of all standard items. All of these must then be delivered to the site (see Figure 8). In the case of the formwork system, fabrication can refer to actual manufacture of special forms made of steel or other materials for a specific project, or adjustment of existing modular formwork systems to suit project specific geometry requirements. For structures that have repetitive shapes, this may be required only once for each set of shapes (typically, for each floor). Given that formwork is essentially the negative form of the final concrete shapes, functionality to derive formwork surfaces from concrete member surfaces is highly desirable.

2.5.3. Form, Rebar, Pour and Strip

Building the formwork, tying the rebar, pouring the concrete and stripping the forms are the actual production tasks performed on site (Figure 9). They begin with marking out the geometry required, a task commonly done by a surveyor. Once the basic form is prepared (underside for a slab, or one side of a wall or other vertical member), rebar and MEP systems are installed. Vertical members must then have their formwork closed and tied together (the locations of such ties are often a design feature of importance to aesthetics of the members, and so it must be possible to represent them explicitly in a building model at their designed locations).

As a production process, quality control is commonly performed. Two stages of inspection are common: once all assembly tasks have been completed and before pouring commences, and then again after the forms are stripped. Given that concrete requires time to cure and harden, once horizontal members are stripped they must be shored and sometimes also covered to avoid moisture loss during curing.

2.5.4. Production Control

The final stage covered in the process model is production control (see Figure 10). This stage is concerned with measuring what is done for purposes of accounting, production control, production planning and to collect data to support estimating and production planning in the future.

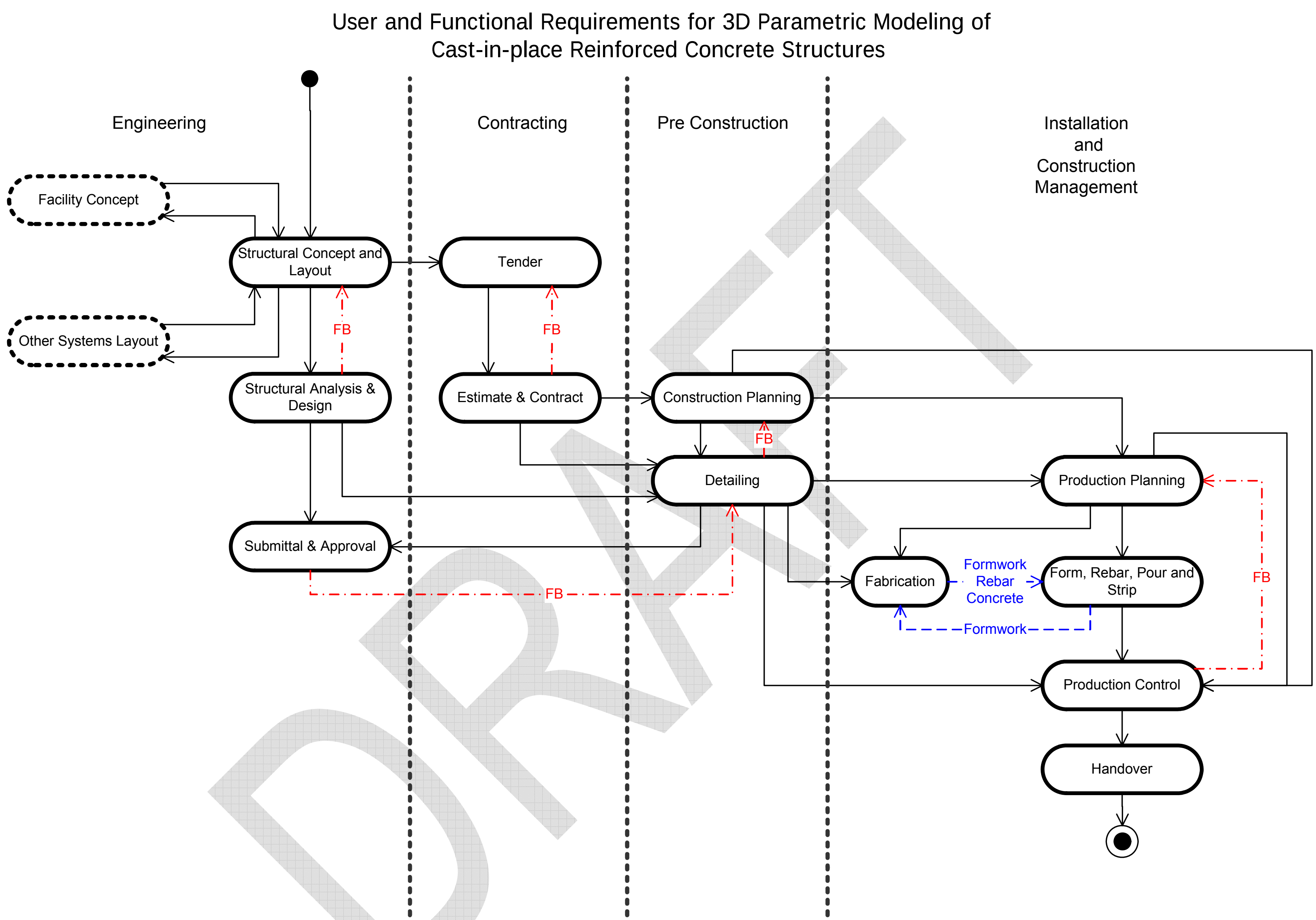


Figure 2. Construction project of cast-in-place RC structures

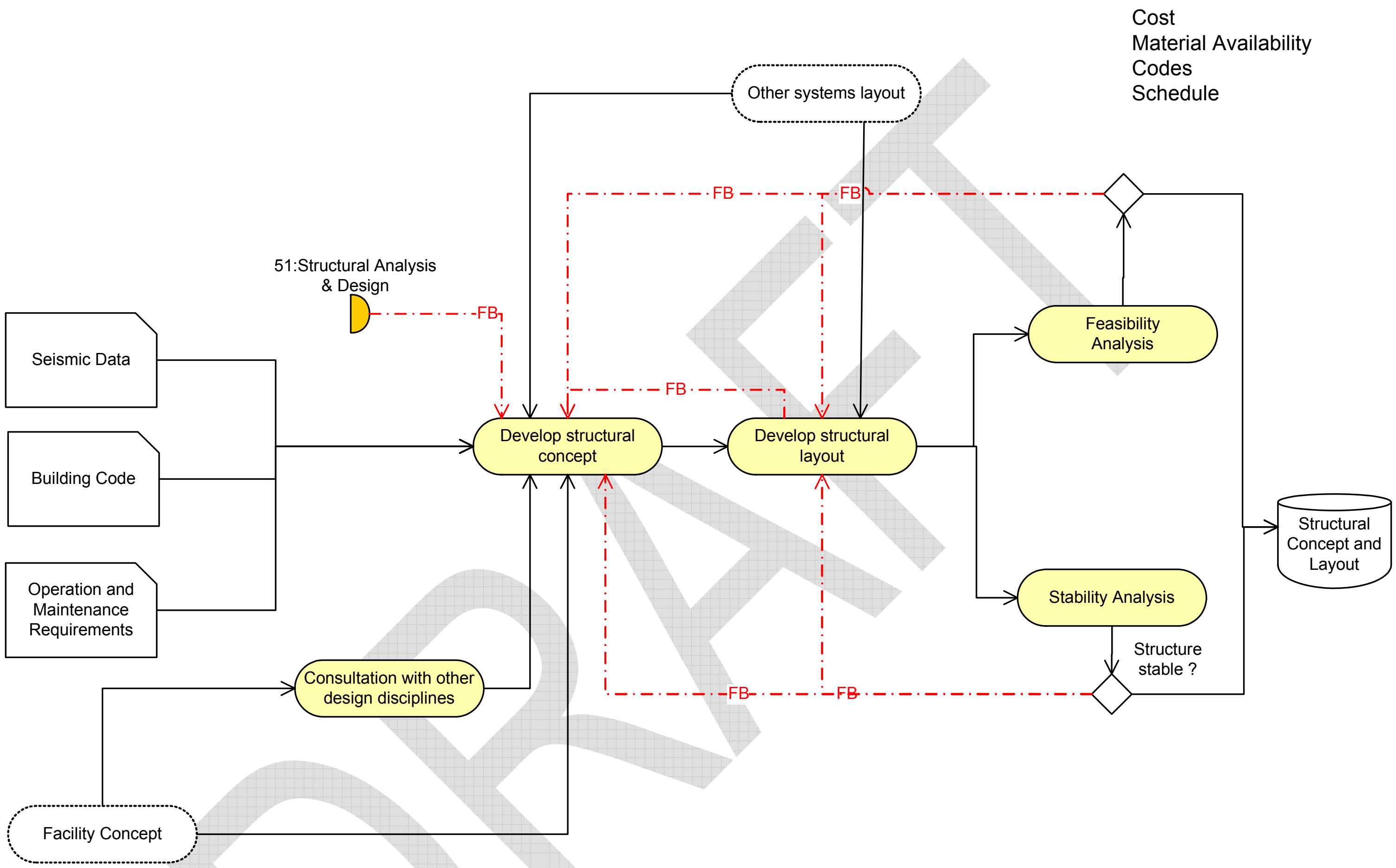


Figure 3. Structural concept and layout

User and Functional Requirements for 3D Parametric Modeling of
Cast-in-place Reinforced Concrete Structures

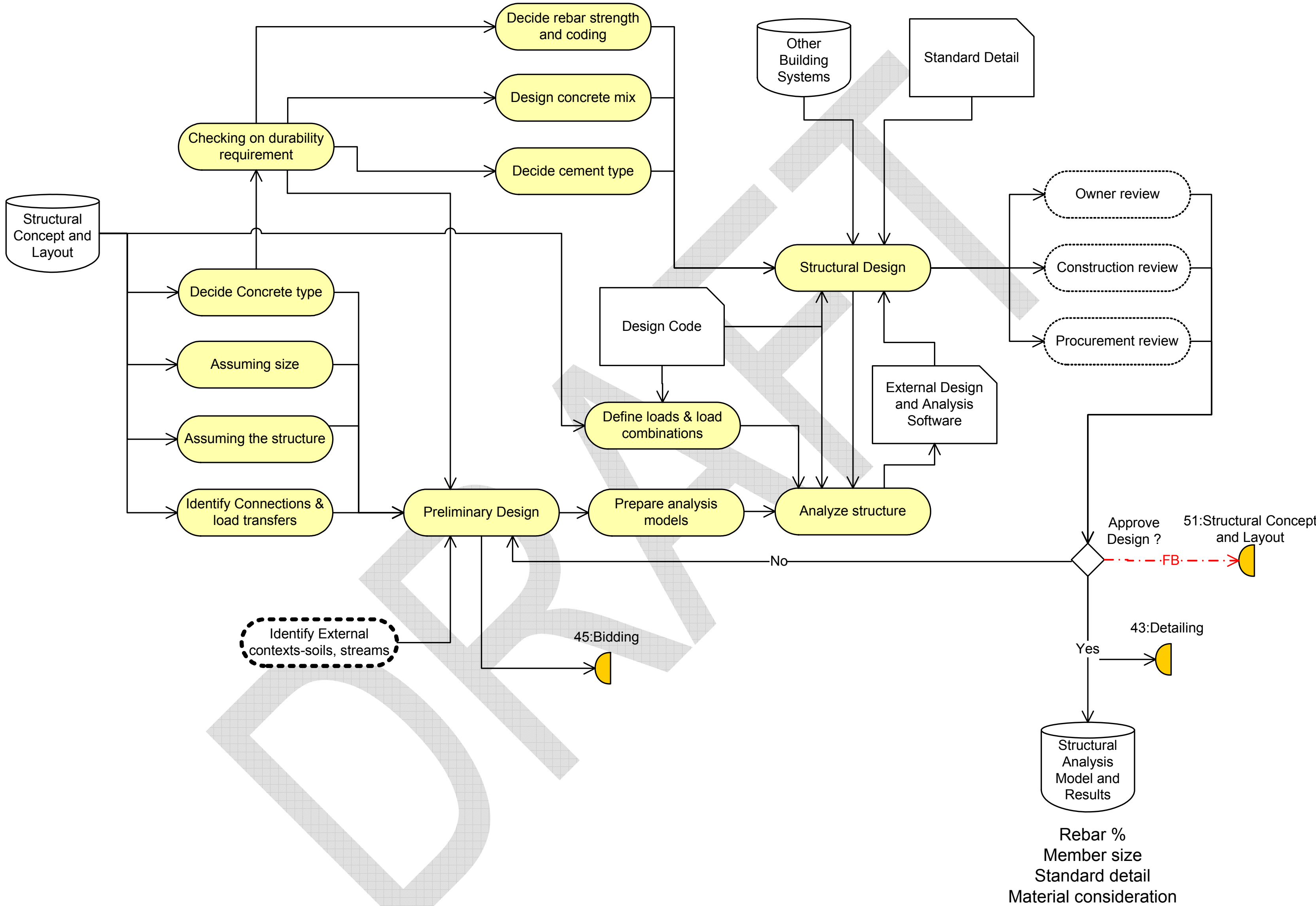


Figure 4. Structural analysis and design

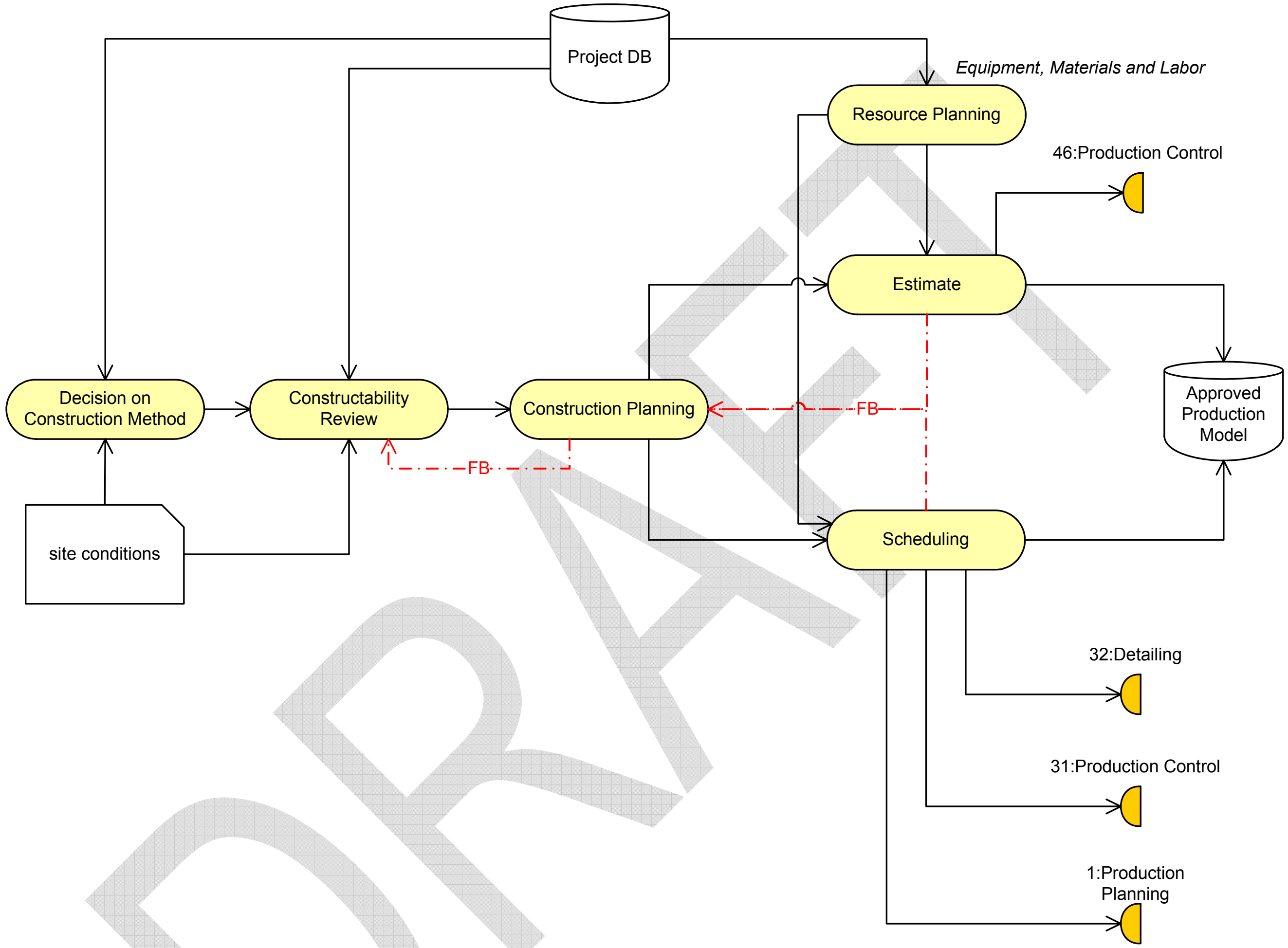


Figure 5. Construction planning

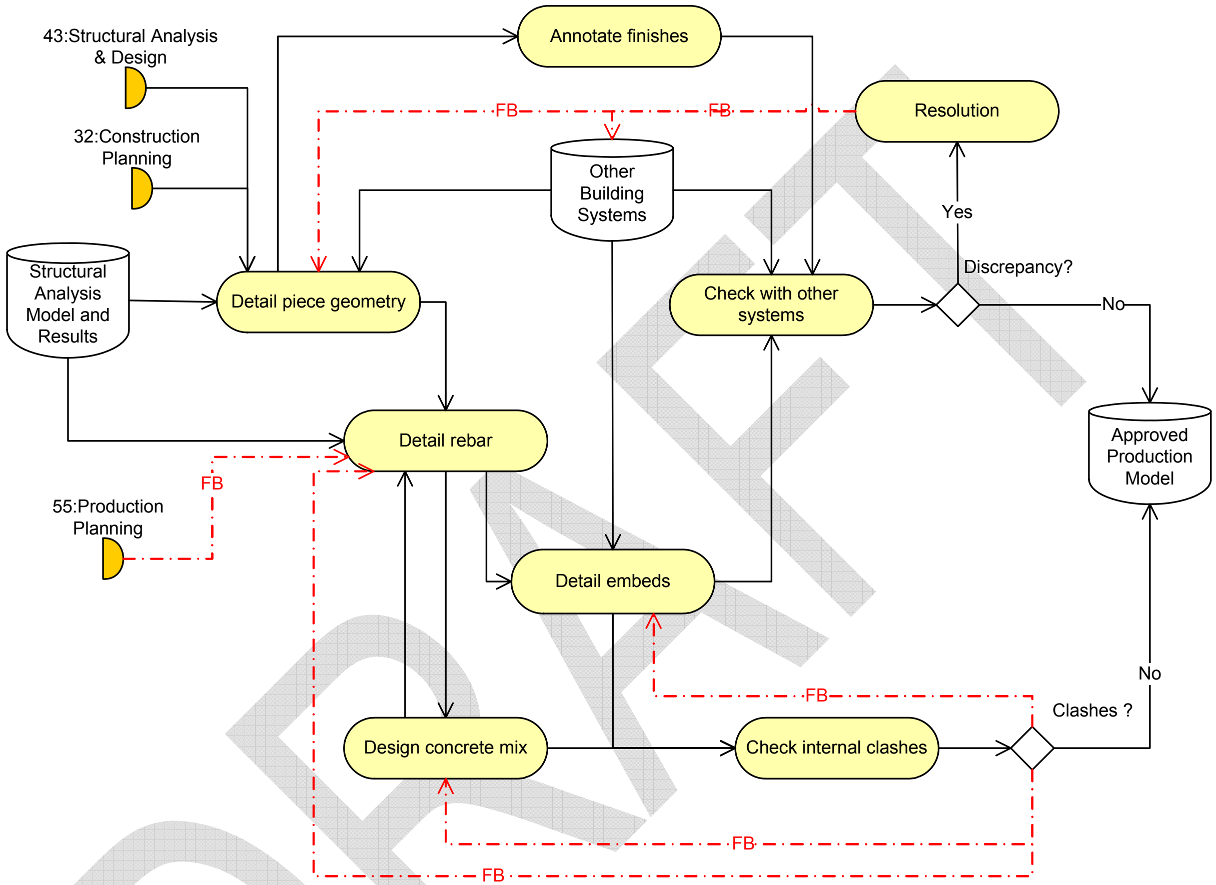


Figure 6. Detailing (construction drawings)

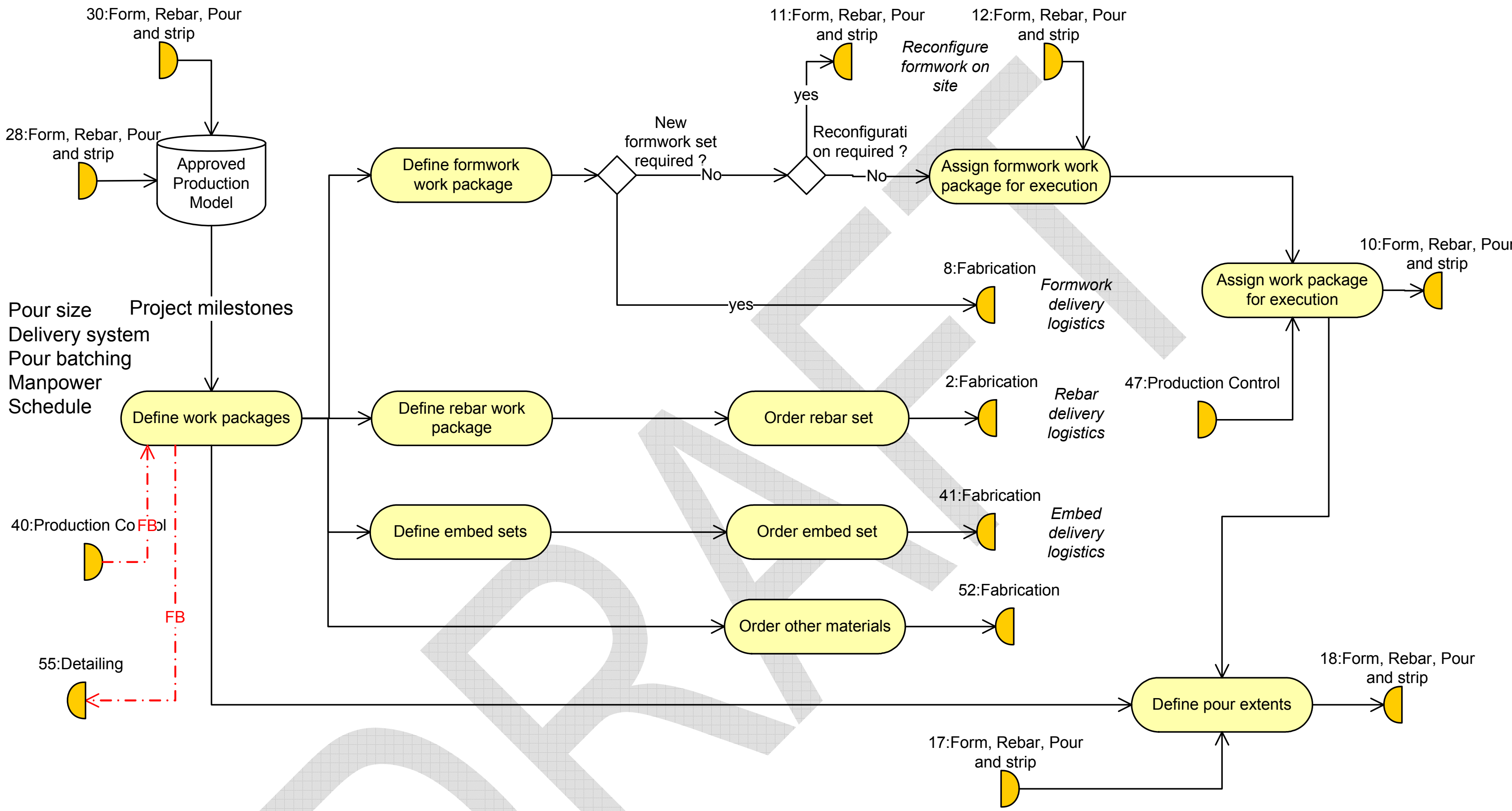


Figure 7. Production planning

User and Functional Requirements for 3D Parametric Modeling of Cast-in-place Reinforced Concrete Structures

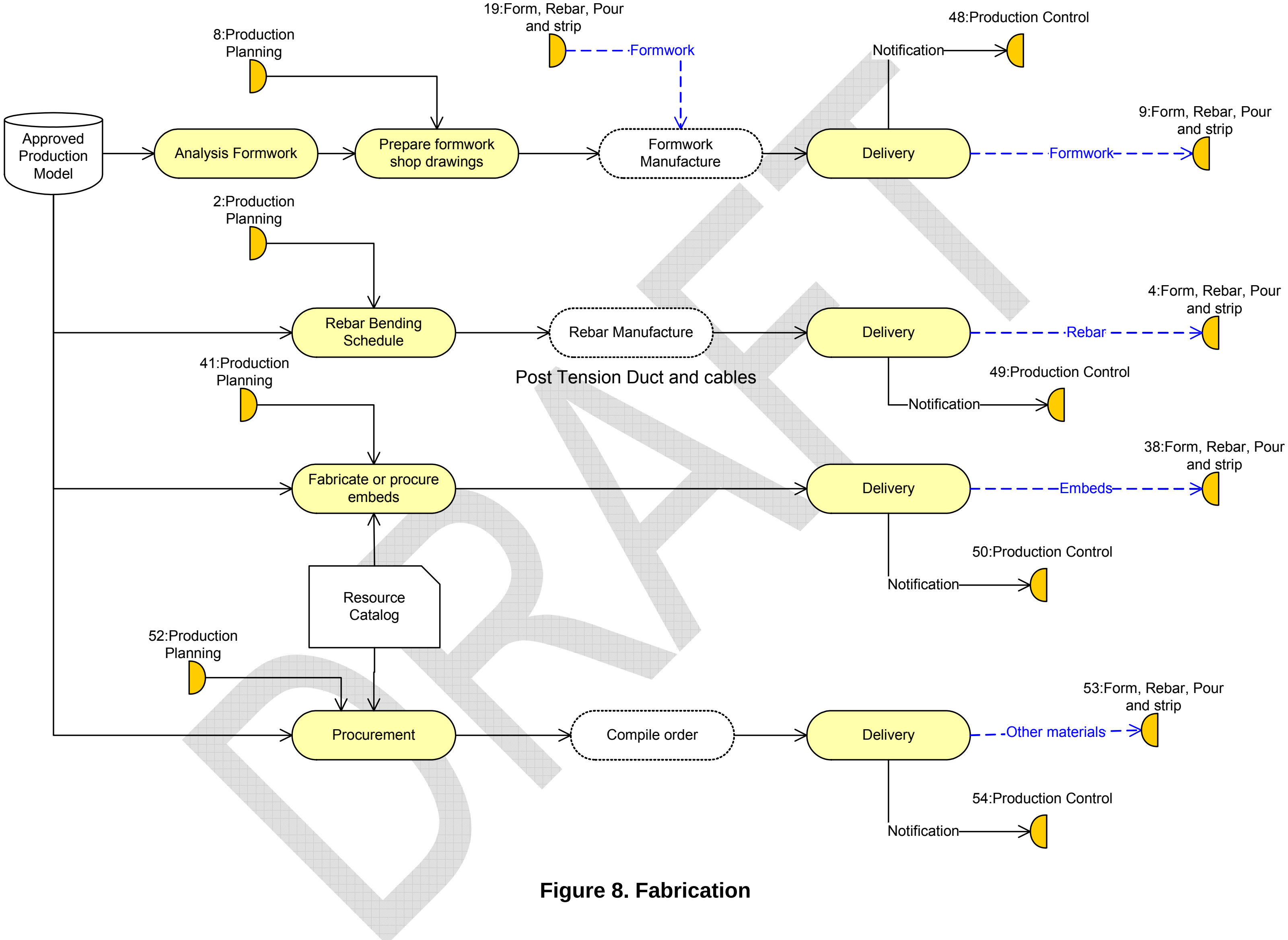


Figure 8. Fabrication

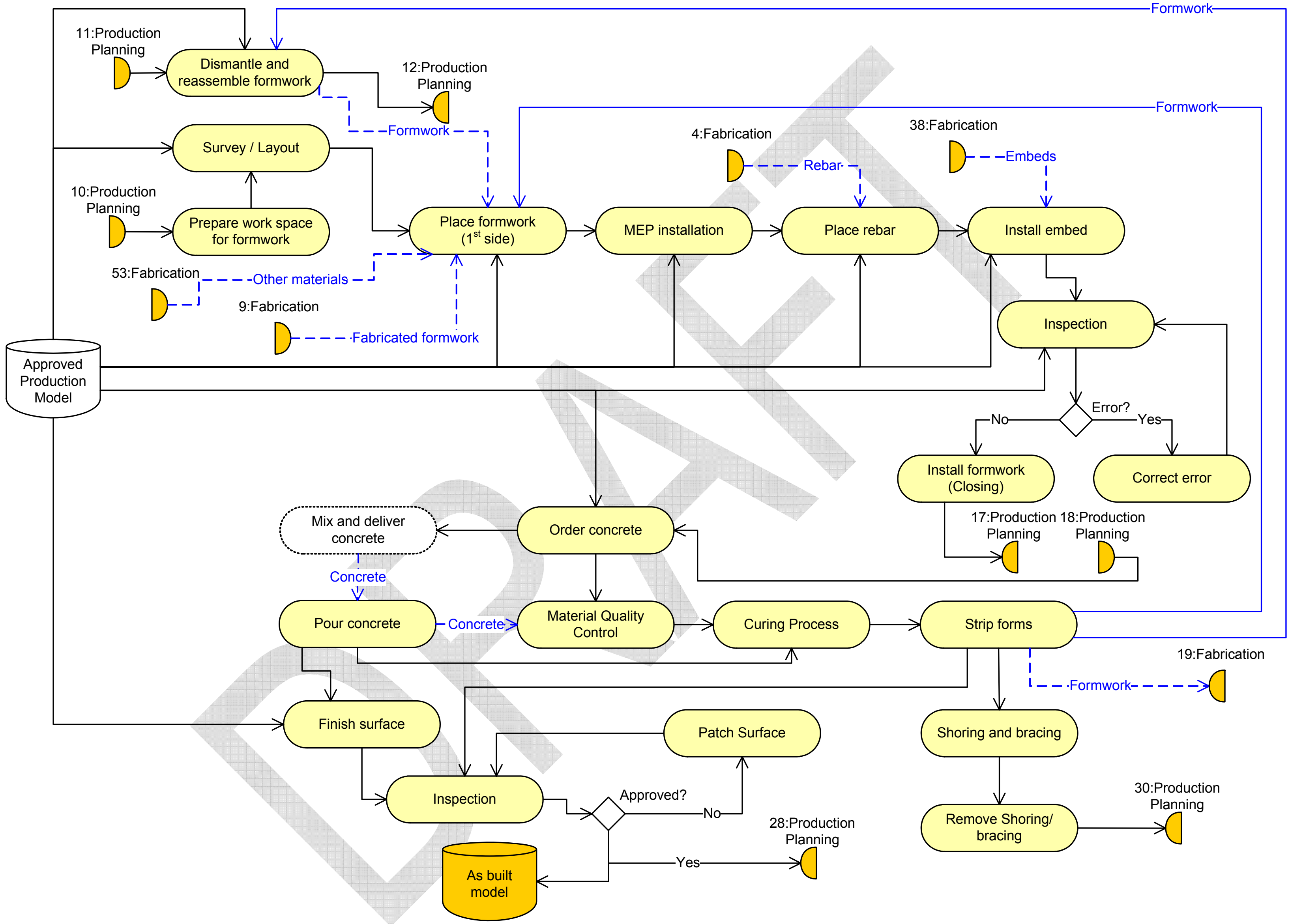


Figure 9. Form, rebar, pour and strip

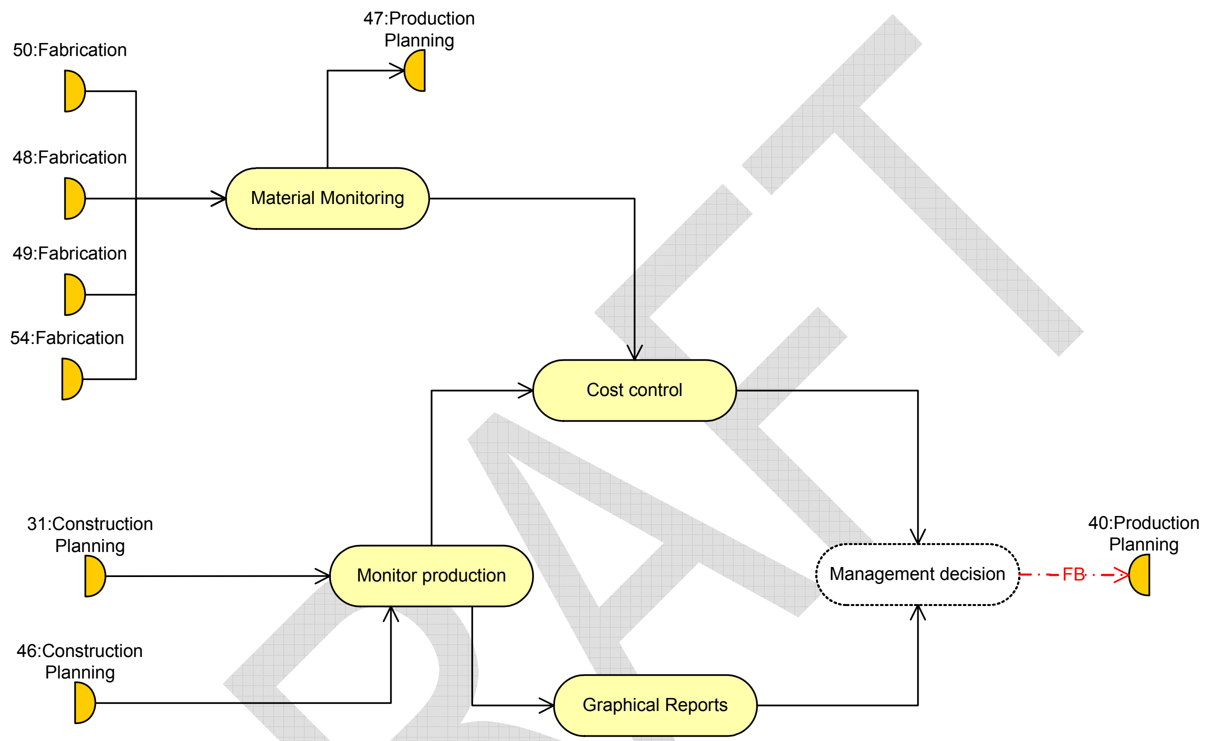


Figure 10. Production control

3. User Requirements

3.1. Overview

This chapter defines the end-users' requirements in order to support the various design, analysis, engineering and construction management activities of all actors throughout the process of cast-in-place reinforced concrete construction. The issues raised highlight the needs of the CIP concrete industry that go beyond the functionality currently provided in software; requirements that are already clearly provided have not been repeated. The requirements presented and detailed here provide the raw material for extraction of the functional requirements that are outlined in chapter 4 below.

After the brief introduction to each sub-section, a table is provided to give an overview of the different items included in that sub-section in relation to the process stage for which they are relevant. The headings in the tables refer directly to the major stages delineated in the sub-section headings of Chapter 2.

3.2. Modeling CIP Structures

3.2.1. CIP Objects

General	• Support structural members with arbitrary geometric shape
Conceptual Design	• Handle beam, column, slab, panel, wall
Engineering	-
Contracting	-
Pre Const./Detailing	• Multi-story column, waffle slab
Installation	-

An RC shape is defined from multiple overlapping shapes. Structural shapes themselves may be highly overlapping, corresponding to the multiple structural elements that TS is to recognize (for analysis and reinforcing). In addition, for decorative or other reasons, additional sub-shapes may be added to the structural elements to define the final form.

Columns may be multi-storey, with intermediate connections, with constant, tapered, or stepped cross sections (see Figure 11). For flat slabs, drop panels may be added at columns locations shown in Figure 12. Special capitals, such as those in Figure 13, are usually created with special formwork. Thus a fixed geometrical shape should be able to be added to a column to define a capital. (These are classified as part of the structural shape, of course).

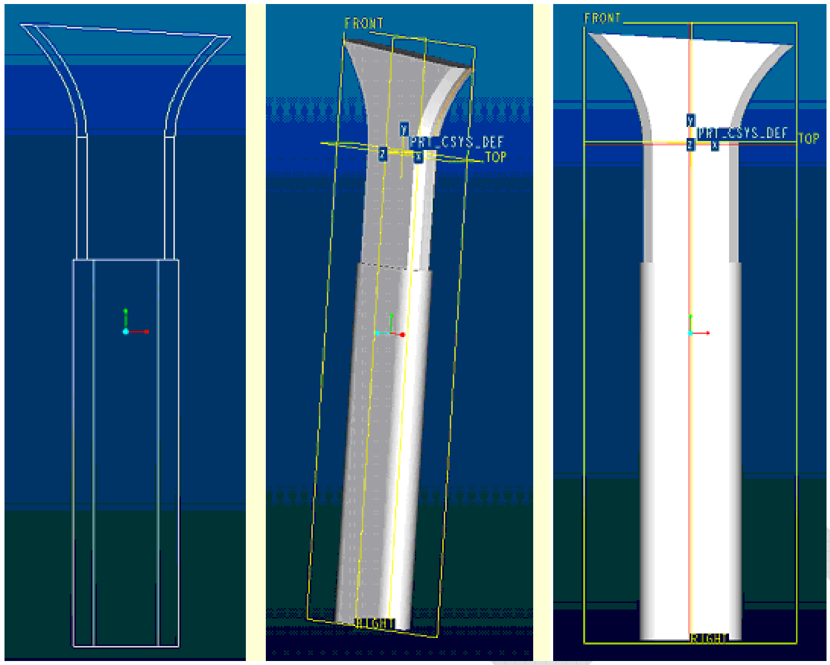


Figure 11. Multi-storey column with intermediate connections

Beams may be multi-span. However, they may be controlled by intermediate nodes serving as control points.

Slabs may be large monolithic elements, defined by a sequence of points (with line interpolation) or with a connected set of edges, possibly including curved segments. A slab may have multiple holes, defined in the same way. Slabs often have edges with drops. See Figure 14. Special slabs should include one way (ribs) or two ways (waffle) joist system slab. See Figure 15.

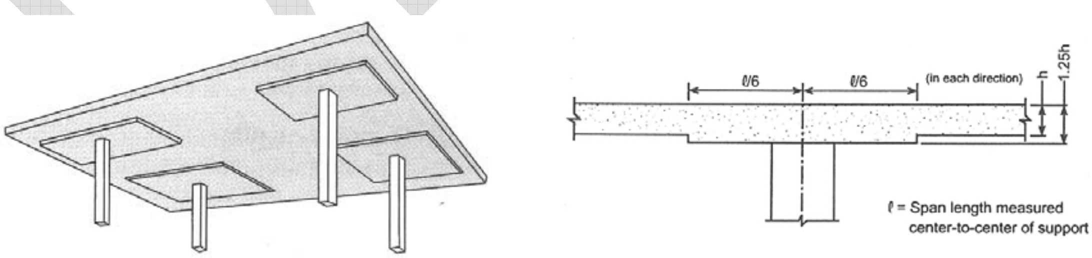


Figure 12. Flat slab with drop panels (Fanella 2000)

Figure 13. Column Capital

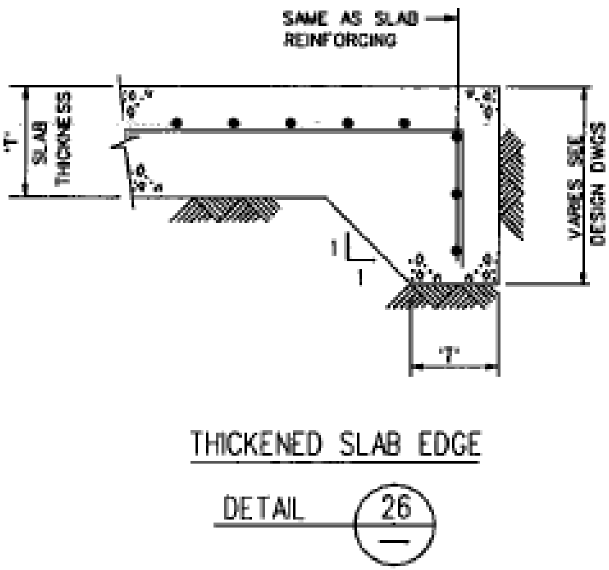
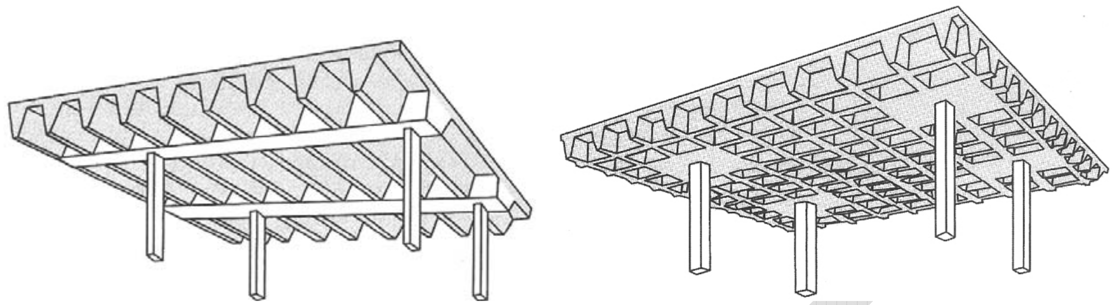


Figure 14. Slab edge with drops (Example by Bechtel)



(a) Ribs Slab **(b) Waffle Slab**

Figure 15. Various slab systems (Fanella 2000)

Walls/Panels also may be multi-span, with intermediate intersecting columns or walls.

In the above cases, elements should extend to the terminal parameter line or point used to define them. They define the extent of the column, beam, wall or slab for analytic purposes. (Connections are not limited to their overlapping region but rather typically overlap further the parts they connect.)

Thus each type of shape is to be flagged as to its function: structural column, structural beam, slab, wall, non-structural, etc. These part classes will be used later for determining reinforcing layout.

3.2.2. Modeling intersecting concrete volumes

General	• Manage intersection concrete volume
Conceptual Design	-
Engineering	-
Contracting	-
Pre Const./Detailing	• Support for connections with external systems • Insertion of embeds and pass-through
Installation	-

Reinforced concrete is unique in that there are no physical boundaries between structural components. The concrete material is continuous where components are joined to one another, so that delineation between components is conceptual and not physical[2]. However, in a 3D-based object model, the structures are nevertheless modeled using distinct objects that represent components that are defined by their functions such columns, beams etc. As a result, objects are commonly placed such that they have overlapping volumes.

For example, the section of a waffle slab shown in Figure 16 is composed of four model objects: the slab itself, a tee cross-section beam that protrudes below the slab, an inverted tee beam that protrudes above the slab section, and blocks of filler material that create the waffle ribs in two directions. As modeled, all of the latter three objects have overlapping volumes with the slab object.

A similar, common situation occurs where concrete walls have columns along their lengths, usually with the outer face of each column aligned with the outer face of the wall, and the columns are thicker than the wall. Part of the volume of the column overlaps the volume of the wall.

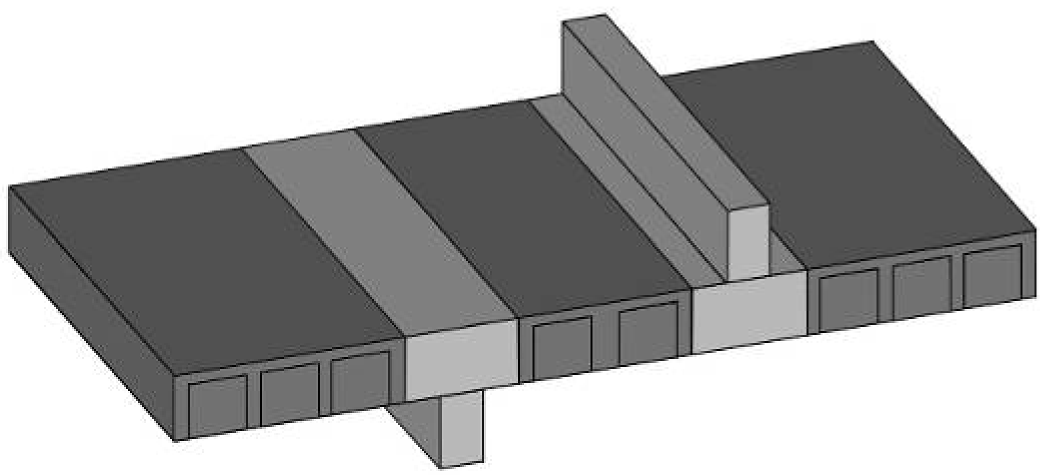


Figure 16. 3D-based Waffle Slab

[2] For the purpose of this discussion, we ignore pour stops between successive pours of concrete. Their positions are often unrelated to the conceptual boundaries of actual components, and thus they are not relevant in the majority of cases contemplated here.

Another example is the case where two beams intersect at right angles, as shown in Figure 17. The volume of concrete at the intersection of the two beams is equally a part of each of the two beams.

Most 3D solid modeling software, including Tekla Structures, can resolve overlapping volumes using Boolean operations. The rules that govern resolution of intersecting volumes – i.e. the way in which components cut one another – are specific to particular domains and are applied (whether explicitly during modeling or implicitly by preprogrammed library connections) according to the modeling intent of the user. The result in each case is a model that expresses those rules.

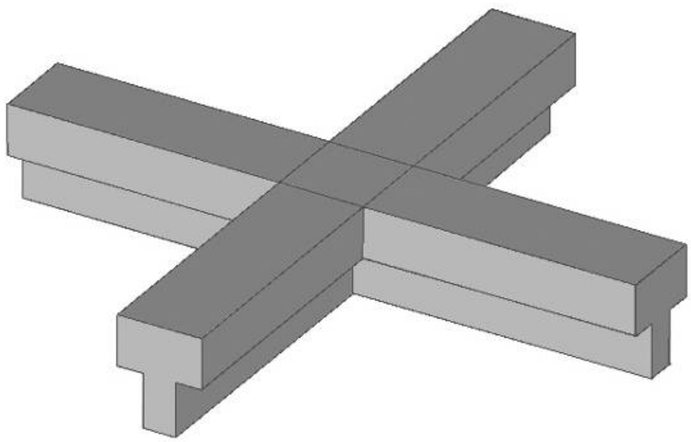


Figure 17. Two beams intersect at right angle

However, in the case of CIP concrete, there are different functional needs from the model, each of which requires that the volumes be resolved in different ways (according to different sets of rules). Some of these are as follows:

1) **Detailing reinforcing**

The model with the overlapping volumes serves for detailing reinforcement. As shown in Figure 18, rebars commonly overlap and extend from one component into another to create structural continuity – from the point of view of reinforcement. Overlapping concrete volumes are both necessary and desirable.

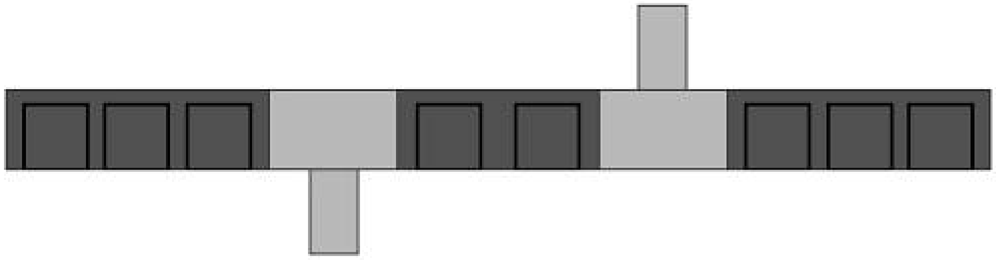


Figure 18. Definition for detailing reinforcing

2) Preparing bills of quantities

Bills of quantities are prepared for estimating and for construction contracts. They typically contain different line items that are priced according to the resources that must be invested in order to produce a unit of the item. In the waffle slab example of Figure 19, the line items might be Table 1.

Figure 19. Definition for bill of quantities

Table 1. Items for bill of quantities

Item ID	Description	Unit of measurement
1	Waffle slab thickness 20"	Sq.ft
2	Upper beam 10"/20"	L. ft.
3	Lower beam 10"/20"	L. ft.

The beams that protrude above and below the slab, although having the same dimensions, are different for pricing and therefore are separated into two lines. On the other hand, there are cases where the basic component cross-sections are different from one, but after resolution of the issue of overlapping concrete, they "become" the same, and will therefore appear combined into a single line in the bill of quantities.

The fact that the slab has embeds and the beams are actually T beams is not relevant for this representation. What used to be a slab with lightweight block filler is now one part with a new name "Waffle slab", and the T beam is now a rectangular beam with different dimensions. This means that by resolving the overlapping concrete we have created new entities with different attributes.

There is no one definitive way to define object shapes for compiling bills of quantities, and practices may differ between states, owners and local company requirements.

Note that in this and the next two examples, rebar should not be affected in any way by the resolution of the overlapping concrete volumes.

3) Preparing bills of material for procurement of materials during installation

In this example, the lightweight block filler materials or other embeds (other than rebar) that are included within the concrete must be measured for procurement, and their volume must be subtracted from the concrete volume with which they overlap[3] (see Figure 20).

[3] Note that although the filler material is not concrete, it is also considered to be a case of modeling with overlapping volume with the basic concrete slab. It would be cumbersome, from the viewpoint of computational complexity, to actually subtract the volumes of all the blocks from the volume of the slab, and providing a separate functional view obviates the need to generate such a complex physical model.

Figure 20. Definition for bill of material

The rebar cages for the tee beams extend through the volume of the slab as well as that of the beams. Therefore, if a user wished to order all of the parts needed for casting the slab, the rebar cages of both beams would have to be included. This aspect is treated in detail in the section dealing with pour breaks and pour stops below (see the Section 3.6.1).

4) Formwork design

One part of formwork design is sequencing. In this case the slab and the lower beam is cast together as one piece and the upper beam will be cast after the slab concrete has hardened. As shown in **Figure 21**, the overlapping concrete volume for the lower beam was resolved by performing a union operation between the tee beam and the slab, creating one single piece to be cast using one formwork set up. There are alternative ways to specify pours, for example using pour IDs. Union of the geometry may not be required.

Figure 21. Definition for formwork design

5) Drawing

Drawings should not represent overlapping volumes or separate adjacent elements. The concrete part is shown as one 'cast unit'. The filler blocks are drawn to show the different material and the fact that there is no concrete in those places. In general, all occurrences of overlapping concrete columns are resolved by uniting all of the concrete elements into one object with a single contour line, either geometrically or in display and drawing (See Figure 22).

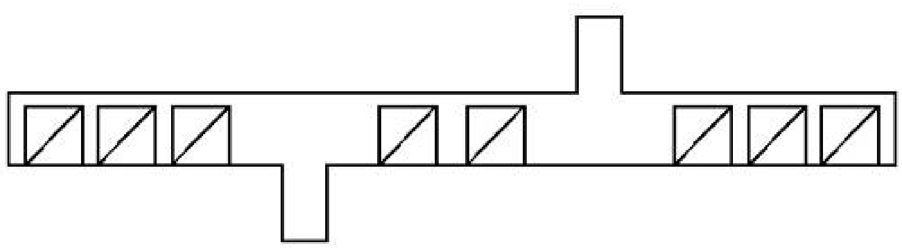


Figure 22. Cross-section drawing of the slab

These requirements can be classified as belonging to three types of representation:

1. **Physical Model**: This model represents the actual components profile and dimensions. The overlapping concrete volumes are not resolved, allowing the user to apply the proper detailing and reinforcement. The physical model is the 'parent' or 'root' model, and as such is the definitive source for the geometry for all of the functional models and the drawings.

2. **Functional Models**: These models are needed for a variety of tasks, including structural analysis, preparing a bill of quantities, extracting a bill of materials for purchasing, and designing formwork. Each of these is based on the physical model, but requires different treatment of the intersecting volumes. Treatment of intersecting volumes also differs within individual functions, according to local practices – for example, the standard method of measuring quantities for concrete members may differ according to the demands of specific clients/owners, or according to the work practices of different construction companies.

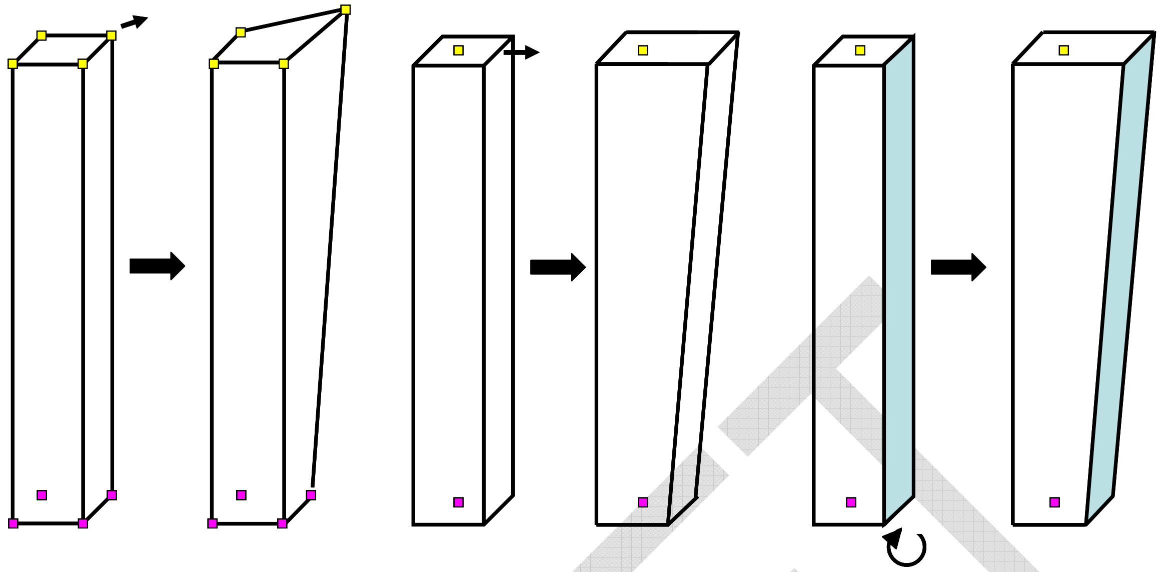
3. **Drawing representation**: In general, reinforced concrete drawings depict the structure as a whole as a single 'cast unit', similar to the physical model. No lines should be drawn around overlapping volumes or between adjacent members in any views.

3.2.3. Manipulation and Behavior Parametric behavior

General	• Minimize model manipulation through project evolution and different design phases • Changing object dimensions on the fly • Manipulating continuous beams and columns
Conceptual Design	• Design intent
Engineering	• Manipulate analysis model
Contracting	-
Pre Const./Detailing	• Dynamic control
Installation	• Track changes between design and actual - Geometry and location

CIP structures can be modeled using a basic set of logical members, such as columns, beams, slabs etc. These are useful points of departure, because they are classified according to structural function. However, the actual geometry required is flexible, and in many cases it is difficult to generate the geometry using purely prismatic extrusions (there are also situations in which geometry may not be prismatic at all, such as decorative features. These are discussed in section 3.2.5, ‘Special Geometries’; this section focuses on the primary structural members). Therefore, the basic logical elements in the building need to be adjusted to fit the overall actual geometry. In order to transform basic logical members, the ability to control and manipulate the geometry of the members is essential.

The 3D objects that represent the actual building shape are defined by points, edges and faces. In theory, the objects can be manipulated in three ways: manipulate points, manipulates edges or manipulate faces. All three manipulations can include translation, and the line and face manipulations can also include rotation. The images in **Figure 23** illustrate three different manipulations.



a) by translating a point b) by moving an edge c) by rotating a face.

Figure 23. Geometry manipulations

When trying to generalize this manipulation, we can see that after the manipulation the shapes have different cross sections at the bottom end and at the top end. The first manipulation, where a point is moved, creates a non-planar face (a warped face). This is fairly rare for any face that requires formwork, but does occur from time to time. **Figure 23**(a) shows an example. Warped faces are more common at the top (unformed) surface of concrete pours, particularly where slopes are required for drainage. Figure 24 shows a typical situation where a point on a slab edge is dropped to create a self-draining surface.

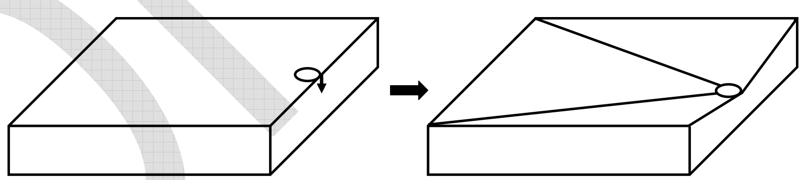


Figure 24. Warped top surface of a slab for drainage.

In the examples in **Figure 23**, all of the changes in the shape apply from bottom to top. There are common cases where change is needed at a singular point. One example of such behavior is that of a column with a capital. Assuming that the column and its capital are one object, we can relate to the change as a manipulation on the original column as shown in Figure 25.

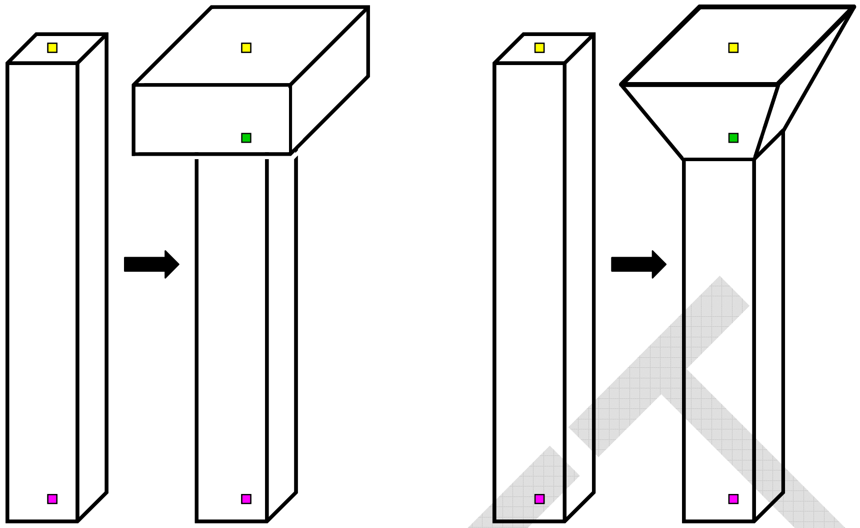


Figure 25. Column with capital.

The right image of Figure 25 is a combination of gradient change and singular change. The column was broken into two parts, with an additional control point introduced where the prismatic shape changed. In the right image, the bottom part has a single cross section, while the top part has different cross sections at its bottom ant its top, with a gradient change between them.

This concept is useful also at a 'meta-level' of members such as columns and beams. Beams are very often placed in continuous sections, along support lines between multiple columns; columns extend through multiple floor heights, with or without changes in their cross-section from floor to floor, or indeed between floors. In both cases, the continuous beams or columns exhibit behavior as whole functional units (e.g. a continuous beam with multiple spans), not only as single members. The association between continuous beams or columns should be maintained beyond the time when they are modeled first, for two reasons:

- Design intent: while there may be later local changes to cross-section or reinforcing, it is often convenient to maintain one 'default master' cross-section and to be able to manipulate the whole continuous beam.

- Analysis: when analyzing continuous beams using static analysis software, the unit of analysis is the continuous beam or column.

- Continuous beams need not necessarily be collinear, but they must have common vertices.

Manipulation of the location of vertices should be possible on the fly. It is natural to move the end of one beam and the start of the next beam using just one same handle. The same is true for column sections below and above a floor – they should use the same single vertical axis, with axis offset values for each segment if needed.

The basic idea behind this functionality is to let the modeler manipulate geometries by moving handle points, lines or planes on the fly. Two main uses will be served directly by this capability:

- Building the model itself will be more efficient and natural. The modeler can achieve the actual geometry naturally, quickly and accurately, while still maintaining the basic functional/logical structural elements in the model.

- Manipulating a design model to reflect 'as-built' geometry can be accomplished simply. CIP structures often have discrepancies between their designed shape and their actual shapes as-built. The reason for this lies in the construction method itself. Using formwork in the field, casting concrete with different equipment causes many small inaccuracies. In some cases, proper detailing of components that have to fit the concrete structure requires accurate as-built geometry. Measurements made on the actual structure need to be applied to the model, and the manipulations described make this possible[4].

In general, at the structural member level, all functional shapes should be defined parametrically, based on cross-sections and control lines/surfaces (in plan, a control element is a line, but it also has height so that it can be snapped to at any elevation.) Profiles are generally defined to be swept along a constructed line or path between grid

[4] One quick way of doing this is to digitize the structure with a laser scanner. The result is a cloud of points in 3D space that can be calibrated with the model coordinate system. To create the “as built” 3D model, small changes must be made to the modeled elements to fit the laser scanned image. To achieve that, the manipulations described are essential.

intersections. For each profile, its placement relation to the constructed path can be controlled, so that for example, a column may have stepped sizes, with offsets allowing the external edges to align (see Figure 27).

At the meta-level (whole structures or large parts of structures), a general set of control surfaces should be defined to manage the basic layout. This will usually be the building grid together with a set of levels, but may also include additional control planes.

Parametric control entities for members and structures should include not only planes (along grid lines or not), but also cylinders and spheres (for domes). Construction lines offset from control surfaces are also required, to manage local features of RC objects.

Figure 26. Example of multiple story columns (Barton Malow).

A different concern is the need to manipulate the analysis model. When irregular geometries are created, the analysis model becomes more and more complicated. To simplify the analysis and to get more coherent results, the structural engineer will manipulate the sticks and nodes, to create a more refined analysis model. Figure 27 shows a situation where a column changes cross-section. The appropriate analysis model is for the top node of the lower column to coincide with the bottom node of the upper column. Unless care is taken when modeling the geometry initially, and through any changes, the result is likely to be a misaligned analysis model. Manipulation using a line movement in this case would make resolution of the problem efficient.

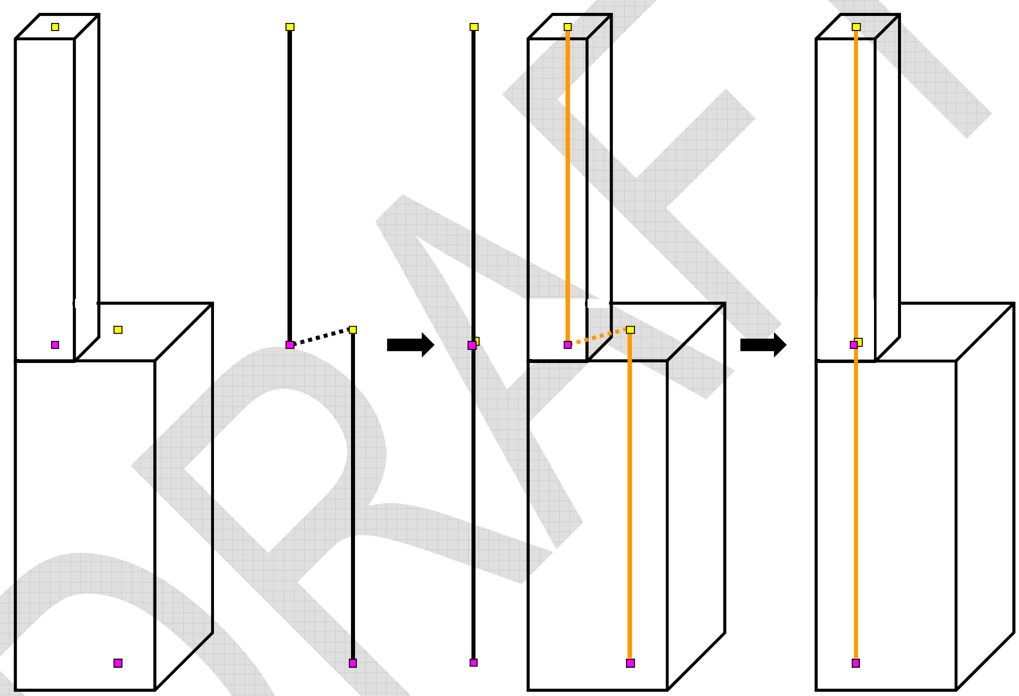


Figure 27. Manipulating the analysis model.

3.2.4. Hard and soft clash checking

General	• Definition and display of tolerances (envelope/shadow) • Part tolerance • Interface with other disciplines, other building systems
Conceptual Design	• Conceptual Clearances
Engineering	-
Contracting	-
Pre Const./Detailing	-
Installation	• Compare with laser scan • Deviation reports

Hard and soft clash checking is an important part of design and constructability review. For most buildings that have a reinforced concrete structures, the structure cost is in the region of 30% of the total project cost. Reducing rework from inconsistency with other domains such as HVAC and electrical systems enhances productivity and reduces construction cost. Checking inconsistency with other building systems is a very important aspect of the system. Inter domain hard clash checking should be provided in the system. Information describing other systems can be provided in the format of 3D DXF, 3D DWG, and other standard formats such as IFC, CIS/2. The system should have interfaces to import data in those formats and compare the geometry to check hard clash.

For most building systems, there are two basic types of clashes possible between objects: a physical overlap is called a “*hard clash*”. A "*soft clash*" occurs when components are too close to one another or too far from one another. These are also termed 'clearance clashes' in some software applications, such as Navisworks Jetstream package.

However, for CIP reinforced concrete, physical interference between objects is often permissible. Not only rebars and post-tensioning cables can be embedded in (and therefore physically clash with) the concrete; other embeds, such as anchors for steel components or pipes running through walls, can also legitimately be designed to ‘clash’ with the concrete material. It is therefore necessary to make a logical designation for every object that determines whether that object is allowed or not allowed to clash with concrete. This can also be implemented with some intelligence about what members should be relocated to avoid a clash. In general, objects in the model that are critical should not be relocated unless absolutely required; other objects in the model that have secondary importance may be adjusted or relocated as required.3

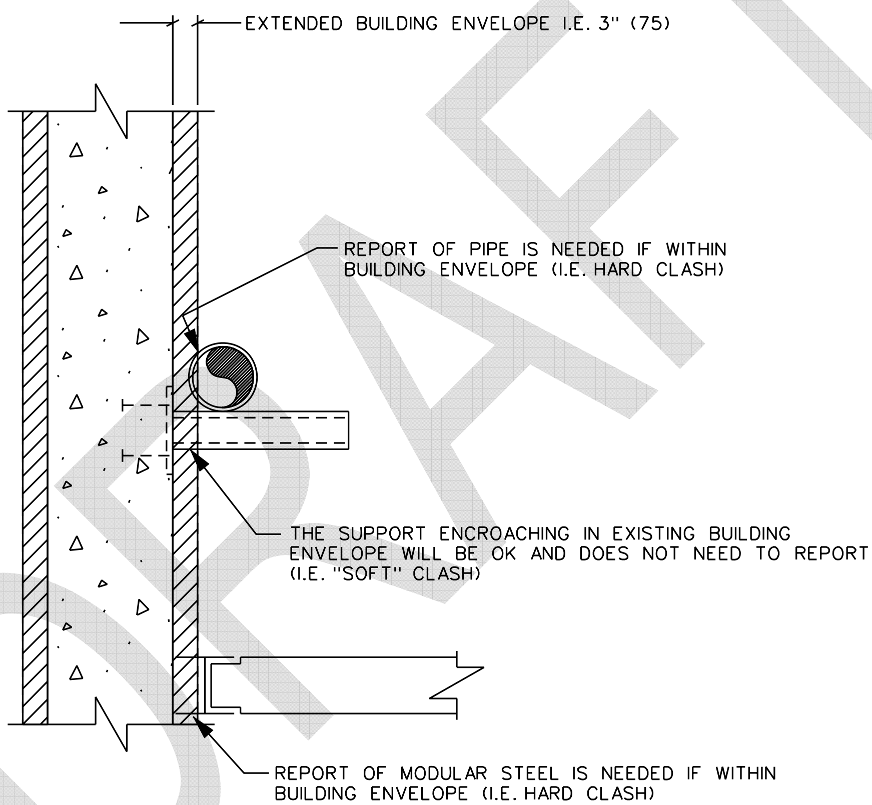


Figure 28. Clash checking for plate to edge

Figure 29 shows some typical situations. The pipe is in a soft clash situation with the concrete wall, because it is closer than the allowed tolerance. In resolving this problem, the pipe should be more readily moved than the concrete wall. However, the pipe support has a hard clash with the concrete wall, yet should not be reported, because this is the desired situation. Thus the pipe would have to be logically annotated as not allowed to clash, while the support is allowed to clash.

The system should have the ability to check for these types of issues on an individual or group basis. Clash checking should not create so many clashes that it would be easier to check clashes manually. This can be achieved through the designation of "weighted" clashes.

To cope with soft (clearance) clashes, software must allow definition of an envelope of clearance around an object. Ideally, this should be a property of the element, specified as a thickness, with an option to make this a “hard” or “soft” envelope. This approach gives the user the flexibility in using the envelope for many different aspects of interdisciplinary coordination such as tolerances of construction, clashing of other systems like formwork, beam deflection, and interference with other objects. It would be also be beneficial if the envelope could be applied to reference objects in a model. Although not absolutely necessary, it would be useful to be able to turn off individual surfaces of the envelope around objects.

Soft clashes should not be limited to certain elements. The system must address soft clashes for items such as post-tensioning cable and embeds as well as concrete and reinforcing steel. The system should allow these clashes to extend outside the structure. For example, there must be adequate room for the jack to be placed on the end of a post-tension cable during stressing. There must also be sufficient room for the ram to extend and for two men to operate the stressing machine.

Soft clashes also related to tolerance issues because the facility can be used to ensure that defined tolerance values will be sufficient to avoid hard clashes in cases of imperfect field work. The “Inter-Industry Working Group” report dated Jan. 14 2006 states this concerning tolerances:

> "No structure is exactly level, plumb, straight and true. Fortunately, such perfection is not necessary. Tolerances are a means to establish permissible variations in dimensions and location, giving both the designer and the contractor parameters within which the work is to be performed.

They are the means by which the designer conveys to the contractor the performance expectations upon which the design is based or the use of the project requires. Such specified tolerances should reflect design assumptions and project needs, being neither overly restrictive nor lenient. Necessary rather than desirability should be the basis of selecting tolerances."

Therefore any BIM software function developed for clash detection must have capability for engineering / contractor group to manually adjust the tolerance requirements to suit specific project requirements (i.e. to specify the envelope distances for soft clash checking).

Numerous examples of both hard and soft clashes are provided in Table 2 and in the figures from Figure 29 to Figure 33. Note that the issues of clash and clearance checking for reinforcement are dealt with in detail in section 3.5.4 below.

Table 2. Example of hard and soft clashes

Hard clashes	Soft clashes
Rebar and post-tensioning cable occupying the same space Plumbing pipe or conduit through column capital Plumbing pipe or conduit penetrating elevated concrete beam where post-tension cable is present.	Not enough room outside the structure to stress post-tension cables Rebar in conflict with electrical conduit but system is unsure if ductile conduit can be adjusted slightly and still meet electrical design intent Reinforcement / embed congestion

Once clashes are identified, the clash detection interface should clearly identify the specific point of the clash with an “x” drawn on the elements associated with change of color of the clashed objects. Both of these visual cues are valuable to pin point the clash and see which members are associated with the clash. There needs to be a method of storing the clash “view” for quick reference as well as external communication. 2D or 3D

PDF are recommended for this. A clash management toolset needs to be developed that allow clashes to be identified, stored for later review and resolution with a comment or document trail that supports the agreed upon solution to the problem.

Even if clash checking is provided, the possibility to integrate reinforced concrete structural models with 3rd party products such as NavisWorks Jetstream, etc., should be maintained.

As this is a fairly complex subject, at this time, it may be only practical to provide a basic set of tools to accommodate basic to intermediate clash detection between Tekla Objects and also between Tekla Objects and Reference Objects. The full scope of this is outside the RC BIM area.

Finally, we also note that clash checking can be sued to evaluate constructability by identifying “sequence clashes”. A sequence clash refers to a situation where a part or tool cannot be maneuvered into place at some stage of project execution because of interference with some other object. This check can only be performed by running a stepped 4D analysis of the structure through time, checking at each step whether hard clashes occur between major building parts and their temporary support structures (formwork or scaffolding, for example).

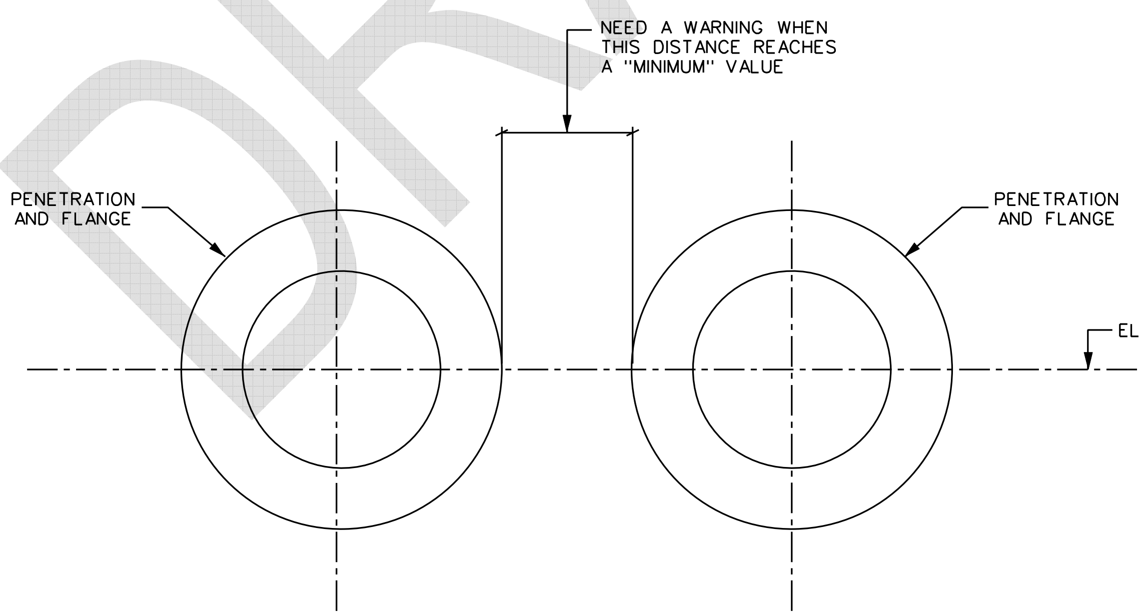


Figure 29. Clash checking for flange to flange

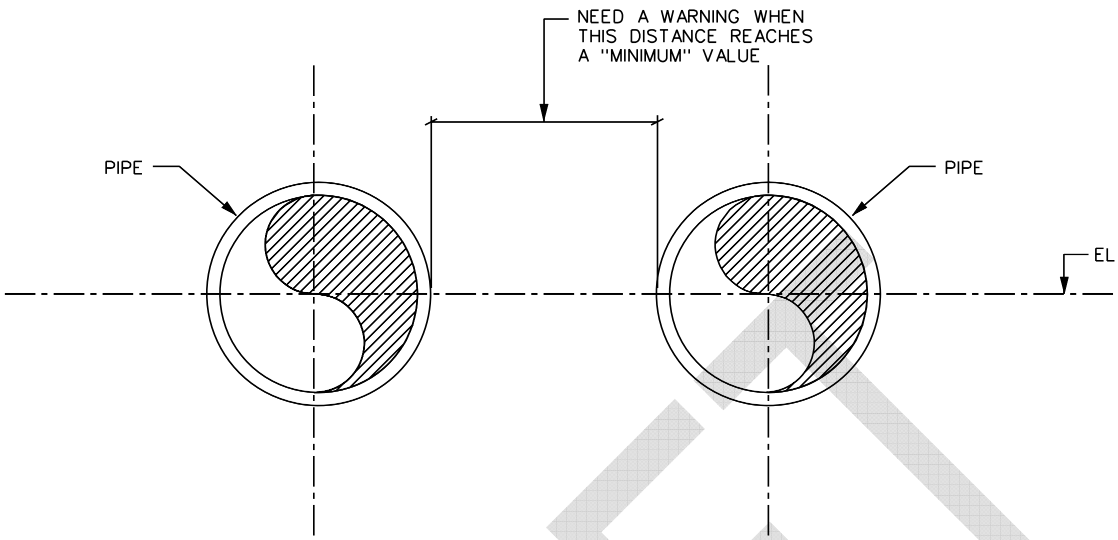


Figure 30. Clash checking for pipe to pipe

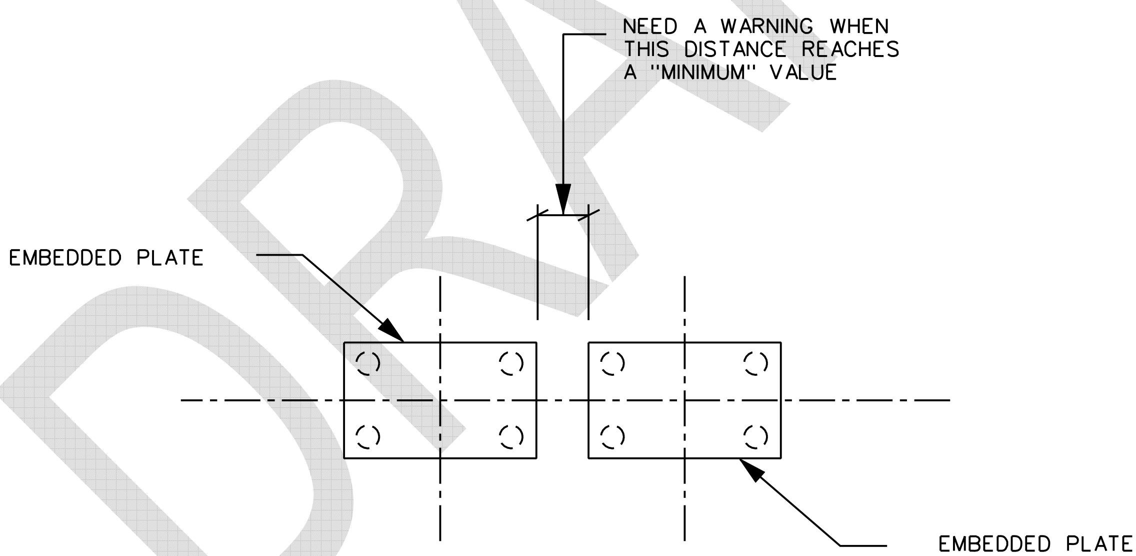


Figure 31. Clash checking for plate to plate

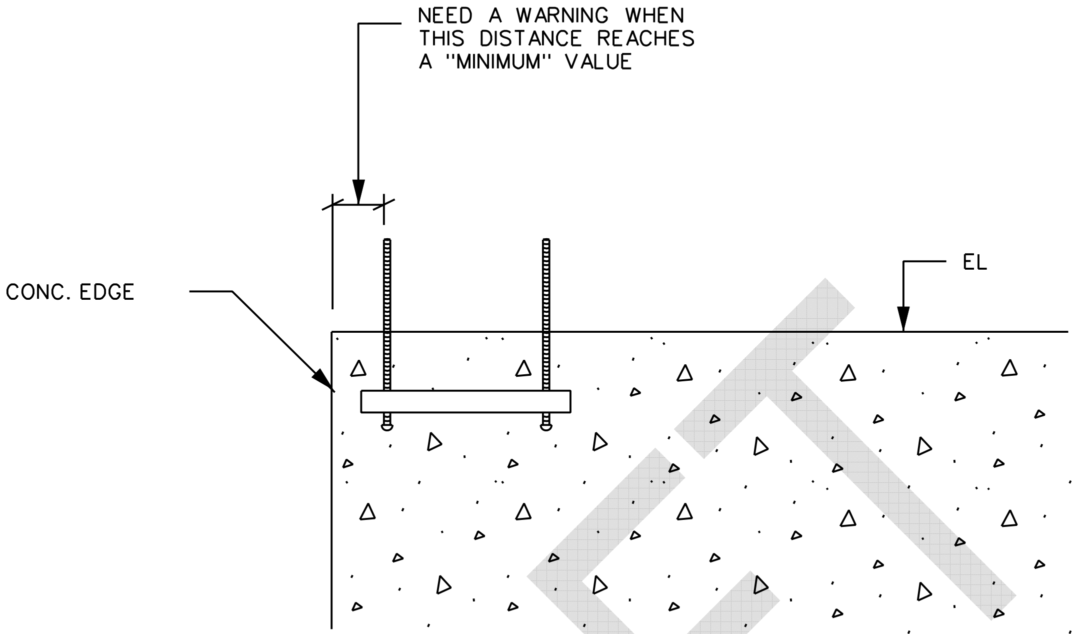


Figure 32. Clash checking for plate to edge

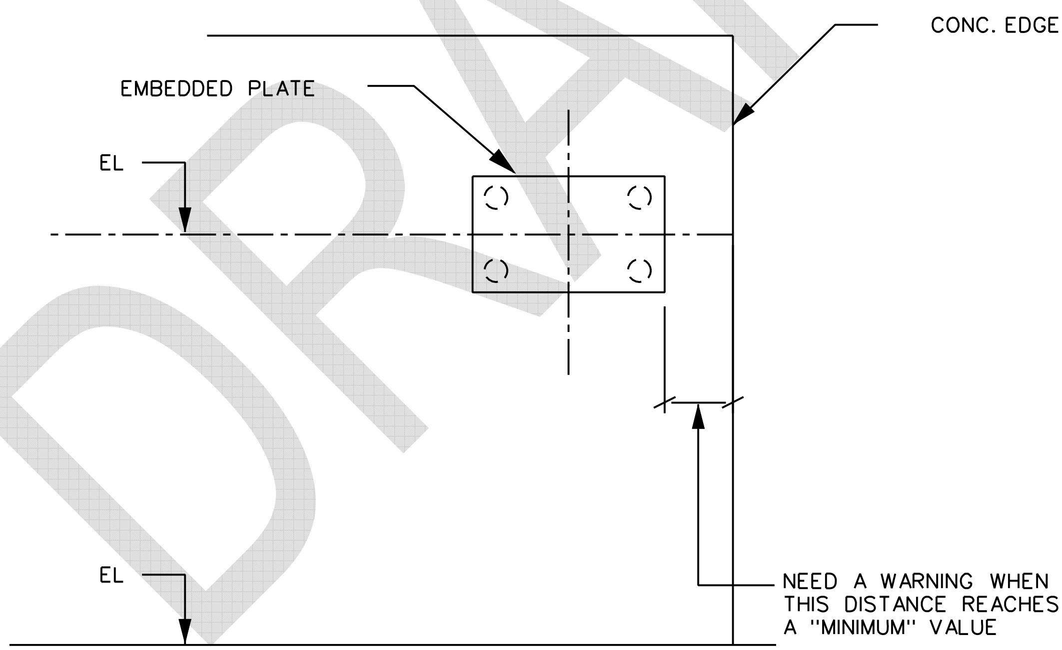


Figure 33. Clash checking for plate to edge

3.2.5. Special geometries

General	• Structure with double-curved surfaces • Sloping • Decorative features
Conceptual Design	• Double-curved surfaces
Engineering	• Modeling of creep
Contracting	-
Pre Const./Detailing	• Sloping of slab surfaces for drainage , trench drains
Installation	-

The following special geometric shapes are using in RC construction:

- Column capital
- Waffle slab
- Sloping slabs (including slope on top only, requires variable thickness slab)
- Slab with depression (e.g., pits and sumps)
- Spiral ramps and stairs
- Belled caissons
- Steel Reinforced Concrete (SRC) column (see Figure 34)

What level of curved surface modeling creation and editing is required, as versus importing existing curved models? An important consideration in developing these capabilities is that it may be easier to import complex surfaces to Tekla after generating them with tools developed specifically for this purpose.

Where functions for generating complex surface geometry are defined, they should fulfill the following guidelines:

- There should be a hierarchy to the geometry definition ? edge, surface, solid
- Edges should be definable by equations ? high order polynomial, and trigonometric functions
- Surfaces should be able to be three dimensional (not just flat)

- Surfaces should be definable by geometric shapes or a series of edges
- Surfaces should be able to be extruded or rotated to form solids
- Axes for extrusion should be definable by equations (similar to the definition of edges)
- The user should be able to import volumes or surfaces from other software applications

3.2.6. Steel/concrete composite elements

General	-
Conceptual Design	• CFT (Concrete Filled Steel Tubular) element • SRC (Steel Reinforced Concrete) element
Engineering	-
Contracting	-
Pre Const./Detailing	• Steel concrete composite decks
Installation	-

Composite steel and concrete elements are common. The unique feature to model is at the interface between the steel members and the concrete, where there are usually special studs, ribs or meshes that create the shear connection between them. Some examples of composite elements are:

- Composite beams – steel beam with shear studs
- Concrete encased beams
- Steel Reinforced Concrete (SRC) column – determine accurate net concrete volume (Figure 34)
- Steel plate walls with concrete filling
- Ribbed metal deck
- Walls with embedded steel shapes (wide flange, etc.)

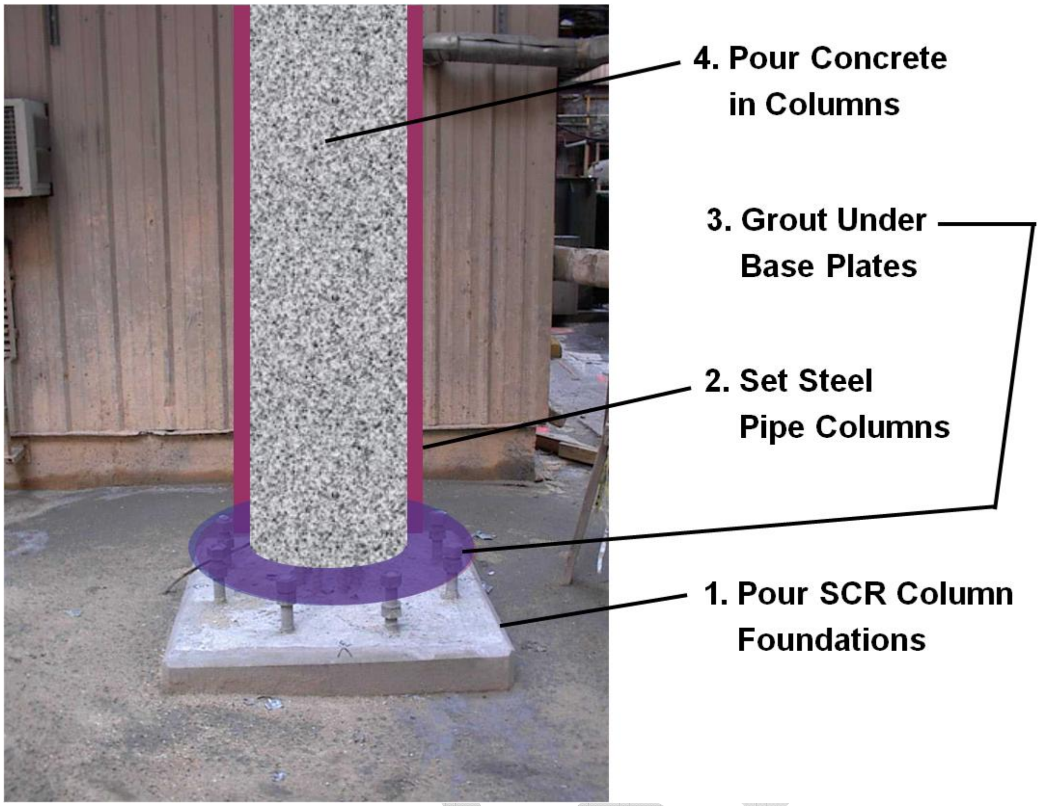


Figure 34. Steel Reinforced Concrete (SRC) column

3.3. Analyses and Design

This section covers the user requirements for engineering analysis and design of CIP structures. The process functions include:

1) modeling the physical structure,
2) defining loads,
3) defining the analysis model (sticks, plates and nodes)
4) structural analysis of assemblies (frames, floor systems, whole building structures)
5) recording the results of analysis (reactions, moments, shears, deformations)
6) design of individual members (concrete sections and reinforcing)
7) modeling the reinforcement intent, or detailing reinforcement, in members in the physical model
8) checking for satisfaction of design requirements and for code compliance

While, there are alternative sequences in which these functions can be done. The sequence above is largely followed for most structures. Structural analysis and design of reinforced concrete is performed using a variety of structural analysis models and design procedures, according to engineers' judgment. There are multiple software packages and routines, both commercial and those developed in-house in structural engineering design offices, for different models and members. Some packages focus on single activities among those listed above, but some also deal with multiple steps in a single continuous operation.

Therefore, the structural analysis and design interface of a BIM system for CIP concrete must enable two way data sharing for all of these steps. The interface should address the ability to perform some of the steps directly within the BIM model, with appropriate direct user interfaces, while others are done by external software operating through two way data exchange interface. The BIM software is largely used as a pre-processor and a post-processor to the external applications. As such, it must store all of the information (it can serve as the 'persistent data store' for structural information between runs of the external software) and also must enable visualization (reporting) of the information related with the different stages.

3D BIM method allows structural engineers to integrate structural analysis and design data of CIP reinforced concrete structures. Now, structural design data which is modeled in the preliminary design stage can be automatically generated into simple structural frame analysis, which is one dimensional analysis. Structural design data should be able to be changed according to the analysis results of automatically generated structural analysis model. Structural engineers require higher dimensional analysis to handle complex real problems of CIP reinforced concrete structures.

The following sections lay out both the data storage requirements and interface requirements for operating analysis and design software from the BIM platform. While recognizing that different software packages may bundle different functions together, its specification separates them for sake of clarity.

3.3.1. Defining loads

General	• Define all loads
Conceptual Design	-
Engineering	• Define loads and load cases for analysis
Contracting	-
Pre Const./Detailing	• Model temporary loads defined by time of action (e.g. load from shoring of slab above)
Installation	-

After modeling the geometry of the physical members of CIP reinforced concrete structures by creating structural elements in the BIM model, users need to apply various design loads. Applied loads can be point loads, line loads, area loads with uniform or variable distribution. A function to automatically distribute area loads applied to a structural element like a slab member is especially needed to lessen time-consuming modeling process.

Application of gravity loads to the physical members is more intuitive than applying the loads to the analytical model (see section 0). For building type structures, the gravity loads on floors generally require review of the Architectural/MEP plans, which is data that is more accessible from within the BIM model than the analysis model environment. Nevertheless, the functionality should also include provision of line loads to structural analysis members.

While most structural engineers feel more comfortable applying lateral loads (from wind and earthquake loads) to analytical members in an analysis software application, this can often present a challenge for the BIM model to maintain the loading information on the members throughout successive round trips of modeling and analysis. In the end, it becomes a balance/tradeoff of flexibility, which is normally in the analysis software, and the desire not to duplicate effort and introduce human error.

While pouring concrete, since distributed loads can affect structural safety, temporary loads defined by time of action should be recorded. Loads from shoring of a slab being poured can cause damage to structural members supporting the shoring such as slabs, beams and columns. This function would allow users to check the stress state of structural members according to concrete pouring to simulate the construction of CIP reinforced concrete structures to test this aspect of constructability.

Most structural BIM packages (like Tekla Structures, Revit Structures and Bentley Structural) allow point and line loads to be applied to structural members and also uniform pressure loads applied to areas (of a slab). They also support load cases and combinations at a basic level with the potential for two-way synchronization during modeling-analysis round trips.

Today, and in the near future, it seems that the structural engineer will remain doing most analysis tasks (while the BIM Modeler (drafter) performs modeling tasks) in the structural analysis software even though major geometry layout and preliminary member sizes can come from the BIM model. If loads originate in the analysis model, they need to be transportable back and forth from the BIM model in successive round trips, so that on a second trip from analysis back to model to analysis, previously defined loads are still specified for members of the model that have not changed. This should be achievable even without providing functionality in the BIM model to manipulate loads.

Ideally, the Engineer would have the ability to specify loads in BIM model with full mapping to structural analysis software with no loss of information in a round trip. The success of this process lies largely on getting the Engineer involved in the creation of the BIM model as well as having more mature capabilities for loading a structure within the BIM applications.

The following list identifies the loading types that must be supported:

- Member self-weight applied as a uniform (or linearly varying load for tapered members) in three main global directions (X, Y, Z) specified as gravity factors.
- Point loads on joints specified in global coordinates (Fx, Fy, Fz, Mx, My, Mz)
- Linear member loads
  - Single and Multiple point load definitions (six degrees of freedom applied at absolute or relative distance from member ends)
  - Uniform loads on full or partial length of member for six degrees of freedom based on local coordinates of member or global coordinate system of model
  - Linearly varying loads on full or partial length of member for six DOF based on local coordinates of member or global coordinate system of model
- Area Loads (applied to floor or wall elements)
  - Uniform pressure load in local or global coordinates of element
  - Allow for object to have a “span” definition for one or two way distribution of loads to supporting members
  - Allow for a point or line load on an area element with automatic distribution to the surrounding framing
  - Point, line or area loads distributed on an area but with attributes to reflect the changing magnitude of loads over time. This function is intended to enable modeling of time-dependent loads, such as the weight

of recently poured concrete that reduce as the concrete is cured, hardens, and eventually supports its own weight.

- Shell elements (drawn as individual shells or internally meshed from Area element definition)
  - o Uniform pressure load in local coordinates of element or global coordinate system
- Grouped combinations of any of the above, called loads cases. Load cases require a load multiplication factor for each of the loads included. Typical values are 1.4 for dead loads, 1.6 or 1.7 for live loads, and others for other load types. A factor for a specific load may be zero in any particular load case, indicating that that load does not apply under the conditions defined by the load case.

Load information for structures must be available to contractor for use in scheduling/sequencing, formwork design or for use by formwork designer / engineer and others. Load capacity over time is also another key piece of information. To maximize efficiency, contractors need to know the earliest possible time concrete can be loaded and with what. When re-shoring can be removed is also important. For example: "If I am constructing a building that has three levels of reinforced concrete with 8 levels of structural steel above, I would like to begin erecting the structural steel at the earliest possible time. When is the earliest I can begin to erect steel and how many levels can I erect over time while the concrete is reaching full design strength? This information could be communicated from engineer to contractor by assigning "Design Strength @ Hour" or "Design Strength @ Day" UDAs (assuming the material performs as planned).

3.3.2. Defining the analysis model

General	• Interface with other structural systems
Conceptual Design	-
Engineering	• Analysis input generation • Visible analysis model – sticks and nodes • Automatic change of design model in accordance with modification of analysis model • 3D finite element modeling of shells and irregular shapes
Contracting	-
Pre Const./Detailing	• Connect design and analysis results to a member
Installation	-

Most, but not all, modern structural analysis applications today can handle physical member data and automatically (internally) create a traditional finite element representation of the physical data for purposes of performing the numerical (finite element) analysis to determine forces, displacements, stresses, etc. With these types of analysis programs, the BIM software integration could be accomplished using a one-to-one mapping of the physical member to the analysis model member, greatly simplifying a "round trip" exchange of data.

The best way to achieve this is to maintain an 'analysis' model in the BIM software that parallels the physical model. In this way not only is the export to the analysis software package simpler, but the analysis results can be easily related to the physical structure when returned from the analysis package, because the relationships between the physical elements and the analytical elements (to which the analysis results apply) are maintained internally within the BIM system. This functionality is the subject of this section. In the text that follows, 'BIM analysis model' (or simply 'analysis model') refers to the representation of the structure within the BIM system that is the 'backbone' for structural analysis.

It is preferable that the analysis model be built by the system as the physical model is built, but also the user should have the option to delete analysis elements without affecting the physical members at any time and then "re-build" the analysis model with a single command.

The analysis model is also the primary way of combining different structural systems, such as CIP concrete with steel or precast parts or assemblies, for analysis.

Unlike the physical model, the analytical model must have a level of geometry integrity to be used successfully in analysis software. This includes basic elements such as Columns and Beams that must be connected to one another for structural analysis but can be offset from one another in their physical representation. This results in the analysis line not being positioned at the center of the member. Figure 35 provides a simple example that often occurs in high-rise frame structures, where one face of the column is fixed and one face varies as the column gets smaller throughout the high of the building. This example also shows a similar relationship with the beams, where one beam face usually stays in a single vertical plane and as the beam width is reduced, only the inside face is changed. The examples shown are from Bentley Structural. Autodesk Revit Structure 4 has very limited capabilities to model with this detail at the present time.

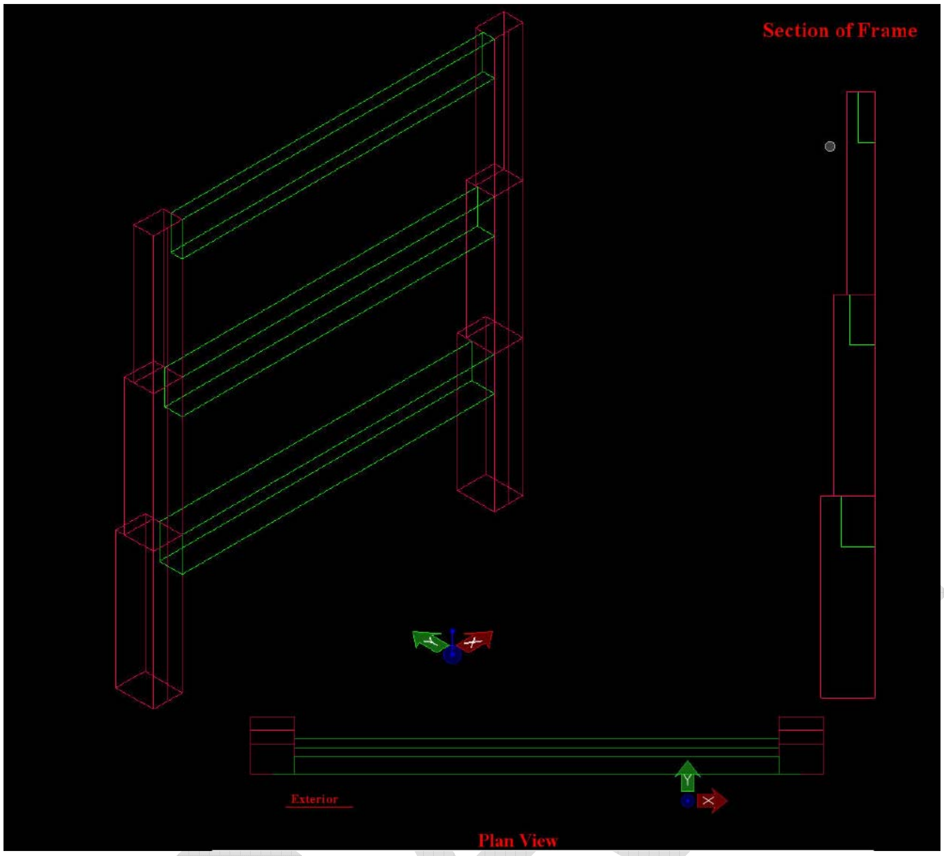


Figure 35. Simple example related to a level of geometry integrity

This detailed level of modeling is generally accomplished in allowing the physical member to have cross-section offset attributes (y & z offsets) while maintaining a consistent analytical line representation. Refer to Figure 36 and Figure 37.

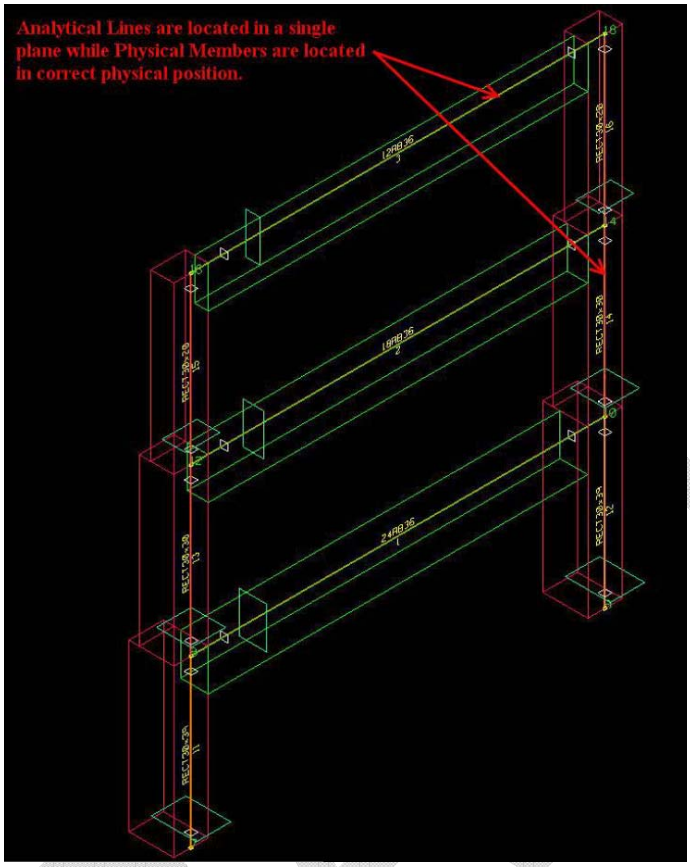


Figure 36. Modeling example for considering cross-section offset

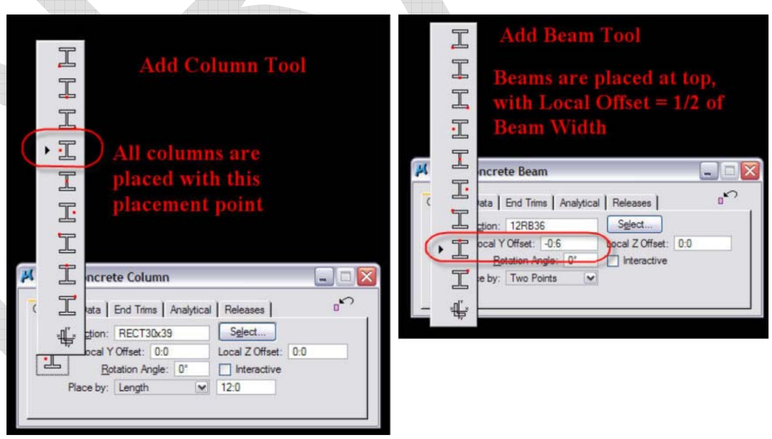


Figure 37. User interface for definition of cross-section offset attributes

Some analysis "purists" would consider this approach to be philosophically inconsistent but for most structures, this is the only practical way to deal with analytical modeling associated with more complex physical member configurations. Depending on the structural analysis program, these offsets may be able to be passed into the application to maintain a consistent graphical display (eg. S-Frame V7) and even allow this data to be round tripped back to the BIM software.

It is worth noting that offsets may also used in conjunction with a member's "placement point" parameter, which allows the cross-section to show relative to the work points (start/end) of a member.

3.3.2.1. Member Types of Analysis Model

The following subsections describe the modeling behavior and attributes for different member types of analysis model. The attributes are the parameters that would normally be passed to an Analysis/Design application, and must therefore be associated with the structural members of analysis model.

1) Linear Members (Beam)

- Start and end point (X,Y,Z) of the member's 'analytical line'.
- Rotation angle of the cross section relative to the central axis of the member.
- Y and Z offsets of the cross section relative to the analytical line.
- Cross section type (eg. Rectangle, circle, T, Inverted T, L, or custom shape. Note that in the case of a custom shape, the cross section definition in the BIM model must be available to the analysis software package as well. Where this is not the case, the cross section properties may suffice. The onus will remain on the engineer to check the capabilities of the analysis software before generating a non uniform user-defined cross-section that needs to be transferred for analysis.
- Cross section dimensions (parameters) (overall section profile) e.g. depth = 36, width = 24)
- Cross section properties (Ax, Ay, Az, Ix, Iy, Iz)

- Material properties (modulus of elasticity (E), shear modulus (G or Poisson ratio), density, coefficient of thermal expansion). If the analysis software will be used for design/code check, then the concrete compressive strength for design (f'c) is also needed.
- If the analysis software will be used for design/code check, then the appropriate member type for design (i.e. column, beam, wall) must also be provided. This relates to which clauses of the design code are checked for an individual member.
- Where columns are modeled physically to span multiple floors, the software should be able to split them into individual analysis elements. Similarly, continuous beams should be split into individual span elements for analysis.

2) Area Members (Walls and Floors)

Unlike linear elements, the physical modeling and analysis integration of area elements is considerably more complex, with a high degree of dependence on what information the analysis software can accept as input. A brief description of the parameters and functionality for each object type follows:

Wall Objects (e.g. foundation walls and shear (core) walls in a high-rise building, etc.)

- There is typically a one-to-one relationship of the physical wall segment and the analytical (shell) object in the BIM software. For a building, this usually results in shells too large for analysis purposes. Therefore, it is desirable for the BIM software to either mesh a physical wall internally, or at least to send the meshing parameters to the analysis software for meshing there. Figure 38 and Figure 39 show a simple example where the single wall is modeled as a single mesh element in the analysis model of the BIM software, but requires subdivision into at least four shell elements for actual structural analysis. The dialog box in figure 5 shows the preprocessing that an engineer needs to perform to divide the BIM analytical single shell into four shells for the actual analysis; in this case it is done in the analysis software – ideally, it could be done in the BIM software before exchange with the analysis package.

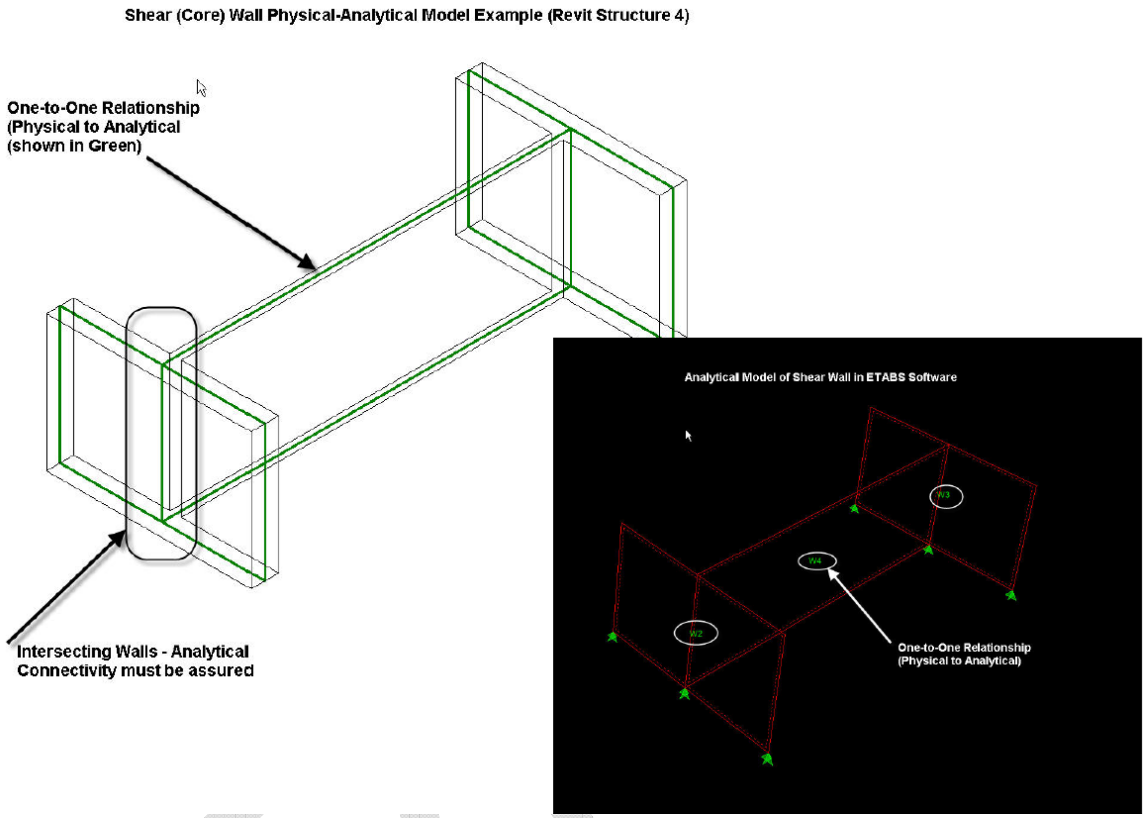


Figure 38. Shear wall physical-analytical model example

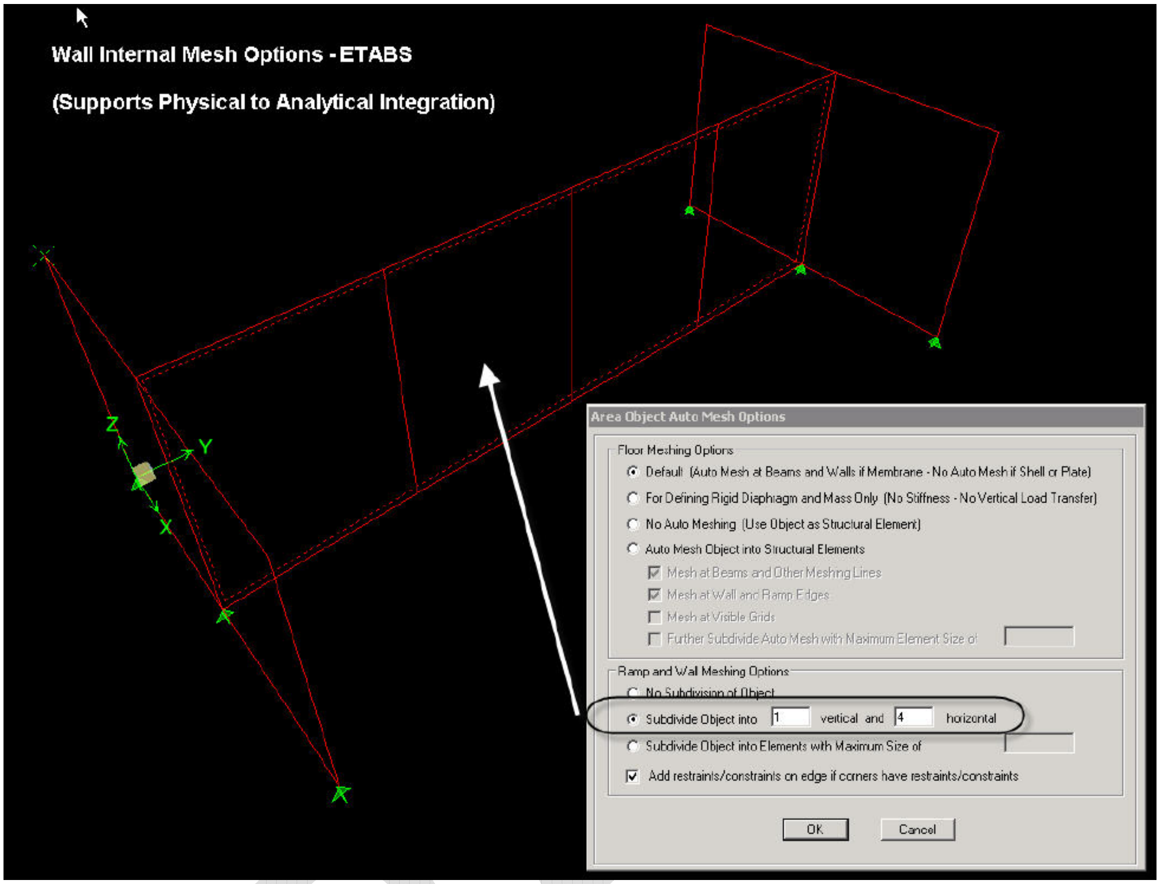


Figure 39. Wall internal mesh options

- It should be easy to model single walls across multiple stories (e.g. a single wall for a specific thickness that may occur over several stories in a high-rise) but with automatic splitting of the wall at floor lines (which is needed for analytical modeling). This could also be thought of as an automatic internal mesh parameter and become an attribute of the wall object. This is similar to the internal splitting that should be available for columns and beams.
- Optional analytical object for shell, automatic creation and recreation of analytical data.
- User-definable position of analytical planes anywhere within the extent of the wall and even beyond the extent of a wall.
- Analytical displays as a simple polygon with ability to turn on/off joint symbols, joint numbers, shell numbers, thickness label, and local axis of shell. (It would also be useful for shell to have two attributes for thickness: a

physical thickness parameter and analytical thickness parameter thus allowing the engineer to easily modify the thickness of the analytical shell without altering the physical thickness. This would be useful to take into account openings (that may not be analytically modeled) or other parameters that may effect the stiffness of the shell for analytical purposes only.

- It is essential that the wall has a “definition” line (or plane) and that the thickness is always measured offset from this line. The “definition” line is independent from the analytical plane and is used to define the physical wall in space predefined. Options for the “definition” line are on the left face, right face, or center of wall, custom offsets should also be possible.
- Support for Curved Walls (in plan) via a curved path “definition” line or other means. As the curved wall is not supported in most analysis software, allow the user to specify number of segments along the wall path to break into rectangular analytical shells.
- Should have ability to model “sloping” walls which are not defined vertical (parallel to z-axis) but defined along an inclined angle from the vertical. This could be handled with two “definition” lines at the top and bottom of the wall, or with a sweep angle from a vertical plane.
- Adjacent walls (that are touching) must know they are connected for purposes of analytical representation as they will share common joints.
- Openings in walls – user must have the option to decide if the analytical representation should be adjusted for openings. There are many times that small openings may exist in the physical model that are not of significant concern for engineering analysis and therefore are not included in analysis.
- Wall objects may have 3 or 4 node configurations to define their overall area.

3) Floor Objects (e.g. Floor slab in a building)

- The integration of Floor (Slab) objects to analysis will depend a great deal on how the analysis software handles these types of object. Some analysis software (e.g. S-Frame and ETABS) have the ability to handle single objects which can define a floor slab, with or without openings. Then, depending on what type of analysis of the floor is selected, the software must mesh the slab (internally or externally) into conventional 3 or 4 node shell elements or alternatively, calculate the tributary boundaries of the areas between framing members that are supporting the slab with the end result of determining the linear loads on the framing. Figure 40 and Figure 41 show an example of the use of Area Objects on a Floor to define loads on floor framing. Figure 42 and Figure 43 show the Area Object used to create a finite element mesh of the floor slab which results in both floor framing analysis forces and finite element shell forces in the meshed slab. Both of these examples utilize Softek S-Frame's robust Panel element Area Load object and Panel element meshing capabilities.

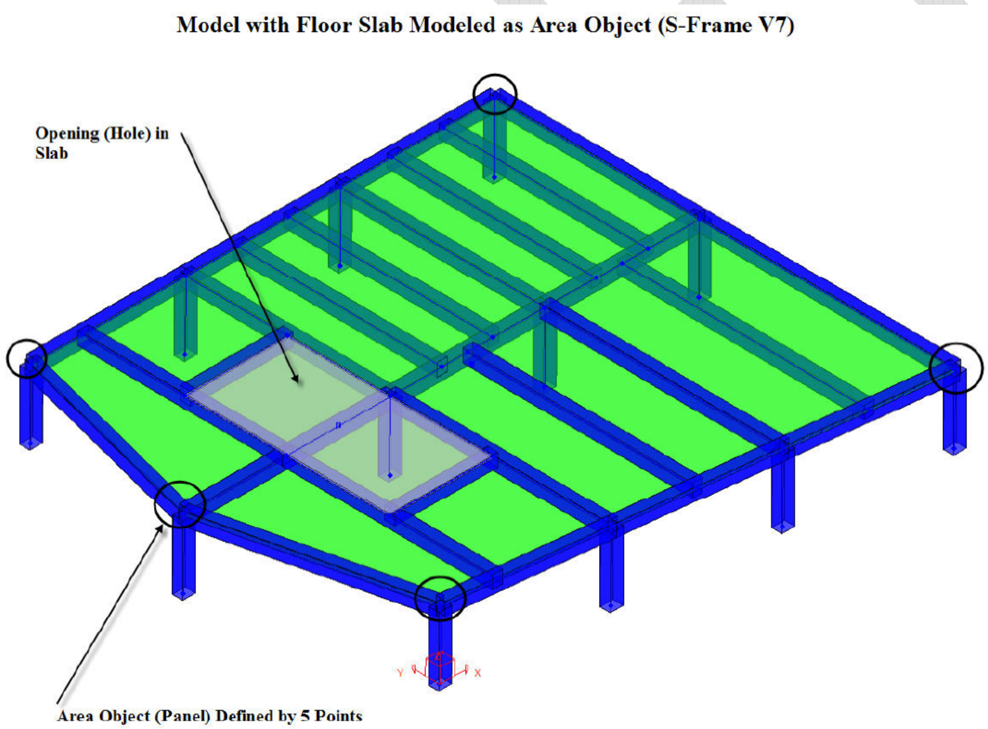


Figure 40. Model with floor slab modeled as area object (S-Frame V7)

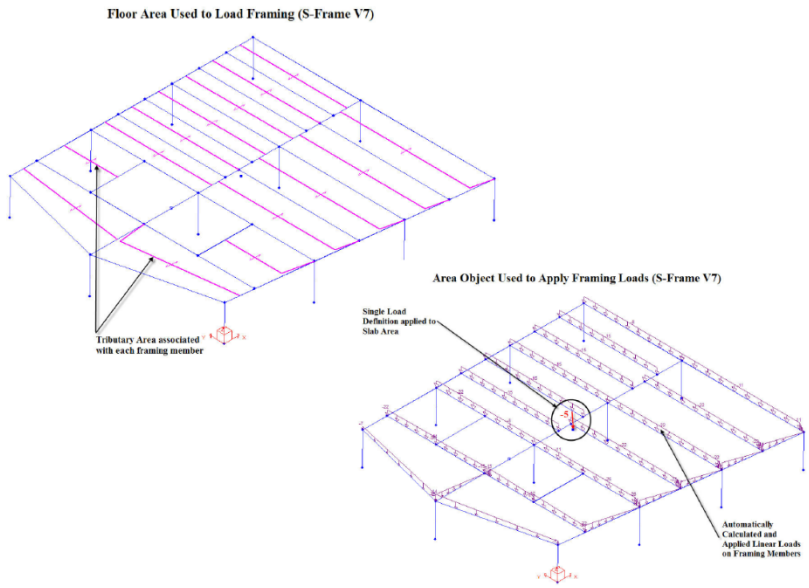


Figure 41. Floor area used to load framing (S-Frame V7)

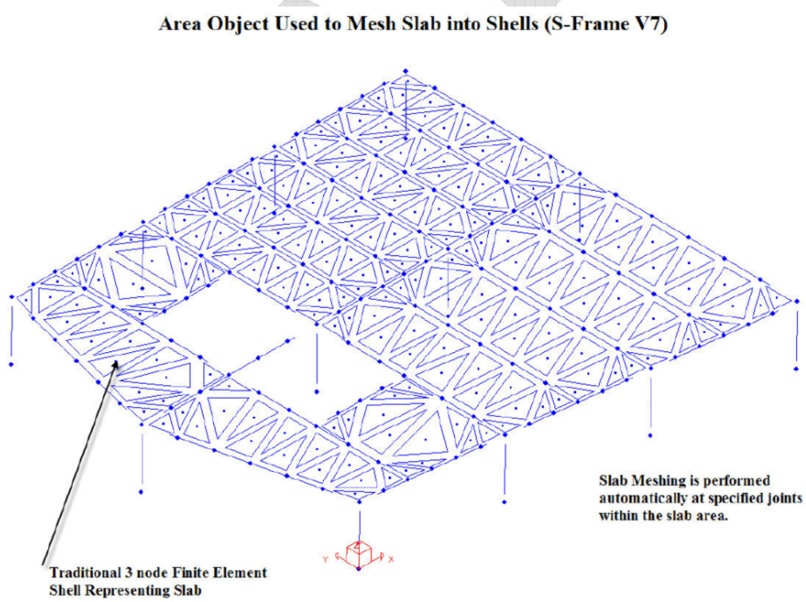


Figure 42. Area object used to mesh slab into shells (S-Frame V7)

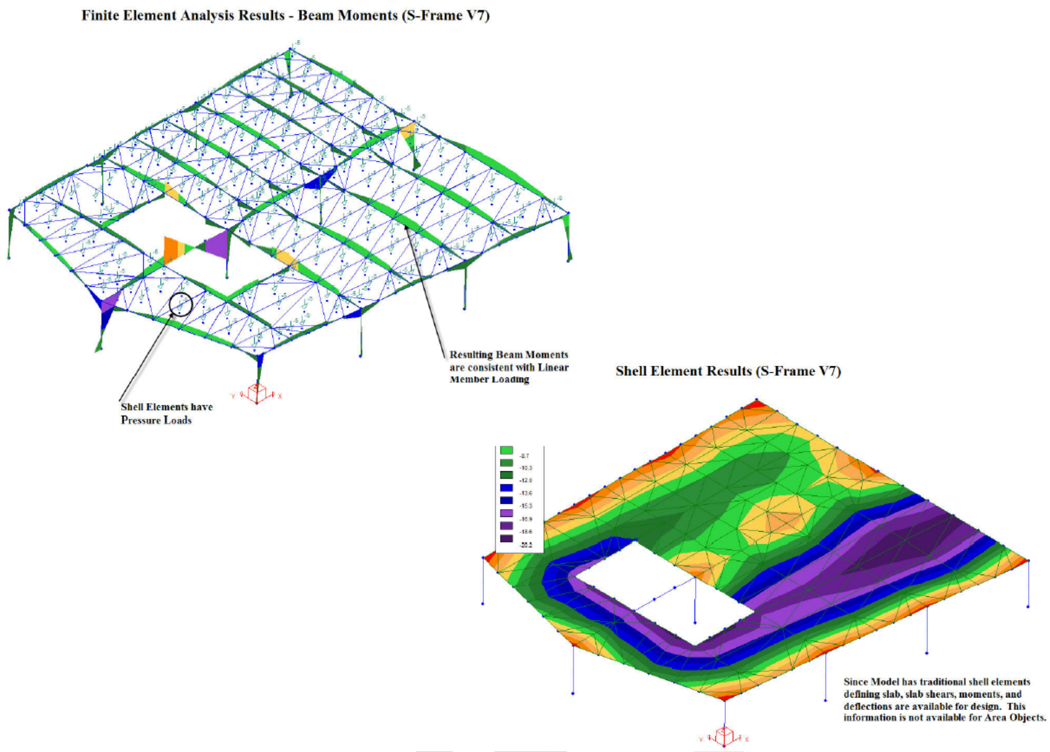


Figure 43. Finite element analysis results (S-Frame V7)

- Unlike linear beam-type elements, the modeling of floor slab elements and the integration of analysis and design of these objects with a round-trip back into the BIM software will require a detailed and comprehensive protocol for the exchange of data that goes beyond the intent of this preliminary specification.

3.3.2.2. Display of the Analysis Model

This subsection outlines the way in which the different analysis model members should be displayed on screen.

- Node numbers, member or area object ID numbers. Allow non-sequential node and member numbering for flexibility of data management.

- For linear members, also display local axis (XYZ), member releases, rotation angles, point and uniform loads. For area members, display local axis (xyz) and pressure loads.
- Joint supports (6 Degrees of Freedom – Dx, Dy, Dz, Rx, Ry, Rz)
- Show analysis model only, physical model only, or both together. Allow users to control whether internal meshing of analytical members (multi-story columns and walls, continuous beams, walls and floors divided horizontally) is shown or not.
- Analytical member creation should be optional for any member specified by user

3.3.3. Structural analysis of assemblies, recording and visualizing results

General	• Analyze structures and report results
Conceptual Design	• Stability analysis
Engineering	• Capability to visualize analysis results along with load conditions.
Contracting	-
Pre Const./Detailing	-
Installation	-

Data exchange is required in order to exchange the load and analysis model data with structural analysis applications. The exchange should ideally use an industry wide format, but may also be accomplished by generating input files tailor made for different applications. Table 3 lists some of the more common structural analysis packages and their export and import capabilities.

Table 3. Common structural analysis packages and exchange formats

APPLICATION NAME	IMPORT FORMATS				EXPORT FORMATS				DIRECT LINKS	OTHER FORMATS
	CIS/2	IFC	DXF DWG	SDNF	CIS/2	IFC	DXF	SDNF		
SAP2000	✓	✓	✓	✓	✓	✓	✓	✓	Revit Structures	Frameworks+, Excel, Access,
ETABS	✓	✓	✓		✓	✓	✓	✓	Revit Structures	Frameworks
STAAD-Pro	✓		✓		✓		✓		Tekla Structures, Bentley	VRML, Excel
RISA	✓		✓		✓		✓	✓	Revit Structures	
GT-STRUDL	✓		✓	✓	✓		✓			
RAM	✓		✓		✓		✓	✓	Revit Structures	
Robobat	✓	✓	✓	✓	✓	✓	✓	✓	Revit Structures	

Once analysis is complete, structural analysis results such as member force, stress, strain and deflection values should be exchanged back to the BIM software for post-processing, where they may serve for design, visualization and design verification. The quantity of data that is generated in a structural analysis is very large. For each element of a finite element model, results are reported for multiple nodes, and commonly for more than one layer (bottom, center and top). There will also be one set of results for each and every loading case that is considered. When retrieving analysis results, there are three options for coping with this information:

1) import all of the data into the model;
2) import only maximum values of each result type for each node, so that the envelope of values to be used for design is available.
3) Only import reference data pointing to external data files (generated by the analysis software package) that actually contain the analysis results.

All of these methods require that the analysis results be written by the analysis software into a file according to a standard format, or that the analysis package can store the data into the model using the model software's API calls. The latter requires that interfaces be written for every different software package; the former is therefore preferable. An obvious candidate for the data standard is the IFC format, but it requires that the analysis packages be able to write IFC output files.

Among the benefits of having analysis results available to the model are the ability to perform design of parts of the structure by reading the results into local member design packages (such as those for beams, columns and slabs). Another is the ability to visualize the structures behavior using graphical methods. The contour diagram (see Figure 43) is one example of the kind of post-processing that is required. The post-processing may be done within the BIM software directly, or alternatively external specialized design result post-processing software such as MSC. PATRAN can be considered. A third benefit is the ability to verify design decisions against the moments, shears and torsion stresses to which they will be subjected. It also becomes possible to trace backward and determine to what analysis types a rebar configuration was designed.

3.3.4. Design of members and modeling design intent

General	• Define member loads (from analysis results)
Conceptual Design	-
Engineering	• Output storage (outlines/concrete strength/rebar %/rebar strengths)
Contracting	-
Pre Const./Detailing	• Detail rebar and embeds according to design parameters
Installation	-

Unlike structural steel, the design (code checking) and general layout and detailing of reinforced concrete members are intimately linked. Beyond the overall concrete profile (cross section), the capacity of the member in accordance with a specific design code (such as ACI 318) is directly related to the number, size, spacing of bars and the type of detailing specified, such as splice type, etc. Therefore, to produce good concrete reinforcing detailing software, there has to be detailed and accurate design information available. Therefore, there may be no practical way to separate these tasks. Furthermore, visual review of the cross section or elevation of a beam, column, slab, or wall is important in evaluating practical constructability concerns as well as the appropriateness of the design.

Today, the structural engineer typically specifies concrete reinforcement for slabs, columns, beams, and walls in a schedule. Working with the schedules is a series of “typical” details describing how the reinforcement is placed within the cross section and along the member including location and type of splicing. These two pieces of information work together, along with the framing plans, for overall geometry, form the engineering input for the concrete formwork supplier and reinforcing detailer. In the near term, these schedules must be produced, separate from the detailers later layout.

A successful RC BIM application will allow automatic generation of slab, column, beam and wall schedules from concrete objects in the model thus still enabling the engineering to communicate “design intent” in a traditional manner. This information can then be used by the rest of the project team, including the reinforcing detailer, for construction documentation. In addition to scheduling of members and reinforcement, the engineer could use the BIM model in other manners. This would include the following:

- Link the member analysis forces and design documentation to the model and schedules. This could be accomplished by allowing external data (file or database data) to be linked to objects in the model, particularly, if the engineer desires to use an application that is not integrated with the BIM model.

- Calculate preliminary quantities of reinforcement based on reinforcing scheduled data and simple assumptions of splice locations, etc. This information could be used by the Engineer in preliminary stages of the design prior to the Detailer coming on board to produce a more complete detail model.

1) Concrete Design Software

Today's commercially available concrete design/code check software varies considerably in its capabilities and usability for basic to complex concrete structural elements. From an engineer's perspective, the applications that provide the most graphical feedback during the interactive nature of concrete reinforcement design are the most efficient and easy to use. Rather than attempt to describe functionality for concrete member design in a list of parameters, it might be more effective to take a particular commercially available application and highlight functionality that is important from an engineer's perspective. Please refer to **Figure 44** through **Figure 50** of the report for more information.

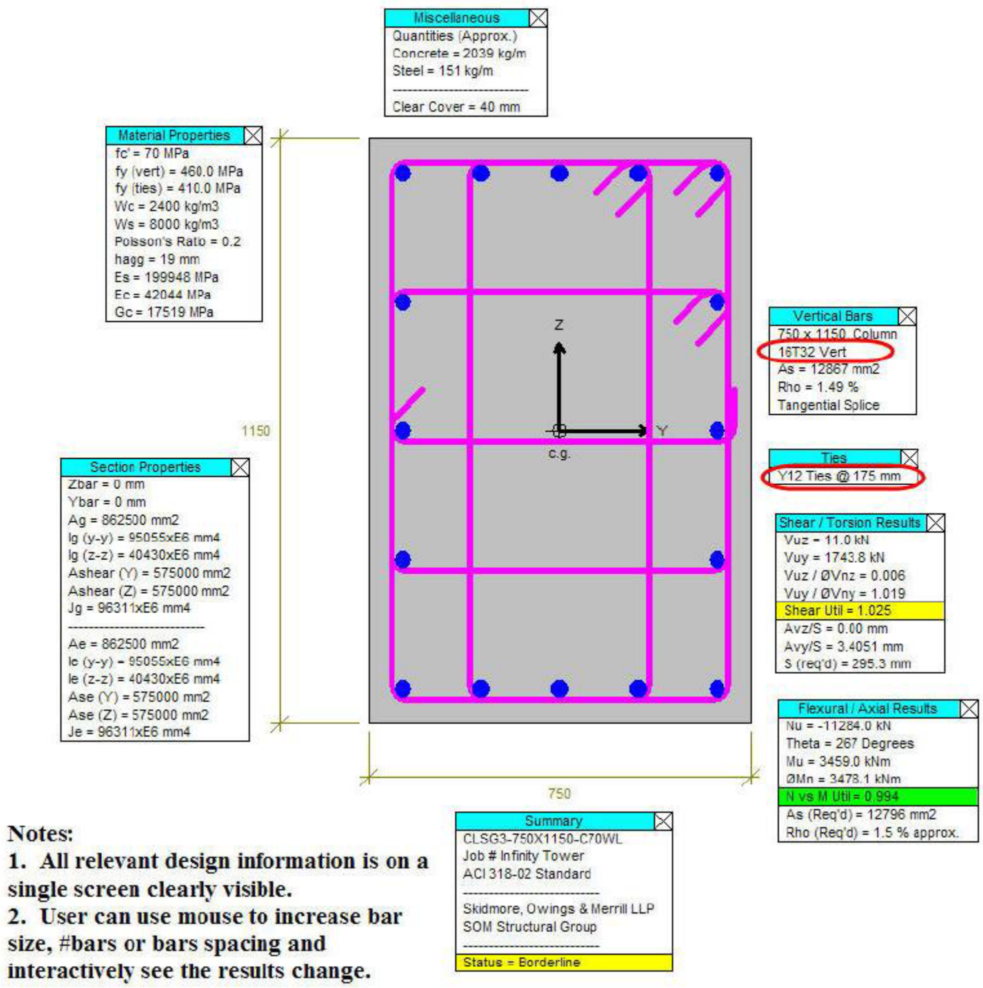


Figure 44. User interface for RC structural member design

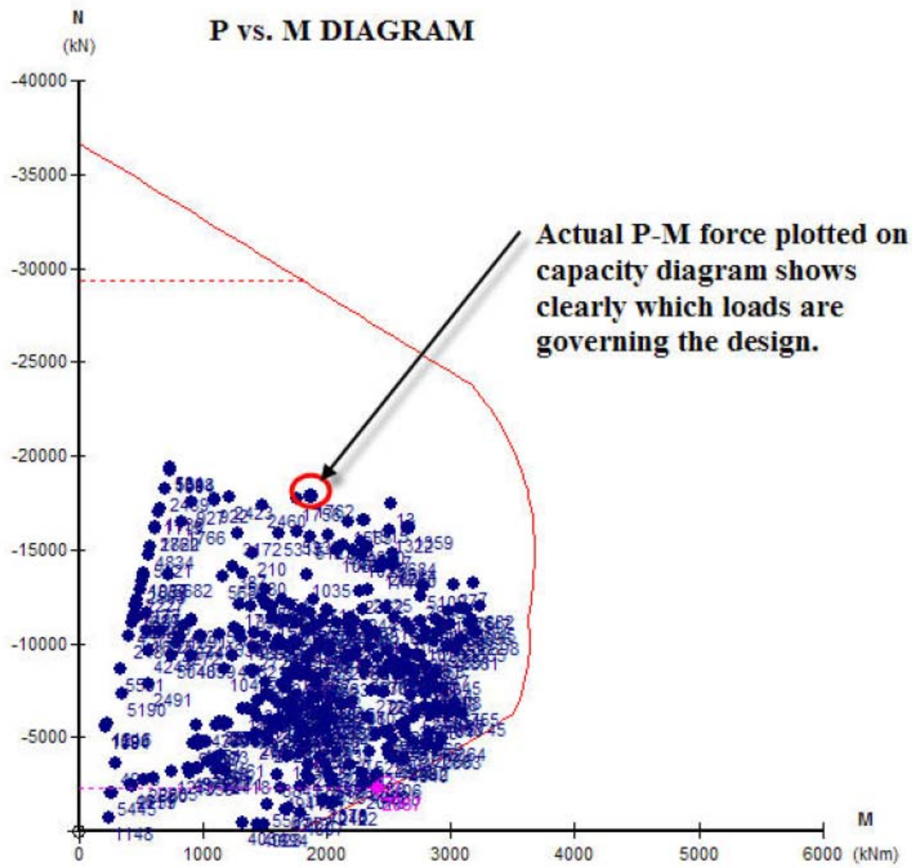


Figure 45. Actual P-M force plotted on capacity diagram

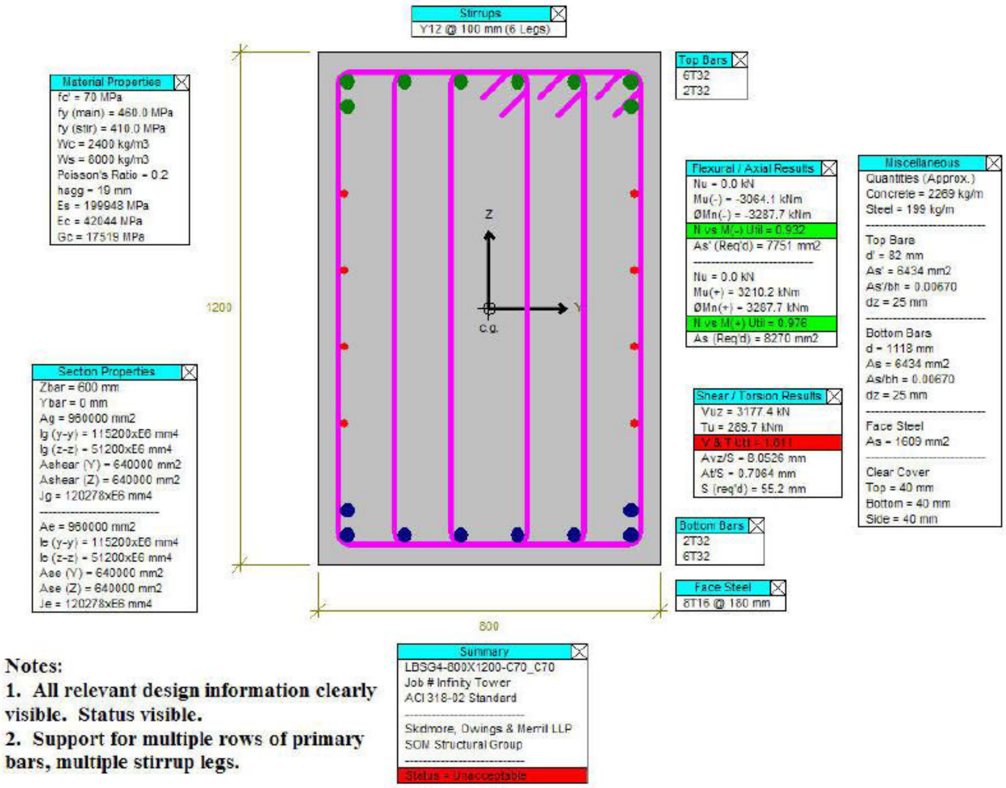


Figure 46. User interface for checking properties of RC structural member

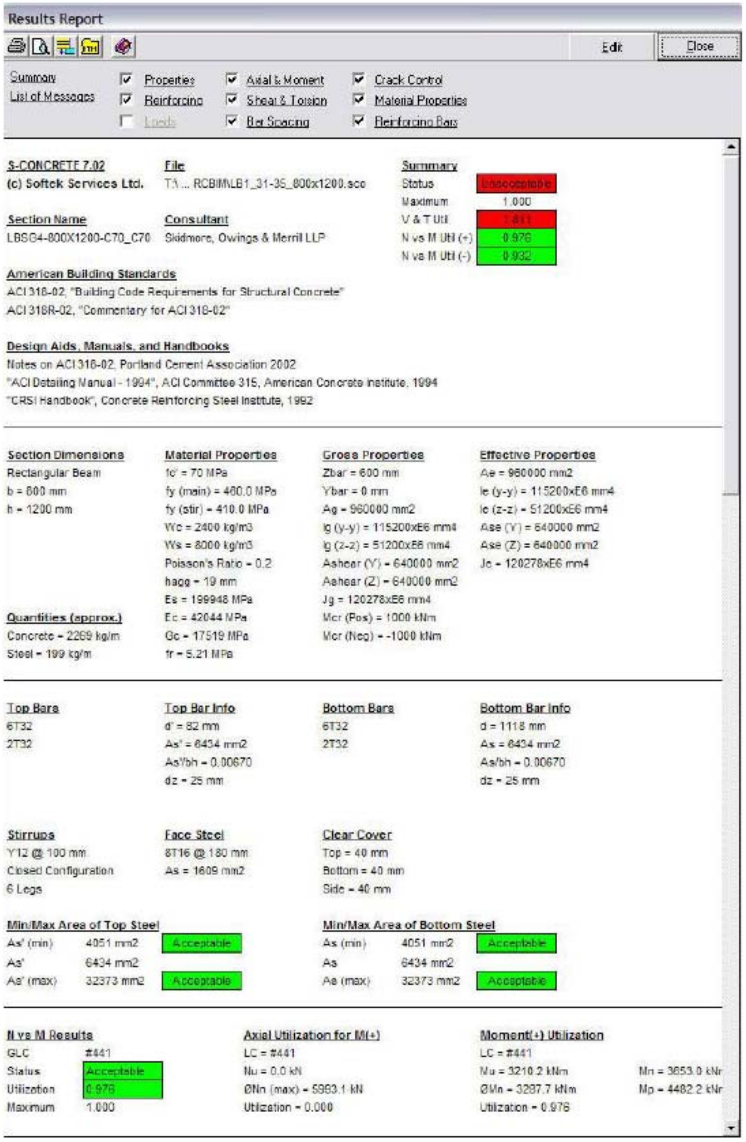


Figure 47. User interface for showing structural design results

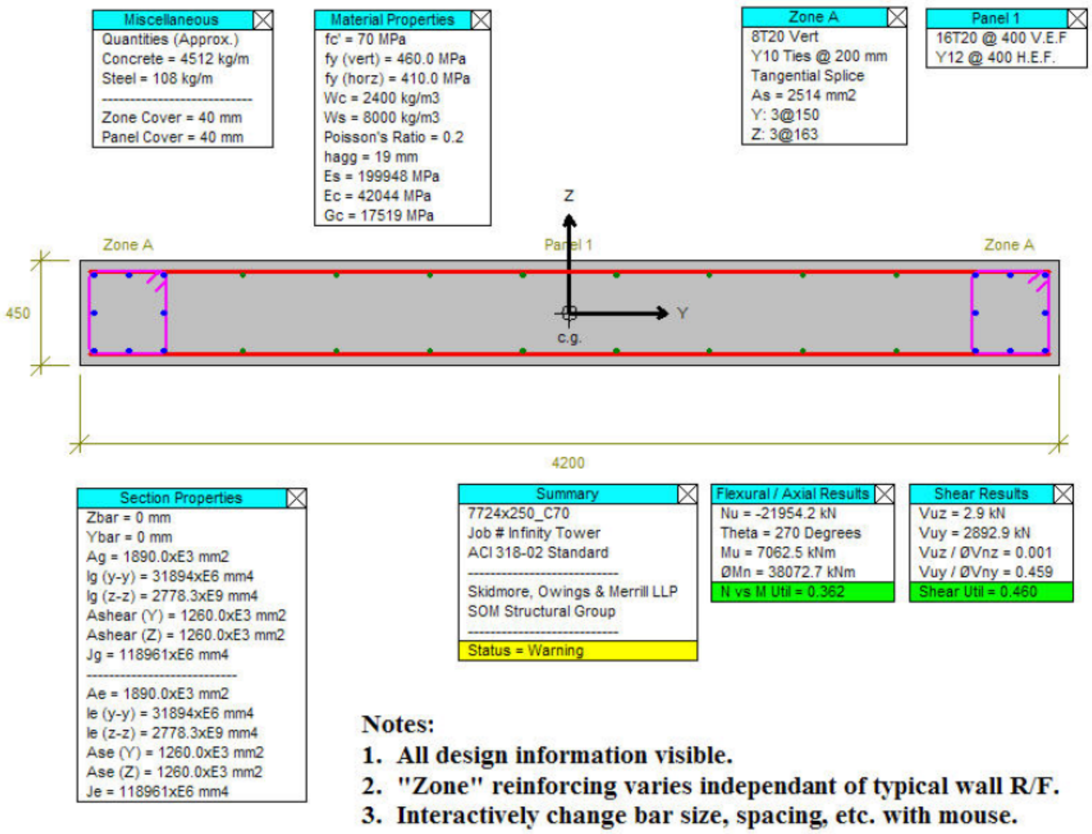


Figure 48. GUI for design of typical wall reinforcement

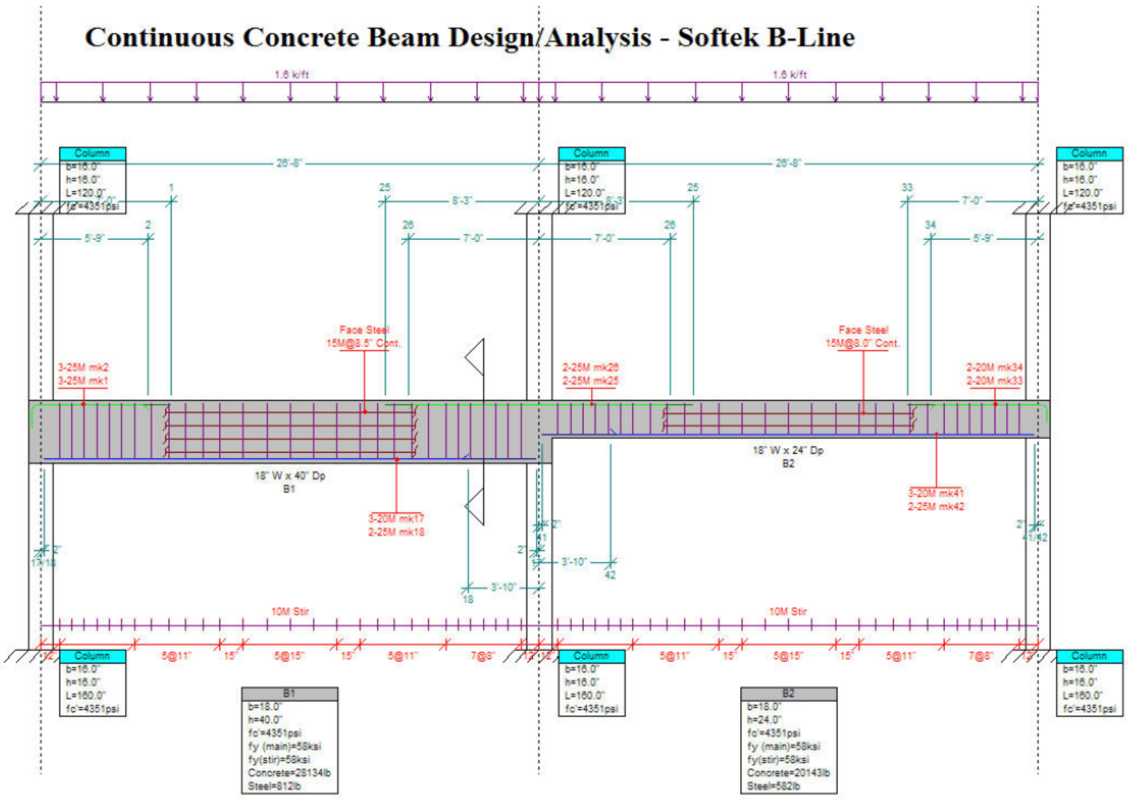


Figure 49. Continuous concrete beam design analysis – Softek B-Line

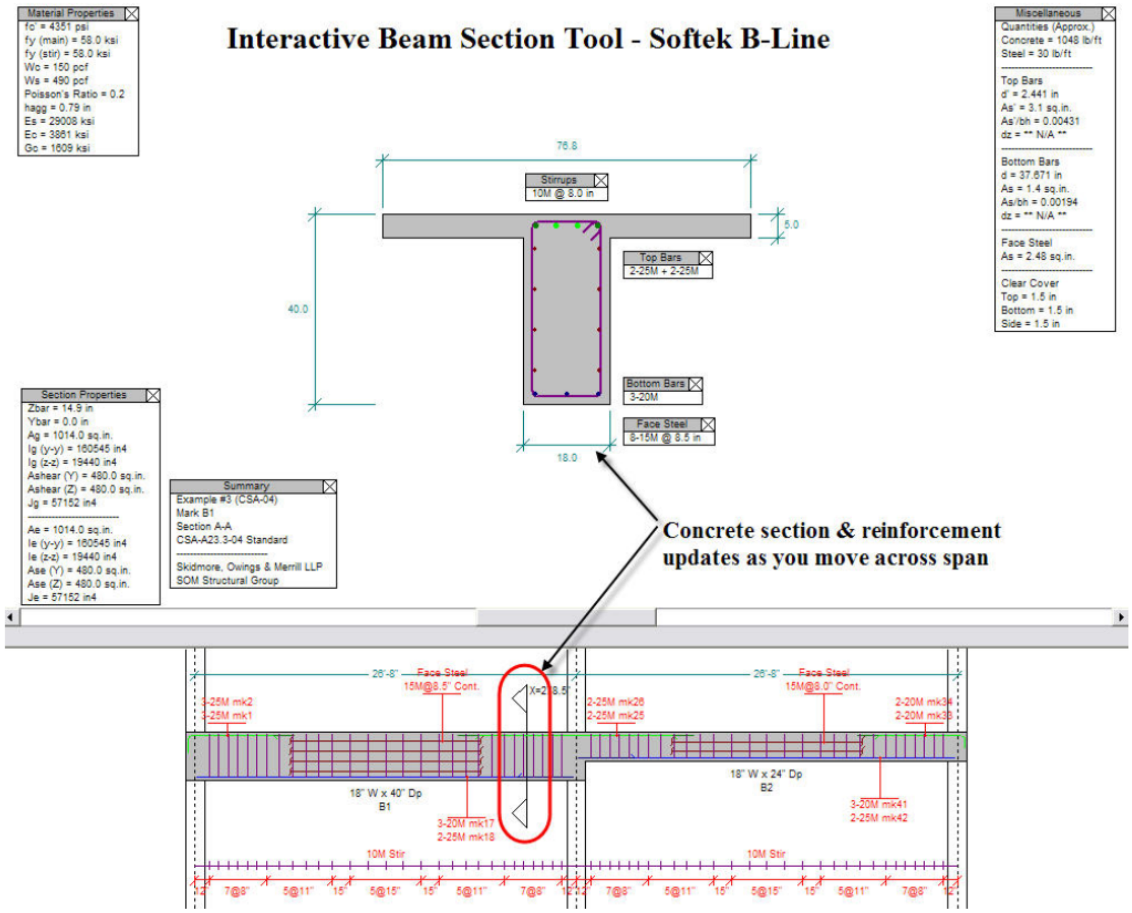


Figure 50. Interactive beam section tool – Softek B-Line

Structural design data can be changeable in accordance with results of structural analysis. To reflect the results of structural analysis to structural design members, mapping relation between analysis and design stage should be defined. This mapping relation can be described by inter-relation between each life-cycle of BIM models.

If analysis & design is combined in a single 3rd party package, additional concrete reinforcing-related parameters may be able to be mapped to the design application. These would include concrete cover (top, bottom, side), bar sizes (min., max.) for primary bars, ties, stirrups, and side bars and bar splice type (lap, mech. Etc.) This depends on what information the engineer wants to specify up front in the analysis and preliminary design process. Alternately, the member design application could generate this information and then pass it back to the BIM software to be used in the modeling and detailing process. The data needed also depends on where the concrete design is performed: in a structural analysis program or within the BIM software (maybe via a 3^{rd} party integration) relative to how design data is being shared between systems.

2) Design of continuous one-way elements for gravity load (slab/beam/girder)

- Requires definition of member properties, span lengths, etc.
- Requires definition of supporting members properties
- User should be able to group elements for defining different analysis runs – especially useful when there are slight centerline offsets from one span to the next
- Output storage of user-defined reinforcing bar types, number & size of bars

3) Design of two-way horizontal elements for gravity loading (slabs)

- Requires definition of member properties
- Requires definition of supporting member properties (columns, edge beams, drop panels)
- Requires definition of 2D floor geometry
- Requires definition of reinforcing zones
- Output storage of user-defined reinforcing bar types, number & size of bars in each zone

4) Design of 3D System for Lateral and Gravity (shear walls, moment frames)

- Requires definition of member properties (nodes, frames, membranes, plates, shells, releases, restraints, constraints)
- Requires definition of 3D geometry, including grids

- Ultimately requires interface that can maintain all applied loads, load cases, load combinations, analysis options (dynamic analysis, P-delta parameters)
- Input through an interactive database would be essential for larger projects
- Output storage of design data
    - o vertical bars - number, size, bar pattern
    - o horizontal bars/ ties - size, spacing

3.3.5. Code and constructability checking

General	-
Conceptual Design	-
Engineering	• Capability to check design against analysis model with functionality to link each member in the BIM model to its analysis condition and result. • Stability through incremental pours/stages • Code checking
Contracting	-
Pre Const./Detailing	• Stability through incremental pours/stages
Installation	• Concrete maturity and self-weight for shoring considerations

Since 3D building information models can provide all of the information describing CIP reinforced concrete structures, functionality for automatic code checking should be provided. Especially, functions for checking comparing analysis results with actual rebar design layouts and details, rebar congestion, and conformance to design codes, should be developed and be a type of automated/intelligent constraint checker. Such a constraint checker could handle design code and constructability checks.

1) Code and other constraint checking

Structural analysis model and structural design results of CIP reinforced concrete structures should satisfy the relevant codes that are commonly supported by professional organizations. The American Concrete Institute (ACI), for example, has published extensive requirements for CIP construction in recent years. The system should have the ability to check these requirements against the design. Not only are structural concerns important but "architectural" codes are important as well. For example, a parking structure must maintain a certain clear height throughout the structure. Parking spaces should be a certain width and depth. Barriers should be of a certain height. The architect may call for wall with form liners or a specific architectural finish. The system should allow this to be checked in the same way code compliance is checked.

The code-checking function should ideally consider a way for quickly reflecting periodically or frequently updated codes, possibly using online updating of the checking module.

Some of the key publications that relate to code/compliance checking of CIP concrete are:

<u>CRSI (Concrete Reinforcing Steel Institute) http://www.crsi.org/</u>
(placement / detailing) (the CRSI references the ACI / ACI references CRSI)
Manual of Standard Practice, 27th Edition
Placing Reinforcing Bars, 8th Edition
Reinforcement Anchorages and Splices, 4th Edition
Reinforcing Bar Detailing 2000, 4th Edition

<u>PTI (Post-tensioning Institute) http://www.post-tensioning.org/</u>
Post-Tensioning Manual, 6th Edition
Construction and Maintenance Manual for Post-Tensioned Slab-On-Ground Foundations
Design Fundamentals of Post-Tensioned Concrete Floors
Design Fundamentals of Post-Tensioned Slabs-On-Ground

Design, Construction and Maintenance of Cast-In-Place Post-Tensioned Concrete Parking Structures

Instructional Manual for Training and Certification of Field Personnel for Unbonded Post-Tensioning - Level 1

ACI (American Concrete Institute) http://www.aci-int.org/

Manual of Concrete Practice 2006

Manual of Concrete Practice 2006 CD-ROM

Manual of Concrete Practice + CD-ROM 2006

IBC – International Building Code

2003 International Building Code, Volume 1

2003 International Building Code, Volume 2

2003 International Energy Conservation Code

2003 International Existing Building Code

2003 International Fire Code

2003 International Fuel Gas Code

2003 International Mechanical Code

2003 International Plumbing Code

2003 International Property Maintenance Code

2003 International Residential Code, Volumes 1 & 2 Combo

2003 International Residential Code, Volume 1

2003 International Residential Code, Volume 2

2) Constructability checking

Constructability can often be a relative term. Degrees of constructability vary based on who the contractor is, the skill set of the workforce and the equipment available. Constructability issues arise in formwork, rebar and concrete pouring.

Constructability can be dealt with at different levels of sophistication, ranging from a simple set of universal rules to a system that is context sensitive. The description in the text box below (Figure 51) describes a context-sensitive and fairly 'intelligent' approach. As a first step, some of the universal rules can be defined, such as rules to identify situations where:

- unsupported rebar cannot be tied,
- rebar congestion prevents concrete placing and compaction,
- formwork in constricted places does not have working clearance for workers to close and tie forms.

Figure 52 depicts rebar that is called out at the top of a grade beam with no means of support. The system should have the ability to check for this sort of constructability issue at various points in the work flow; detailer or engineer. Another example of constructability would be a vault design that does not allow for the formwork to be removed after casting is complete.

Reinforcing constructability issues occur on many projects, especially post-tension structures. This is primarily due to the nature of design. The reinforcing steel is designed by the structural engineer but the post-tensioning is only conceptually designed; supplied forces. The post-tension is most often not coordinated with the reinforcing from a constructability perspective. The system should allow for the verification of constructability between the two systems, i.e. placement sequence, support, etc. The system should allow for these checks to occur at any stage of the workflow. Another example might be that the structural engineer may choose to model the steel reinforcing without laps. At that point, the reinforcing steel may have a high degree of constructability. However, when the lap splices are applied by a detailer, there may be a low degree of constructability at some or all of the lap splice locations. Constructability analysis must take into account who the user is and their position in the overall workflow.

The system must have the ability to assign constructability schemes based on the region, the contractor, the owner and so on. The system must allow these schemes to be modified at will. For example, if the structural engineer specifies sand as sub-base material in a region where it is not readily available or cost effective, the system must alert the structural engineer that sand is not readily available or cost effective but gravel is. The system should allow for this sort of criteria to be assigned at job setup or during the modeling process. Another example from the perspective of the contractor and/or detailer would be that the structural engineer that they are working with on a particular project does not like the use of a particular anchorage brand. The system should have the ability to limit anchorage choices based on the preference of the members of the project team.

The system should also allow the user to input max values for pour sizes and should be alerted when the value is exceeded. This could also be considered a soft clash. These functionalities are more textual in nature but there are some constructability issues that are related to geometry. For example, everyone has heard the term "sky-hook" or "2 x 4 stretchers". The system should have the ability to alert both the structural engineer and the detailer when these situations occur.

Figure 51. 'Intelligent context-sensitive constructability checking

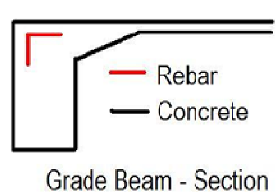


Figure 52. Unsupported rebar

3.4. Model Concrete Material

3.4.1. Design and model concrete mix

General	• Time-dependent attributes (strength, creep, shrinkage)
Conceptual Design	• Conceptual concrete mix for elements
Engineering	• Cement type • Designed psi • Shrinkage, creep, deflection, thermal gradients, continuity • Expected quality data
Contracting	-
Pre Const./Detailing	• Admixtures • Aggregate size and type • Slump
Installation	• Concrete maturity and self-weight for shoring considerations • Placement • Curing • Monitoring • Cylinder strengths

The system should support two ways of assigning material properties to an object. The first method is to assign material properties directly to the object, and the second method is to assign material properties by associating a mix type to the object. Reinforced concrete design specifies concrete strength and mix types. Mix types can be chosen from a list of predefined design mixes depending on the site conditions.

The first method allows association of object specific information such as actual strength, design strength, water/cement ratio, or a reference to specification, which cannot be associated directly with a mix type.

Type of mix should be assigned to a segment of an RC structure. A type of mix should carry information on the design of the mix, cement type, cement strength, aggregate type (e.g. for heavy concrete).

Concrete strength is usually tested and logged for a series of 7day, 28 day and/or other user specified duration tests.

When designing a concrete mix, four sets of properties need to be defined:

1) **Main attributes**: A concrete mix is designed to serve four main design requirements: concrete strength, service conditions and exposure (e.g. in a sulphate rich environment, radiation etc.), placement technology (manually or pump) and compaction and placing, which define the maximum aggregate size, the ratio between large and small aggregates and the slump. The first two requirements, concrete strength and service conditions, are design related whereas the latter two, placement technologies and placing, have more to do with the construction itself.

   Figure 53 shows a simple mock up of a typical Tekla style user interface that serves to summarize the primary data that are needed to define a concrete mix. The data types were culled from mix design forms of the kind shown in Figure 54. The list of information in Figure 53 is not necessarily complete.

   The curing process instructions are part of the main requirements. These instructions can be either predefined or a set of free text.

2) **Special requirements**: For special mixes other properties should be defined, such as desired weight (whether it should be a heavy mix or light weight mix). Other properties can be special admixtures for special purposes, such as accelerators, plasticizers, etc.

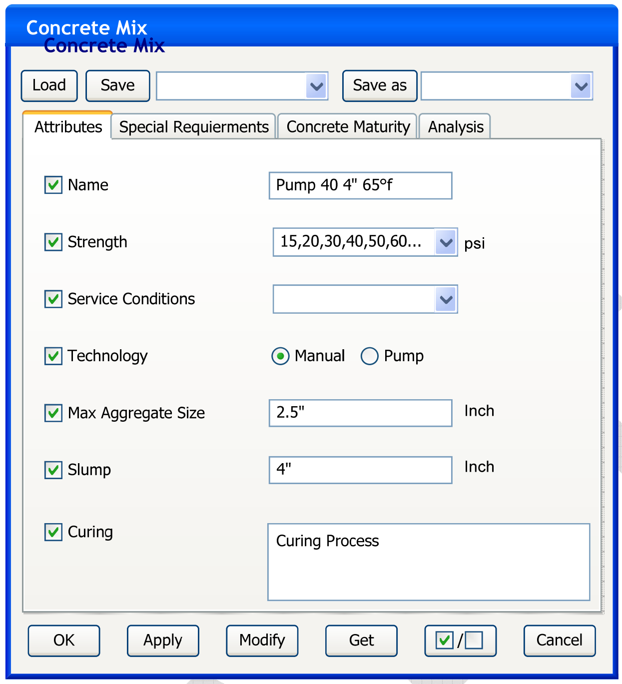


Figure 53. Suggested main requirements tab of concrete mix form.

3) **Concrete maturity**: Concrete maturity information is usually given as a percentage of the maximum strength after 28 days. This information is directly connected to the temperature expected during the actual pour and during the curing period. For example, a mix designed for 30% maturity after 7 days at 75°f would have a different maturation rate if it were poured at 45°f.

4) **Analysis information**: This set of data is to be used when running structural analyses. It includes parameters like modulus of elasticity, Poisson's ratio, shrinkage (mm/m) or creep (mm/m).

Monitoring concrete maturity and cylinder strength is related to the actual pour (see section 3.6.1). From each pour batch several cylinders are taken to perform strength tests. These are usually done after 7 and 28 days (there are cases where more precise monitoring is needed). This information would be accumulated in a database at the company or project level, and it should be possible to import the data for each concrete pour into the appropriate pour work package object. In this way, it can be cross referenced with the mix information for each work package, so that the actual concrete maturity and final strength are recorded. It should then be possible at any point in time to highlight - in color or by filtering - any objects in the model that belong to a pour section and that have not developed their planned strength. This can be critical for considering shoring and formwork support of subsequent parts of the structure not yet poured, and so is a 'mission-critical' piece of information for production planning.

Concrete Mix Design OPC+MS (Gabbro Agg.)

Mix Classes:- A1,A2,A3,B1,B2

IN ACCORDANCE WITH ACI 211.1 & BS : 5328 : 1990 : PART 2(Including BS 882)

CLIENT	NEW DOHA INTERNATIONAL AIRPORT STEERING COMMITTEE
ENGINEER	OVERSEAS BECHTEL, INCORPORATED
CONTRACTOR	TREMIX-ALJABER ENGINEERING
PROJECT	ON -SITE CONCRETE SUPPLY (P.N 25045-000-FC1-DB50-00008)
LOCATION	NEW DOHA INTERNATIONAL AIRPORT STEERING COMMITTEE

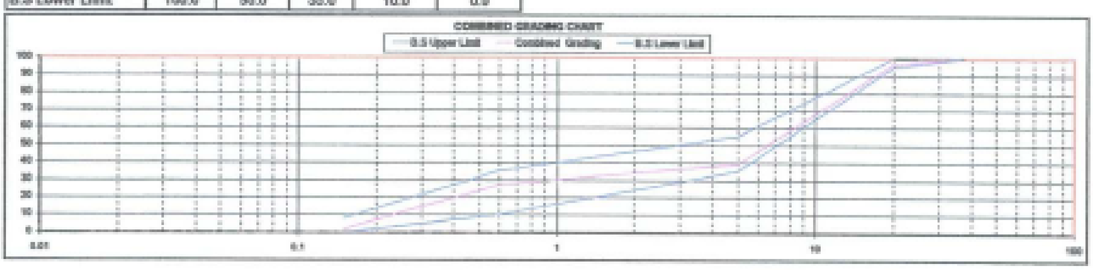
Aggregate size	Type	%agg
20 MM	Crushed Gabbro	37.5
10MM	Crushed Gabbro	25.3
Dune sand	Washed Sand	37.3
		100.0

Mix Class	Cement	Type
55,50,45,37 N/mm²	330 + 0 + 20	OPC+GGBS+MS
fw/c ratio	Target slump mm	% Air
0.36	100 ±25	1.5

Material Size	Volume ltrs/M³	Specific Gravity	Qty Kg/M³	Absorption %	Moisture Correction	Final Qty Kg/M³	Cumulative Kg/M³
Cement	104.76	3.15	330	-	-	330	330
GGBS	0.00	2.89	0			0	330
MICRO SILICA	9.09	2.2	20			20.0	350
Water	126.00	1	126	-	22	148	498
Agg. 20 MM	268.17	2.85	764	0.85	6.5	758	1256
Agg. 10 MM	181.81	2.84	516	1.3	6.7	510	1766
5MM Washed Sand	291.27	2.61	760	1.2	9.1	751	2517
FPC 320 (FEYZIN)	3.40	1.08	3.7	-	-	3.7	2520
FR 008 (FEYZIN)	1.80	1.19	2.1	-	-	2.1	2523
Air content	15.00			-	-		2523
TOTAL			2041				
Total Agg S.G.C		2.7581					

Sieve Size	37.5 mm	20 mm	5 mm	0.6 mm	0.15 mm
20 mm %	100.0	92.0	1.0	0.0	0.0
37.5	37.5	34.5	0.4	0.0	0.0
10 mm %	100.0	100.0	6.0	0.0	0.0
25.3	25.3	25.3	1.5	0.0	0.0
0-5 mm WASH %	100.0	100.0	100.0	73.0	5.0
37.3	37.3	37.3	37.3	27.2	1.9
100.0					
Combined Grading	100	97.0	39.1	27.2	1.9
B.S Upper Limit	100.0	100.0	55.0	35.0	8.0
B.S Lower Limit	100.0	95.0	35.0	10.0	0.0

RAW MATERIALS SOURCES	
RAW MATERIAL	SOURCE
CEMENT	QNCC
20MM CR AGGREGATE	U.A.E Gabbro
10MM CR AGGREGATE	U.A.E Gabbro
5MM WASHED SAND	QSPT-QNCC



FOR TECHNICAL READY MIX CONCRETE CO. LLC

Figure 54. Mix design form - Type I (Bechtel)

From Contractor Tacoma Narrows Constructors By (signature) ____________

To WSDOT Project Engineer ____________

The following mix design is proposed for use on:

Contract Number ____________ Contract Name Tacoma Narrows Bridge

for Concrete Class: *(check one only)* ☒ 3000 ☐ 4000 ☐ 4000P* ☐ 4000D* ☐ 4000W ☐ Lean Concrete ☐ Cement Concrete Pavement*** ☐ Concrete Overlay** Remarks 4.5" Slump

To be used in Bid Items 250, 252, 255 it is proposed for supply from Glacier Northwest, Inc. (supplier) ____________ (plant location)

Designated Mix Design No. (3990) (   )

	Contractor Provided Mix Design		Combined Gradation Mix Design (Attach 0.45 Power Chart)	
Cement	Type I-II 376	lbs/cy	Type ____	lbs/cy
Fly Ash	* 125	lbs/cy	*	lbs/cy
Microsilica	**	lbs/cy	**	lbs/cy
Latex Admixture	**	lbs/cy	**	lbs/cy
Fine Aggregate	Pit Number B-335; (ssd) lbs/cy: Class 1 ____; Class 2 1,315		Aggregates: Pit Numbers ____; Max. Aggregate Size ____	
Coarse Aggregate	Pit Number B-335; (ssd) lbs/cy: AASHTO No. 467 ____; AASHTO No. 57 ____; AASHTO No. 67 ____; AASHTO No. 7 ____; AASHTO No. 8 1,960		Sieve Size / % / (ssd) lbs/cy	
Water (Max)	267	lbs/cy		lbs/cy
Water Cement Ratio	0.53	Max		Max
Water Reducer		oz/100 wt		oz/100 wt
High-Range Water Reducer		oz/100 wt		oz/100 wt
Air Entrainment		oz/100 wt		oz/100 wt

* Required for Class 4000D and 4000P mixes ** Concrete Overlays Only *** Cement Concrete Pavements

Mix Design 28-day compressive strength 3000 psi Mix Design Density 148.73 lbs/cf

Aggregate Correction Factor .002 Fineness Modulus 2.65

*** Flexural Strength ____ ____ ____ ____ ____ Average ____ psi

*** Compressive Cylinder Strength ____ ____ ____ ____ ____ Average ____ psi

Reviewed By: PE Signature ____________ Date ____________

DOT Form 350-040 EF Revised 11/00

Distribution: Original to Contractor
Copies to: OSC Materials Lab; Region Materials Lab; Project Inspector

Figure 55. Mix design form - Type II (Bechtel)

3.5. Reinforcing

3.5.1. Detailing

General	• Standard shapes, available lengths, hooks, other geometric, constraints
Conceptual Design	• Generation of rebar steel material requirements
Engineering	• Conceptual rebar schedule interface • Checking for proper rebar layouts • Automated creation of rebar layout for joist, waffle, and pan. • Capability to choose a graphical representation scheme for rebar schedule out of several predefined sets of representation methods.
Contracting	-
Pre Const./Detailing	• 3D rebar array • Multilayer rebar group • Cover/Anchorage • Automated splicing/overlapping, according to standard rebar maximum lengths. • Elimination of clashes and redundant placements • Continuous beam reinforcing macro
Installation	• Sequencing • Pre assembly • Rebar accessories: support chairs, rebar caps, terminators, splices/connectors • Move laps/splices according to changes in pour stops and/or break locations made during construction

Detailed design for reinforcing has a lot of issues. Reinforcing bar is a structural element for enhancing tensile property of concrete. Hence, the relation between reinforcing bar and structural member should be clearly definable. One of main issues in CIP RC structures is to define structural members based on connecting volumes for representing monolithic element. In addition, reinforcing bars can be a connecting part between structural members such as column to beam, beam to slab and column to slab. Thus, the reinforcing bar can be considered as internal members such as embeds, tendons. And the internal member is a part of connecting volume. Followings are user requirements related to reinforcing bar,

- Automatic rebar layout should rely on the overlapping structural shapes that have been already defined. These shapes should define the logical shape that is to be used for reinforcing layout for the functional designation of that shape.

- Standards, specifications, and practices of company typically determine details of rebar design including spacing, quantity, overlap, shape, connection, layout, aggregate size and other details. The system should provide detail library that covers standards, specifications, and practices of a company and should also allow flexibility to add new type of details to the library. Change of details should be propagated associatively when the changes are made to the design of the details.

- Rebar comes in different sizes depending on the site and nearby rebar mill's capacity. Placement of extensions overlapping, and splicing varies depending on the stock length of the rebar. The system should automatically adjust layout of rebar depending on the stock length. Strength, type and other specifications of rebar material properties should also be considered in rebar detailing, as length of overlaps, hook and connection detail are affected based on material properties.

- The system should validate rebar layout detailing based on code, specification, and company practice. Validation includes soft clash checking depending on aggregate size and concrete slump for dense area, such as column and beam joint, dense overlap area, and other area.

- The system should provide predefined and expandable rebar layout lists for special structures, such as waffle slabs or joists. These predefined details should have parametric behavior depending on their dimension and structural requirements.

- Rebar layout should reflect material characteristics so that modeling should be flexible to changes of rebar.

- Rebar can be layered in multilayer design. For example, main tension bar for beam and girder can have multiple layers with different configuration. And slabs can be designed with multiple layers of rebar.

- Classification of rebar for design, construction and material ordering can be different. In the design stage, anchoring hooks for a beam should be counted as rebar for beam, while in the stage of construction or material order, the rebar should be treated as rebar for column, so that rebar should be delivered and installed while the column rebar is being installed. This requirement is expressed in the section 'Special Aggregations' in chapter 4, which stipulates that all materials such as rebar and embeds must be included in the BOM of the first work package scheduled for execution with which they physically intersect or within which they are contained.

- As CIP is a continuous object, rebar expands to other parts as well. A rebar used for a beam, a column or a slab easily extended as far as a unit length of the supplied rebar. Using uncut rebar is also recommended for a sound structure as well. Detailing for rebar placement should reflect the use of stock length rebar, while classification for each stage of practice also addressed accordingly.

- Rebar layout also needs to consider placement of special accessories such as support chairs, rebar caps, terminators, splices/connectors. Geometric representation may not be necessary in the system, but counting the number of required accessories on floor, work unit or pour unit is a required functionality of the system. For safety concerns, some projects require safety caps for the rebar ends. Safety caps are usually removed at the time concrete is poured, so some accessories should be modeled as temporary objects. These types of accessory are not necessarily to be geometrically modeled, but their quantity and placed rebar should be noted and represented in the system.

- In the construction stage, pour breaks or stops may be changed due to various reasons: weather, worker, resources, managerial issues, conflict with other jobs and so on. The system should automatically adjust layout design – primarily placement of splices - and the corresponding rebar schedules based on changes made to the location of a pour stop. The changes should be reflected automatically and immediately, so that the field engineer can get an immediate update and can order new rebar and work specification for the workers. These requirements are expressed in the pour breaks/pour stops section of chapter 4.

3.5.1.1. Engineering

1) General

Standard shapes shall be available to use in the system in accordance to standard U.S. rebar availability and shapes used. As shown in Figure 56, CRSI and recommended industry practice documents. The standard sizes shall be available to pick from for all tools for quick and easy use. This shall also be intuitive and easy to customize if needed. The rebar shall be stored in the system containing the stock lengths available, hooks, bends, spice length, bend diameter, strength, weight, diameter, area, size, grade, and name. These shall be available and used by all rebar tools, drawings, and reports. There shall also be the ability to add any other information to each rebar in the system both on a piece level and a material level.

2) Checking for proper rebar layouts

Tools are needed to verify rebar layouts to ensure that they fulfill the requirements of the standard design code for rebar detailing.

This requires two aspects:

- Semantic identities for pieces and rebars. Each structural object must be identified as belonging to a specific class of structural object, so that it can have the appropriate rule sets applied to it. Similarly, the system should be able to identify the presence and purpose of each rebar – shear, torsion, tension, etc. - so that all the appropriate rebars can be taken into account within a rule check.
- Rule checking capability.

3) Automated creation of rebar and filler layouts for joist, waffle, and pan slabs.

These are object types that are not catered for in the existing software. There shall be tools that enable the user to productively create drop panels, column capitals, waffle slabs, two way slabs, and other typical cast in place floor systems (note – drop panels and column capitals are part of the column geometry and are described in the section 'Geometry manipulation' in chapter 4). This should be controlled both on a high level as a floor system and able to be customized on an object level. The specific tools that reinforce these type elements have to be parametric and tied into the floor systems. As the floor system elements adjust the rebar detailing must change to suite.

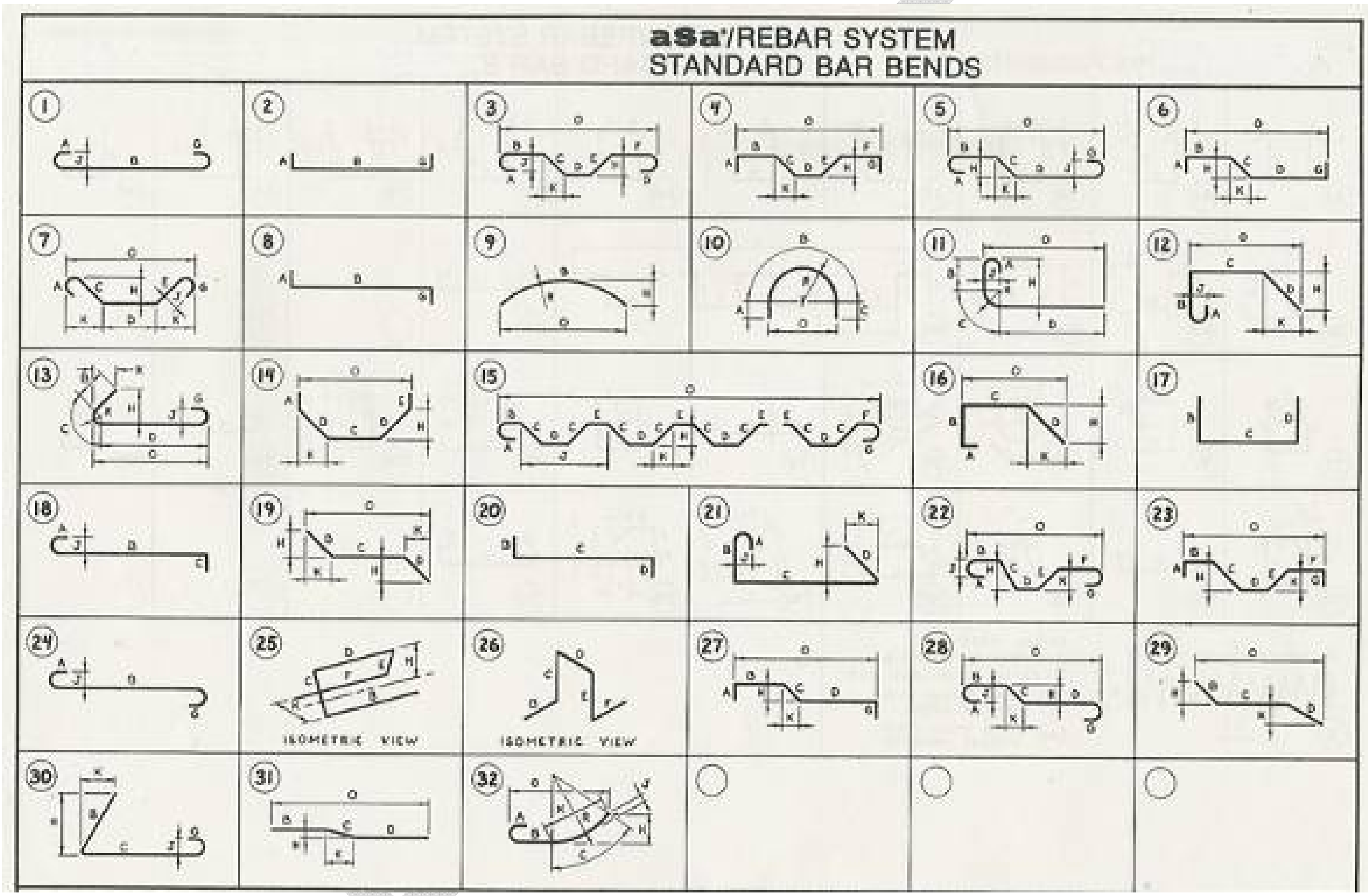


Figure 56. Standard rebar bends

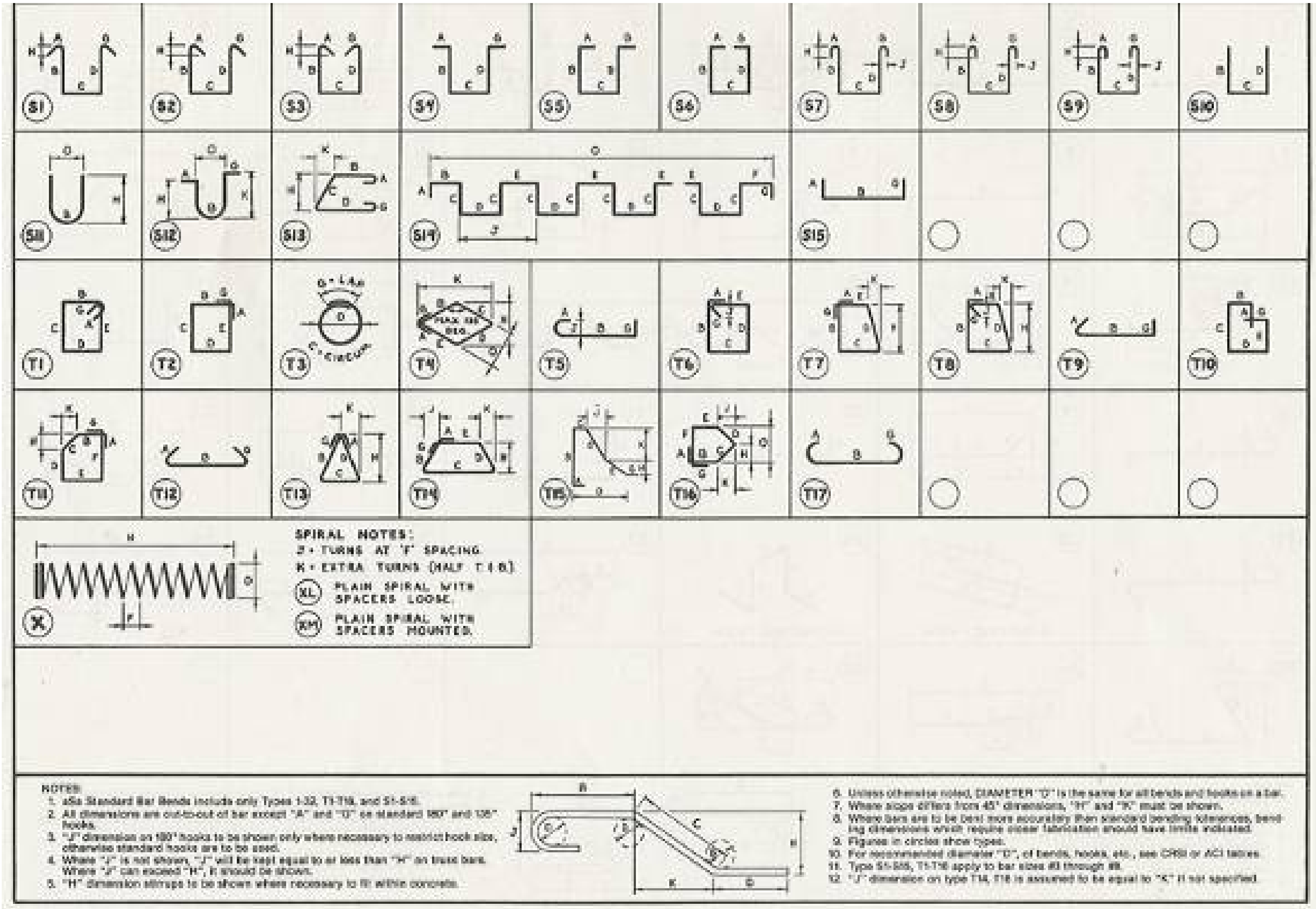


Figure 56. Standard rebar bends (continued)

3.5.1.2. Contracting

1) 3d rebar array

There is needed a tool for creating 3d reinforcement. It is desired to be able to create groups of rebar that are parametric in two directions with the cross section of an element but have user input. There shall also be ability to define the length of the bars that is parametric but not determined by the elements in the model; instead it shall be user input. This tool shall be used for both stirrups and rebar of any custom definable shape. This tool should be a basic level tool for custom, nonstandard parts. There shall be higher level tools that use this tool with more functionality for reinforcing typical cast in place objects.

2) Multilayer rebar group

It shall be possible to create layers of rebar in any concrete object. This functionality should be supported in all high level tools as well as low level tools. This includes mesh and should not limit number of rebar in objects. This should allow for both area driven and number of bars at a spacing driven functionality. Tools should not assume symmetry or any set number of bars throughout cross section.

3) Coverage/Anchorage

All reinforcement tools shall allow parametric behavior to boundaries of functional objects plus a user input parameter for both coverage and anchorage.

4) Splicing/Overlapping

It should be easy and intuitive to supply rebar material available length and have the reinforcement tools use this length, unless overridden, to put splices in rebar elements automatically. It shall be possible to edit on the object level to adjust the splices in place, to add, or take away splices using a lower level splice tool. All cast-in-place reinforcement tools should have this functionality built in to put splices where required holding to offset splices requirements in code. Rebar needs to be able to use splice accessories or welding. Three kinds of splicing types, which are lap splices, mechanical splices and welded splices, should be supported.

- Lap Splices: A lap is when two pieces of rebar are overlapped to create a continuous line of rebar. The length of the lap varies depending on a number of things, including the concrete strength, the rebar grade, size, and spacing. CRSI's "Reinforcement Anchorage and Splices" includes a compilation of required lap splice lengths based on these variables. Contact splices--in which the bars touch and are wired together--are preferred because they are more secure against displacement. Non-contact lap-spliced bars should not be spaced too far apart (see Figure 57).

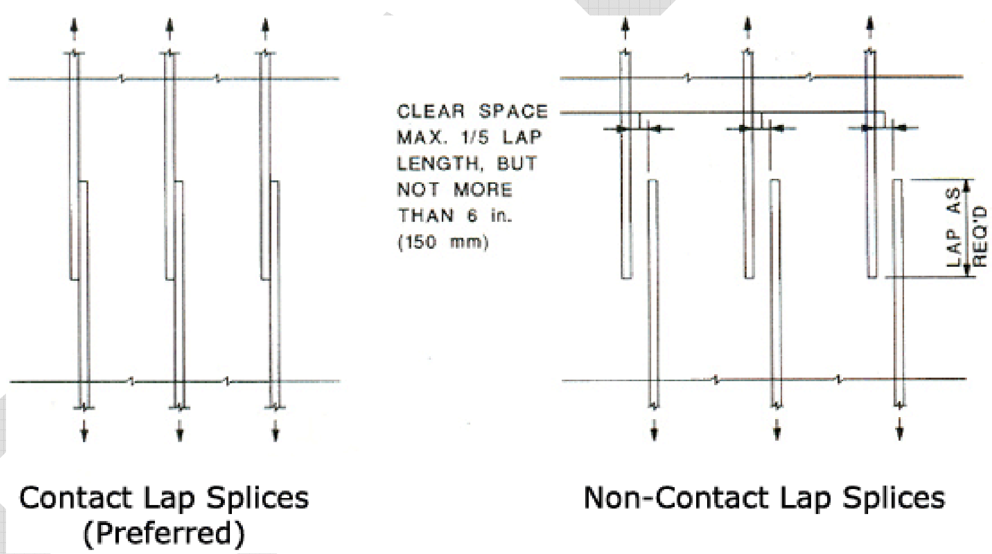


Figure 57. Lap spices

- Mechanical Splices: There are three basic types of mechanical splices: "tension-compression," which can resist both tensile and compressive forces, "compression only," also known as the "end-bearing" mechanical splice, and "tension only". They are often used when there is not enough room for manual splicing. The user interface should be intelligent enough to realize when this is the case and offer a choice of mechanical splices, although this capability is subject to first solving the soft clash identification problem, as defined in section 3.5.4 below. The design of mechanical splices is proprietary. The splices are supplied by a number of manufacturers across the U.S. (see Figure 58).

Figure 58. Mechanical splices

- Welded Splices: In general, CRSI recommends against manual arc welding in the field. However, if necessary, field-welded splices are accomplished by electric arc welding the butts of the reinforcing bar together. Welding should conform to ANSI/AWS D1.4-92, "*Structural Welding Code Ð Reinforcing Steel*" of the American Welding Society.

Splices should not clash with existing rebar. Welding involving special sleeves needs to be shown in the geometry model, as they tend to affect spacing between rebar and covering depth of concrete. The amount of splice overlap should be in both the material information of the rebar and the tool for setting minimum overlap.

5) Clashes and redundant placements

Reinforcement tools should take into account all reinforcement inside the concrete element to avoid clashes within the same element at least, and ideally in intersecting elements as well. Primary structural rebar should drive placement of lower level rebar based upon clearances. Reinforcement tools should also work together to avoid redundant placement of rebar. Stirrups, primary, secondary, and mesh should all work together and be parametric, none should clash.

6) Continuous reinforcement

All cast in place reinforcement tools should allow users to define elements using input, because cast in place elements do not have clear boundaries that are identical to the boundaries of model element. Thus user input may sometime be required to determine the volume to reinforce. For example, beam reinforcement will go partially into a slab and/or column region. This must be taken into account when creating a beam reinforcing tool as well as all other tools. Reinforcement must be parametric to slab, beam, and column geometry changes and placement. Tools must consider various elements and not only the geometry of one single functional object.

This requirement is largely dealt with by the logic proposed for dealing with the intersecting volumes of concrete elements – see section "Intersecting volumes" in chapter 4.

3.5.1.3. Installation

1) Pre assembly

It shall be possible to create reports and diagrams of rebar cages. It must be possible to assign bars to a certain rebar cage and to manage these aggregations of rebar separately from the concrete elements to which they belong.

2) Rebar accessories

It shall be possible to keep track quantitatively of rebar accessories associated with rebar material such as support chairs, rebar caps, terminators, splice objects, and other such rebar placing accessories.

3.5.2. Post tensioning

General	• Representing elements before and after tensioning • Comparative analysis to conventional • Materials required • Field Inspection • Documentation
Conceptual Design	-
Engineering	• Interface to specialized analysis software • Representation of deflection/displacement
Contracting	• Measure curved/arched cable lengths
Pre Const./Detailing	• Sheath objects • Strand object • Anchorage hardware at beam end
Installation	• Bearing Plates and Trumpet • Tendon Sheaths • Tendon Installation • Tensioning of Tendons • Grout Injection • Grease Injection

3.5.2.1. General

Figure 59 shows a typical situation where post-tensioning cables are anchored near the intersection of a number of beams (see Figure 60 and Figure 61).

A CIP object changes its geometric shape when post tensioning is applied. The system should enable tracking of the deformed shape after application of post tensioning using property fields that can express the deformations (as deltas from the cast shape), so that changes of deviation can be monitored. There seems to be no apparent need to actually model the deformations as geometry.

Where necessary, for example for a floor slab where displacements need to be monitored across a grid, a similar approach to that proposed for storing and displaying as-built data collected using laser scanning point clouds could be used – see section 'As-built' data in chapter 4.

Information required for structural analysis should also be provided for post tensioned elements. Analysis results including deflection and displacement should also be fed back from analysis package in the same manner with other analysis result.

Post tensioning uses sheath, strand (cable) and anchorage, these objects required for pre/post tension should be modeled in the system. Geometric representation for accessories and tensioning cables should also be supported in the system. These objects should also be considered in the process of soft/hard clash checking.

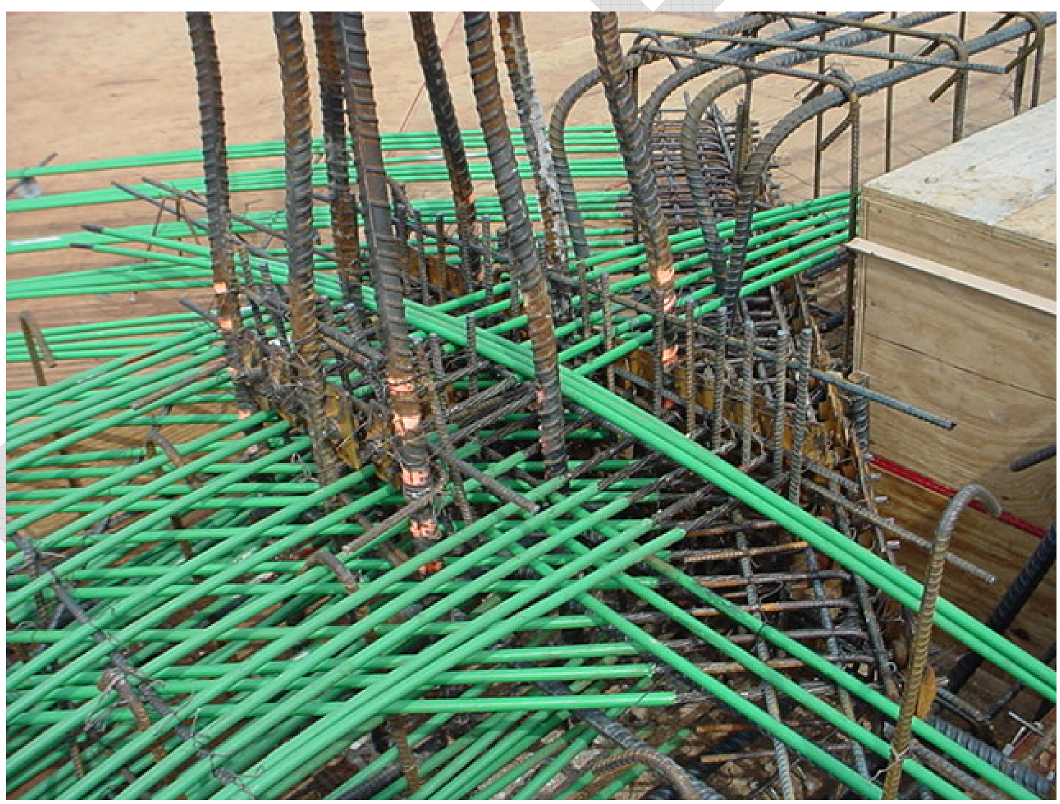


Figure 59. Post-tensioning cables in a floor slab at beam intersections

1) Description of Post Tensioning Works

- Installation of bearing plates, tendon sheaths, transition pieces (trumpet) of the tendon sheaths, vent pipes, drain pipes, and grout pipes.
- Transport of the strand coils from storage to the point of installation, assembly of pack frames, threading individual or a group of strands through the tendon sheaths, cutting the strands to the required lengths, and installing the anchor heads and assemblies at each end.
- Stressing the tendons, cutting strand tails and installing grout caps.
- Performing the field test of the grout mix and grouting the tendons with cement grout.
- Water test of vertical molded holes and repair of leakage prior to strand installation.
- Performing field friction tests as required by the Engineer.

2) Materials Required for Post-Tensioning

- Tendon sheaths including couplings and sheath plug.
- Bearing plates and trumpets.
- Anchorage systems and strand coils.
- Vents, drains and necessary valves or plugs, including injections and vent pipes and valves.
- Admixture for cement grout.
- Permanent grout caps for upper vertical tendon anchorages.
- Stand pipes for vertical tendon grouting.

3) Documentation required

The following are the data required for quality control purpose. It should be possible to carry these data within the model for the post-tensioning objects:

- Condition of the embedded sheaths before tendon installation.
- Date and time for each tendon installed, stressed, and grouted.
- Conditions of strand coils before pushing.
- Conditions of anchorage assemblies after they are installed at site.

- Stressing record of each tendon showing the required jacking force, ram/gauge ID number, pressure gauge reading, and the initial and final recorded elongation of the tendon at each end.
- Grout mix and grouting pressure of each tendon, grout volume, grout temperature and grout fluidity.

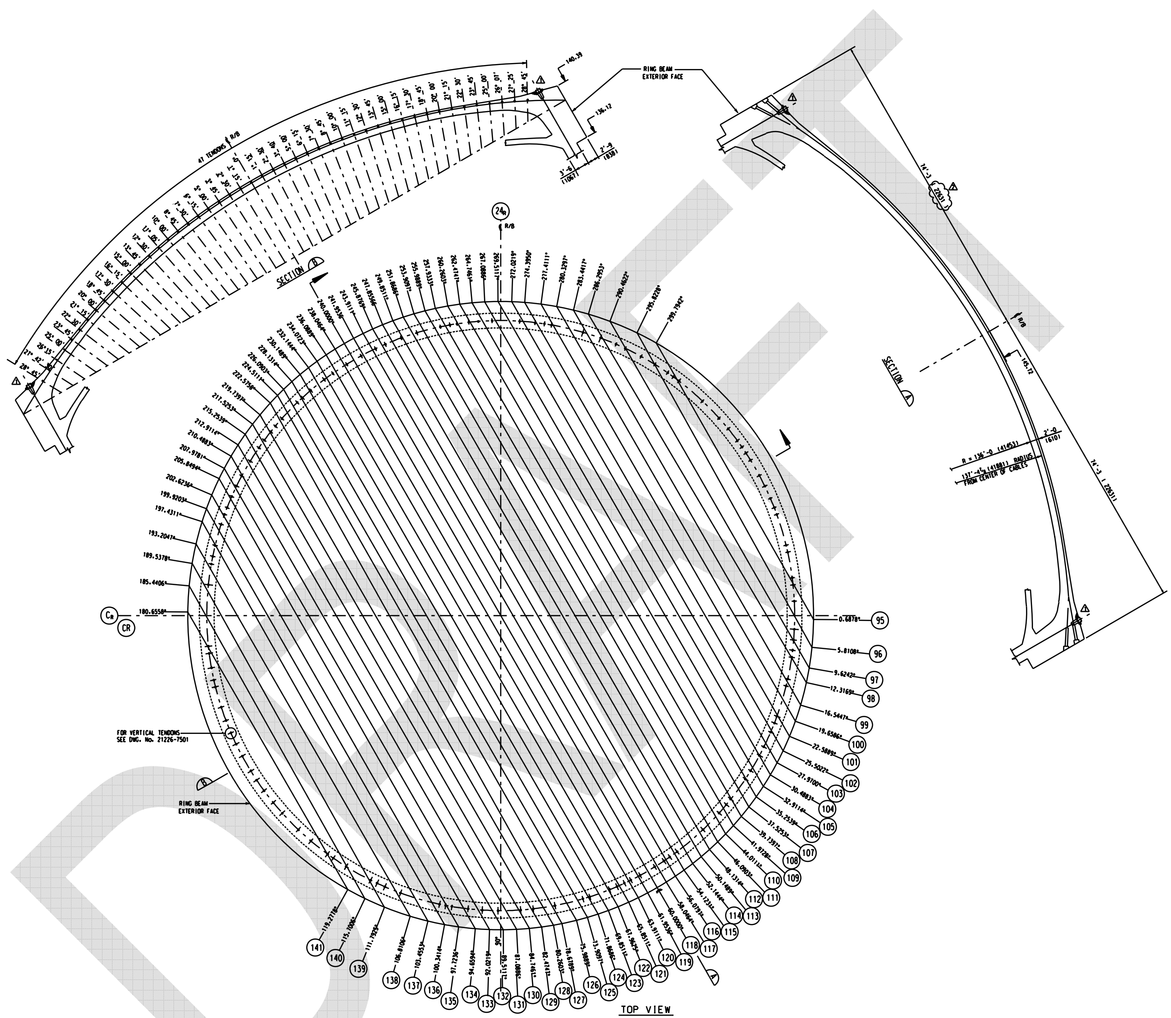


Figure 60. Dome post-tensioning (single layer)

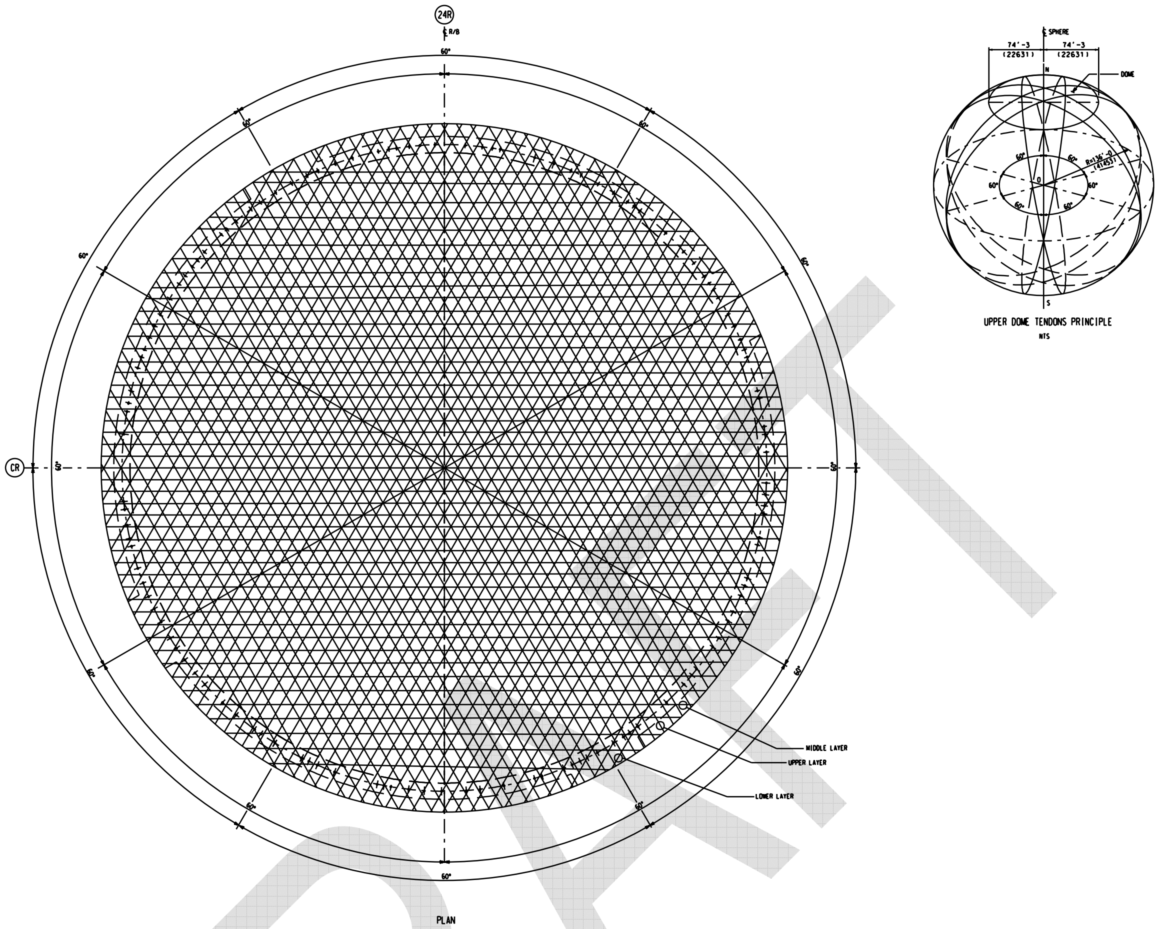


Figure 61. Dome post-tensioning in 3 Layers

3.5.2.2. Post-tensioned Slabs

In a typical workflow for an elevated post-tensioned slab, the structural engineer merely provides the desired forces and it is left to the post-tension supplier's engineer to determine the exact number of cables, their location and their associated structural analysis. In a typical workflow for a post-tensioned slab-on-grade, the structural engineer completes all of these work processes and offers the end user as well as the fabricator a finished product including all of the necessary information for fabrication and placement. The workflow for the structural engineer is shown in Figure 62. The variation in procedure can be seen at top right.

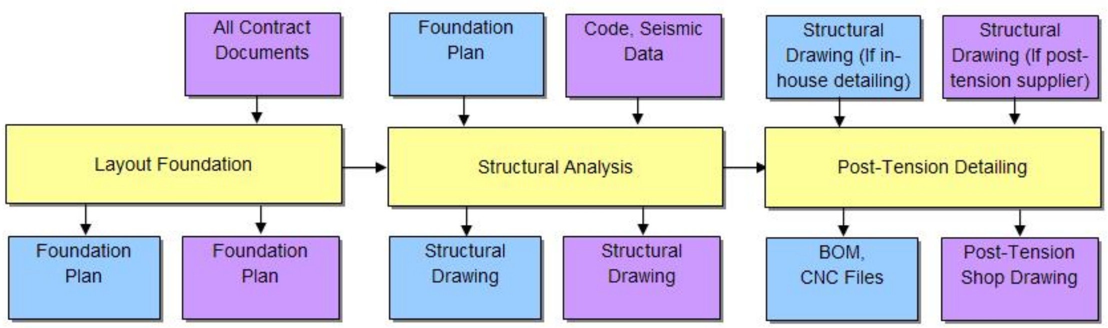


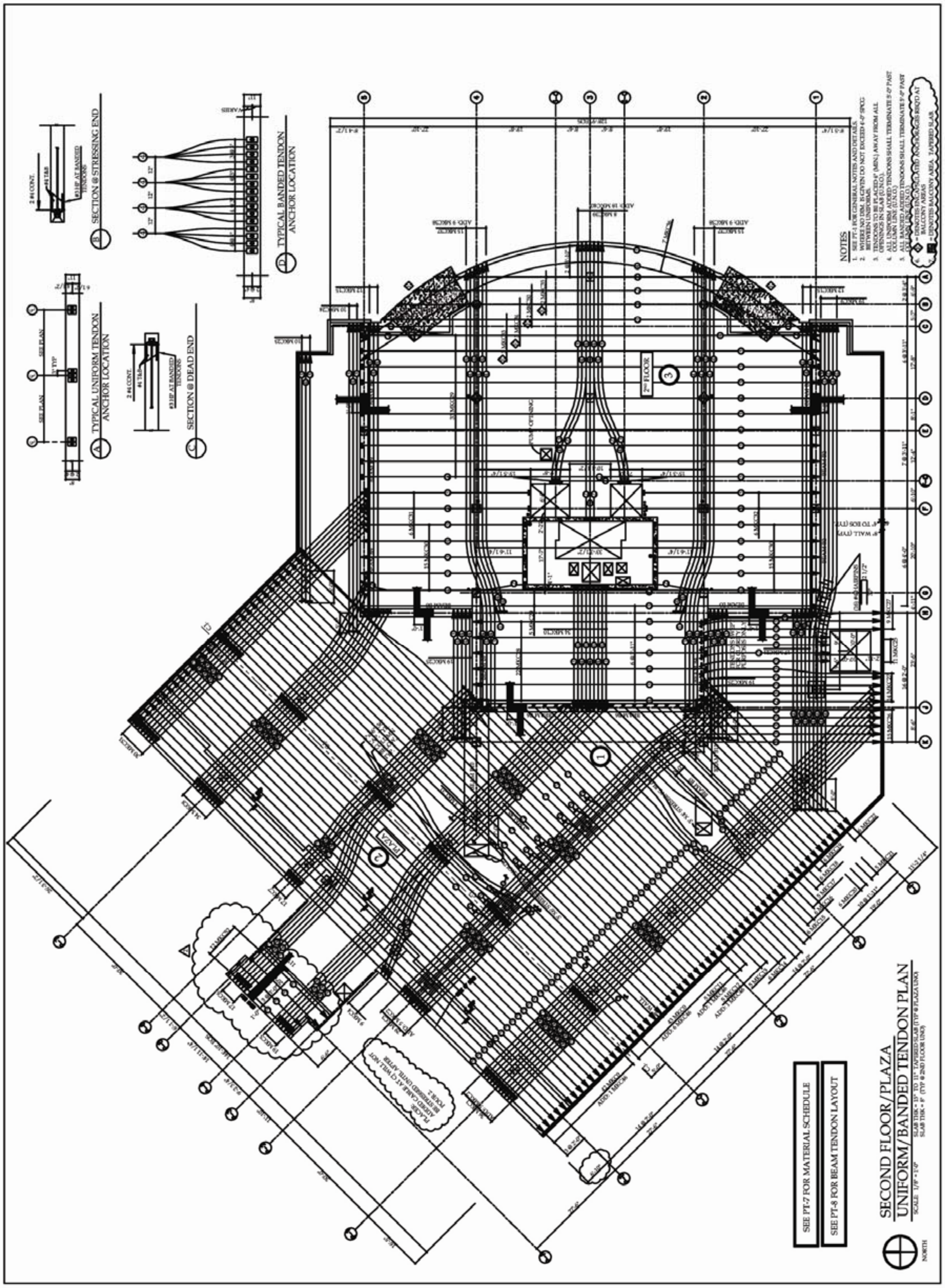
Figure 62. Structural engineering workflow for post-tensioned slabs

1) Industry Standards

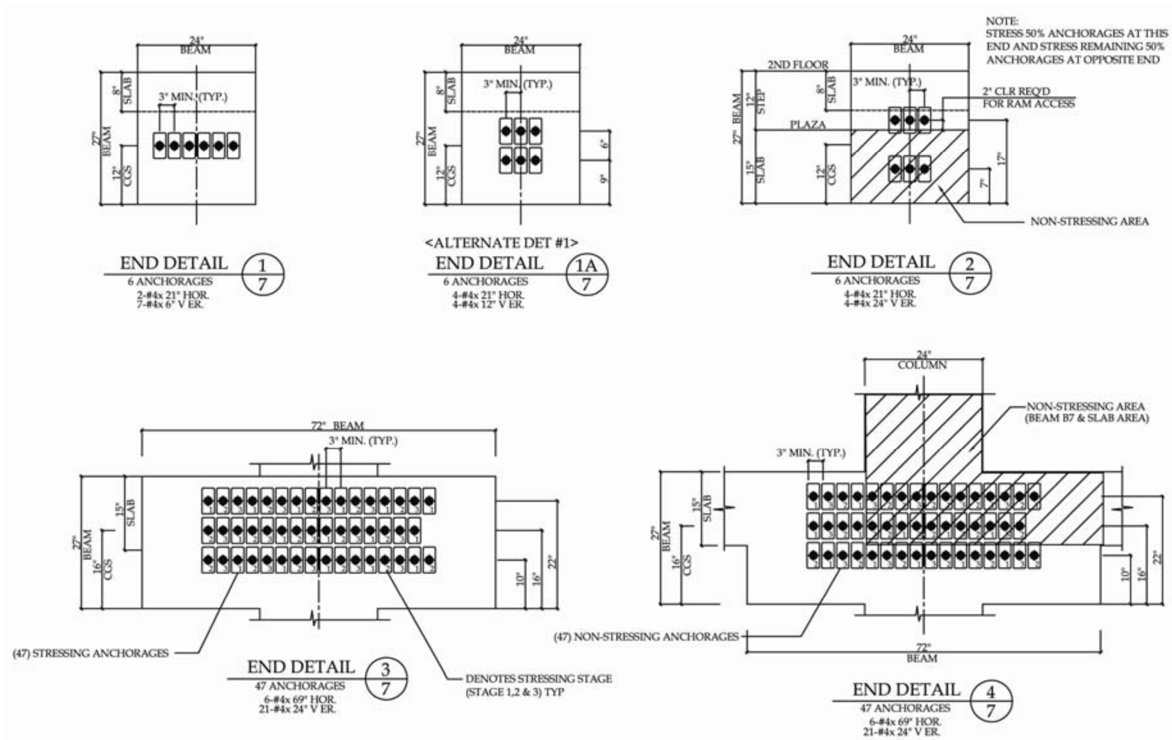
The standards for the post-tensioning industry are authored and maintained by the Post-Tensioning Institute (PTI) (see http://www.post-tensioning.org/). Like CRSI, PTI publishes standards for design, detailing and installation. PTI offers many publications centered on the design, installation, inspection and maintenance of post-tension systems. PTI also has extensive field training programs, complete with formal certification.

Many post-tension suppliers use the same or similar formats for their post-tension shop drawings. The approval process for post-tension shop drawings is very similar to that of reinforcing steel shop drawings. Figure 63 shows a number of details from different post-tensioned slab drawings, but for detailed specifications, example post-tension shop drawings will need to be collected across many product types to address the user's post-tension shop drawing needs. Listed below are the sheets commonly seen in a set of post-tensioning shop drawings.

- General Notes and Standard Details
- Banded Tendons
- Uniform Tendons
- Support Plan



(a) Sample view from post-tensioned slab drawings



(b) Detail of post-tensioning

Figure 63. Sample views from post-tensioned slab drawings

2) General Notes and Standard Details page(s)

The General Notes and Standard Details page(s) contains only those standard notes and details applicable to the respective project. These notes and details are often directly derived from PTI standards. In addition, some fabricators include their own standards on this page(s). Information included on this page could be the following:

- Standard cable sweeping
- Cable clear heights
- Support bar / cable configurations
- Anchorage details
- Stressing sequence and percentages
- Symbol legends
- Cable cutting procedures

3) Plan pages

Plan pages that include banded and uniform tendons could be separated or combined depending on the clarity of the plans. These pages contain information on how cables are to be placed in the 'X' and 'Y' directions. These pages also include information about the quantity of cables to be placed while not all cables are depicted. Only groups of cables are indicated. The number of cables in a group is indicated by a shape placed on the line that indicates a cable grouping; a triangle for three cables, a square for four cables and so on. Back-up bars are also usually depicted on these pages. Back-up reinforcing bars are placed to prevent bursting and shear forces that are present during and after cable stressing. These bars must be shown because these are bars that are not depicted in the reinforcing shop drawings.

Figure 64 shows a plan for a beam strip of post-tensioning cables. The support layout for the same beams in shown in Figure 66.

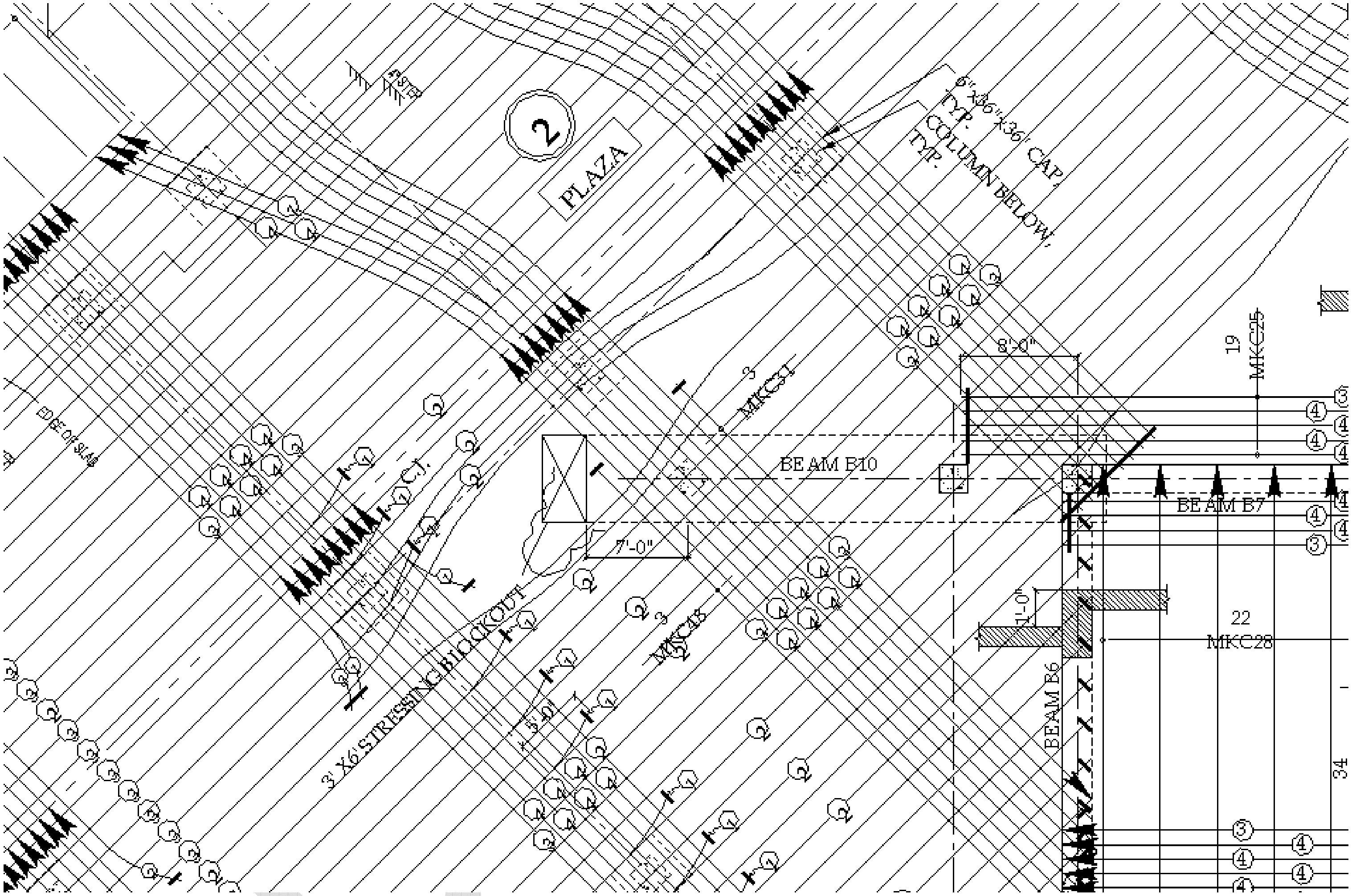


Figure 64. Post-tensioning cable layout drawing for beam cables

4) Support pages

Support pages show how the profile of each cable is to be achieved. Cables are supported by resting on short lengths of rebar, usually #3 or #4 bars that are 3' to 6' long. The support bars then rest on pre-manufactured supports like metal or plastic chairs. The sinusoidal profile is achieved by varying the height of the supports at each support bar location. This is clearly shown in the two figures Figure 65 and Figure 66, where the heights required for the cables are noted next to each support

5) Stressing log

Another key piece of documentation that is not included in the post-tension shop drawings is the stressing log. These are usually completed by the contractor placing and stressing the post-tension cable. This is a book that contains reports (usually spreadsheets) for each stressing area. Stressing areas and pour areas are almost always the same. This area could also be considered as the limits of a post-tensioning, reinforcing and concrete placement "work package." Each cable is listed individually. Included is the calculated elongation, which is provided by the post tension engineer and a minimum and maximum elongation which is based on the standard % tolerance set by the PTI or the engineer. A blank space is included adjacent to the calculated elongation for the entry of the actual elongation. After these elongations are recorded, the % deviation from the calculated elongation is entered and this report is submitted to the structural engineer of record (not the post-tension engineer) for approval. The structural engineer of record then issues a letter stating that the cable stresses are satisfactory and that the cables may be cut and pockets grouted. The workflow for this stage is shown in Figure 67 (Reference Stressing Approval Work Process).

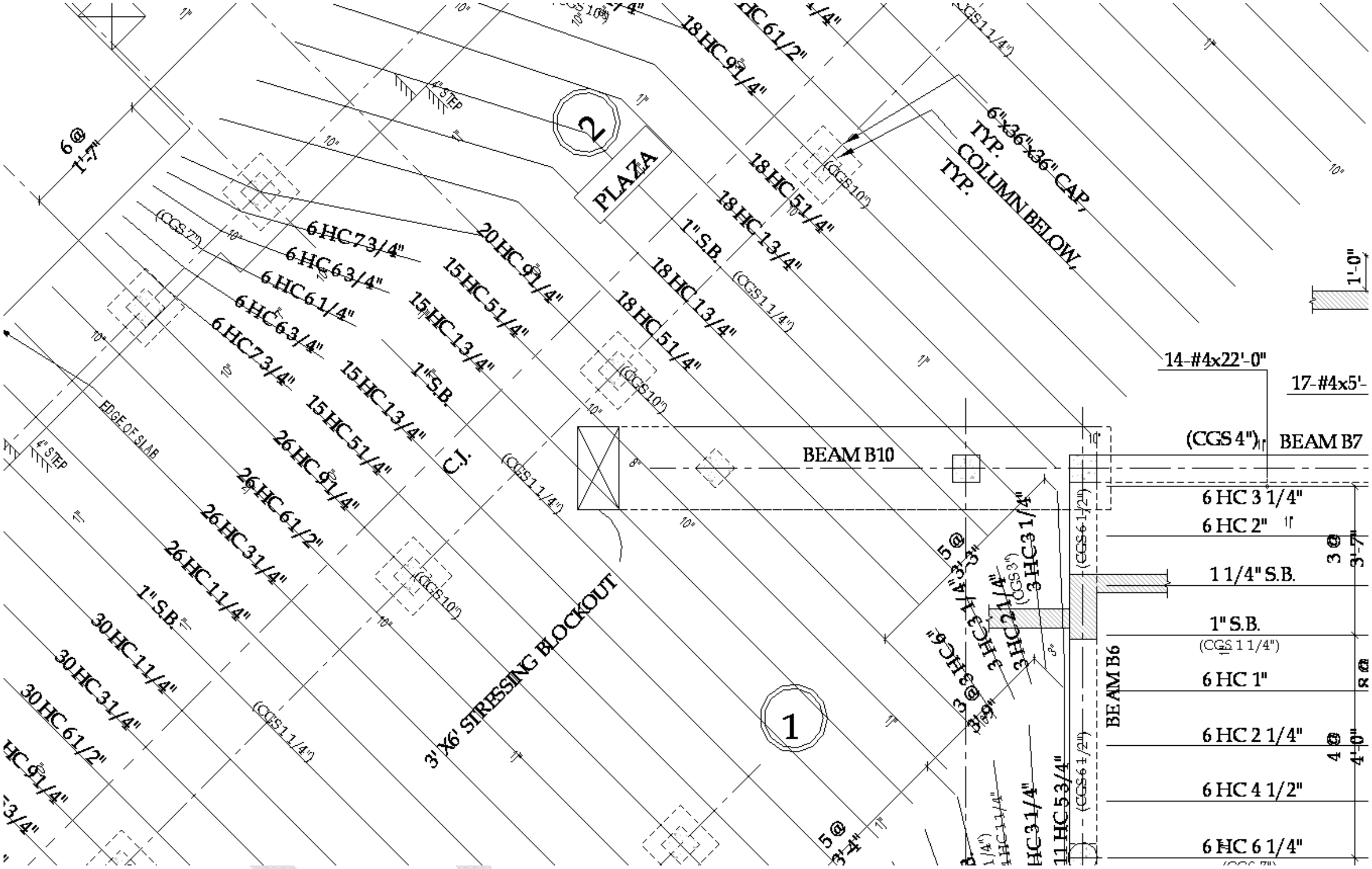


Figure 65. Post-tensioning support layout drawing for slab cables

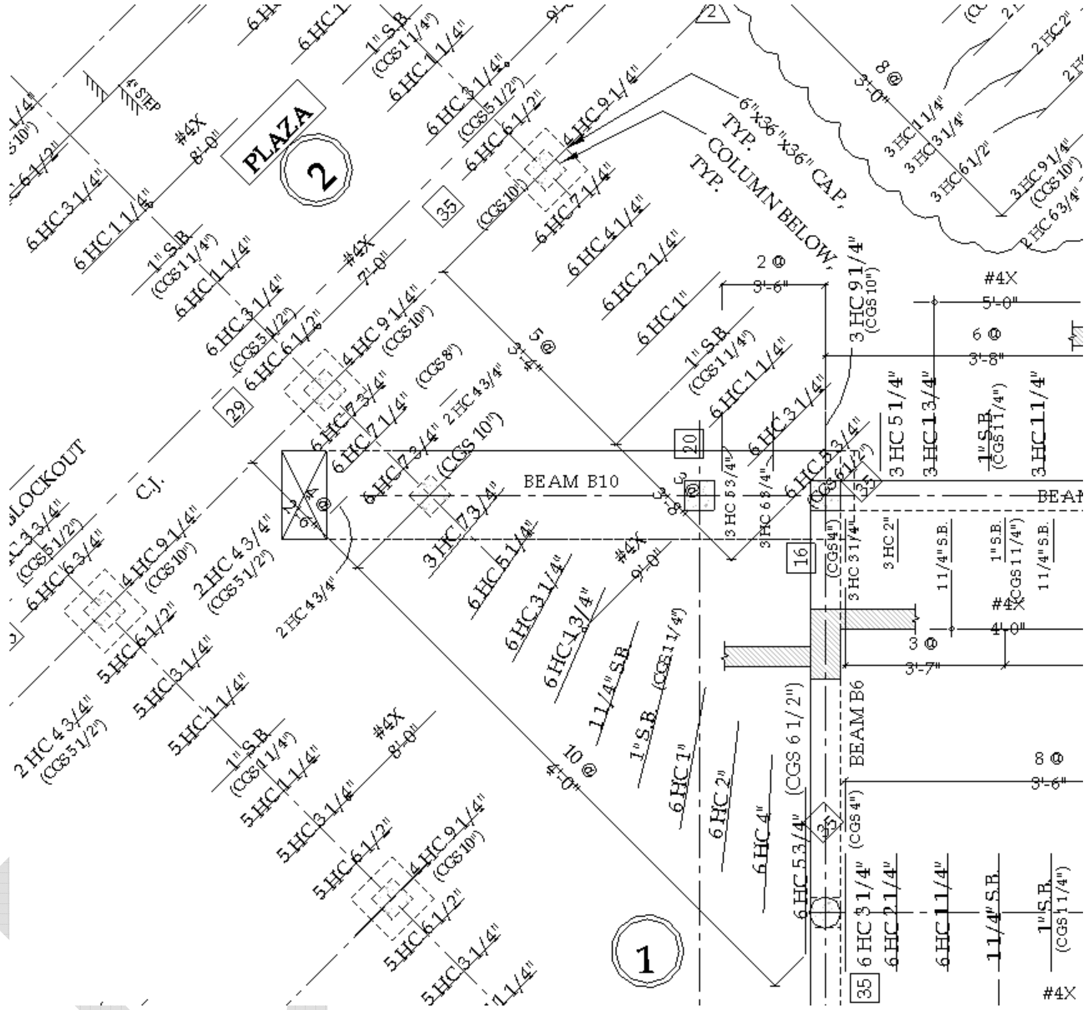


Figure 66. Post-tensioning support layout drawing for beam cables

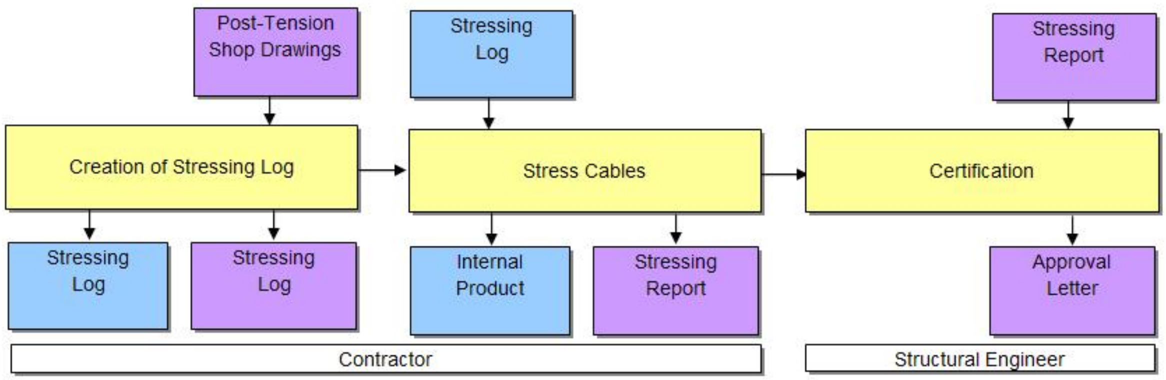


Figure 67. Post-tensioning approval process.

Barrier cables that are used to form railings at the edges of slabs are an often overlooked part of the post-tensioning industry. Barrier cables are stressed with the same equipment as post-tensioned cables in slabs, although the cables themselves are not sheathed in tubes and embedded in concrete, but rather galvanized and run freely between guides from anchor to anchor. Barrier cables are often used in parking structures as a less expensive alternative to concrete or masonry walls, or other more expensive traffic barriers. The PTI also sets the standards for barrier cables. PTI also offers training in barrier cable design and installation.

3.5.2.3. Functionalities

Key deliverables include:

- Post-tensioning shop drawings
- CNC Files / BOM
- Stressing log

Adapting the present methods for the generation of post-tension shop drawing will be quite a bit more difficult than that of reinforcing shop drawings. First, reinforcing steel shop drawings require no structural analysis. The creation of post-tension shop drawings in the system should be able to address all workflows; post-tension detailing with integrated structural analysis, with imported structural analysis and with no structural analysis.

The system should allow post-tension shop drawings to be easily extracted from the model. This will be challenging given that industry standard detailing show only cable groupings with their numbers being indicated by shapes. If all cables are shown in the model, the system must be able to recognize that a number of cables are part of a group, represent them in plan as a single line and place the appropriate shape on that line. The system should also be able to separate or combine pages of uniform and banded tendons at will.

The system should have the ability to generate the standard details and notes based on the properties of the model. For example, if there are no column capitals in the model, the standard detail for column capitals should not appear on the standard details and notes page. Likewise, standard details and notes should be taken into account, by the system, when tendons are placed. For example, if a cable grouping is swept around an opening, the system should know that the PTI standard for cable sweeps is 1:12 and alert the user or modify the model accordingly if this limit is violated.

The results of the structural analysis dictate the number, frequency, location and profile of each cable. The system should be able to populate all cables based on this structural analysis. The user should also be able to do this manually based on textual or verbal structural analysis information. Because the profile of each cable is derived from the structural analysis, the system must be able to place support bars and supports based on this analysis. The system should then be able to provide a support plan that is derived from the placement of support bars and supports in the model.

Creating a stressing log manually in MS Excel, while repeatedly referencing the post-tension shop drawings, is extremely tedious and prone to inaccuracy. The system should:

- Automatically generate stressing logs complete with cable layouts (with a unique piece mark for each cable that also represents the sequence of stressing) and subsequent information as described above.
- Associate the actual elongation of each tendon to its respective 3D model object.
- Enable visualization of these results as well as the interoperability of these results with structural analysis software.

- Enable the structural engineer to issue approval letters from inside the system and the issuance of the letters should be associated with their respective model objects.

The system should be able to provide CNC / BOM information for post-tension cable production and accounting. Every post-tension cable has a head and a tail and it must be represented so that the CNC / BOM information is correct. As with reinforcing steel, tags are an important part of the fabrication process. The system should be interoperable or contain within itself some method for printing cable tags. Like a bar list, post-tension suppliers use what are called bundle lists and master bundle lists. These are used to indicate what length and number cables are for a specific work package or pour. The cables are also spray-painted with various colors to indicate what bundle they are a part of and the system should be able to assign these colors by bundle. The system should also allow the user to visualize these bundle designations. The cable itself is not the only material that must be accounted for. Items like support bars, supports, anchors, pocket formers, wedges, etc. must also be accounted for. The system should allow for the complete modeling of these items and their inclusion in a BOM.

3.5.3. Connections

General	• Connection rebars must belong to elements and to connections simultaneously
Conceptual Design	-
Engineering	-
Contracting	-
Pre Const./Detailing	• Rebars extending outside of element geometry • Continuous elements rebar (e.g. negative moment reinforcement over supports in continuous beams and slabs) • Welds
Installation	-

3.5.3.1. General

There are two kinds of connections in cast-in-place concrete: connections between distinctly identifiable structural members (e.g. column to foundation, wall to slab, etc.) and connections between successive pours of concrete. The two connections sometimes coincide, but the second one may also be located within the volume of a single structural element. In both cases, connections are made by extending continuous rebars from one structural member into the other.

Rebar must be associated with both connection tools and with model elements. Rebar must belong to pour elements that are available for both drawing and reporting. There are not clear boundaries with model objects so more functionality is needed with cast in place connection detailing. This aspect is dealt with in the section 'Connecting volumes' in chapter 4.

For the second type of connection, allocation of the connection rebar is mostly affected by the pour stage, and proper association of connection rebar is required in the system. This requirement implies new functional requirements for pour breaks, which are detailed in the 'pour breaks' section of chapter 4. In a connection such as a pour break, top of column or top of wall rebars are extended up to proper connection length. Connection length could vary depending on the design specification. The structural specification may request placement of connections in two different locations. Half of the connections can be in different locations, or the connection location could be divided into multiple locations. Therefore, rebars must have the ability to be bound to more than one object and more than one user defined attribute.

For example, a wall dowel resides in both the footing supporting the wall and the wall itself. The wall dowel is constrained at the horizontal leg by the width of the footing (standard hook may not be possible). The wall dowel is constrained at the vertical leg by the lap requirement with the vertical wall steel and the depth of the footing. If the wall is moved, the dowel must also move provided that the proper binding between the wall and the footing is maintained. The same should apply if the footing is moved.

The system should also reflect aggregate size and minimum rebar spacing in locating connections, especially where rebar placement is dense. Information on connection rebars, where bars lie parallel to one another in lap splices, should also be reflected in the soft clash checking.

Some connections require special treatment of rebar, such as bending of rebar, welds, use of accessories. These special treatments should be represented in geometric representation and corresponding rebar schedule. Welding involving special sleeves needs to show up in the geometry model, as they tend to affect spacing between rebars and covering depth of concrete. Any object that affects spacing between rebars and covering depth should be monitored by the system and considered in checking soft clash.

3.5.3.2. Connection Tools

Connections need to be in place to handle typical cast in place elements and also work with rebar in components. Connection elements need to be affected by pour stop elements.

- Requires associatively with other objects
- Set of rules that can be developed to allow typical details to be applied to the detailing module. When the rules can't be satisfied, the detailer application "flags" these connections. An organization's "typical details" should be able

to be used to define the “standard” rules. See Section 3.5.5 for samples of typical details.

3.5.3.3. Connection Use Cases

1) Beam - Column - Slab

The connection shown in Figure 68 is a typical connection found in elevated cast-in-place concrete construction. While typical, different users and stakeholders look at the connection between these three elements in different ways (see Figure 70, Figure 71, Figure 72). For example, a structural engineer would likely view these as three distinct elements. A contractor might see these as two elements because a contractor would first pour the column then the beam and slab together. A steel detailer might look at this as three different elements but would have to make decisions about how to logically associate each piece of reinforcement with each element. Further, when the contractor is calculating the amount of concrete in the beam, it ends at the face that shares its vertical face with the outside face of column. However, when the contractor is calculating the length of the beam form that is needed to form the beam the measurement ends at the inside face of the column. This situation requires that the software be capable of representing the geometry in multiple views, where the volume of the connection is associated in different ways with the surrounding elements depending on the view type. This issue is discussed in detail in the 'Connecting volumes' section of chapter 4.

An example of connection classification for an element inside the concrete would be beam stirrups that reside in both the beam and the slab. However, the detailer would most likely only associate the stirrups with the beam object. The bottom horizontal beam steel is located in only the beam while the top horizontal beam steel is actually in the slab. However, the steel detailer would most likely associate the both the bottom and top beam steel with the beam. However, when pouring the concrete, there are often situations in which the contractor would elect to first pour that section of the beam that is below the slab level, and then complete the beam when pouring the slab itself. In that case, the top rebars of the beam should be delivered and installed together with the slab rebar, not with the beam rebar.

The system should have the capability to move freely between connection classification based on the user and the element. This could take the form of pre-defined or custom behaviors that the various users could use to apply to specific situations. The same user may also have different subsets of behaviors for drawings and for BOMs and the system must be able to address this.

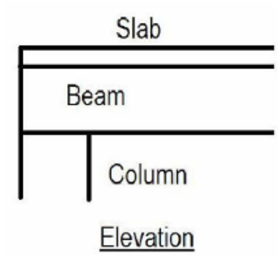


Figure 68. Column-beam-slab connection

2) Slab - Slab

Shown in Figure 69 is a slab to slab connection at a pour break. These pour breaks could have been placed because the contractor installing the slab-on-grade does not have sufficient labor to install the slab all at once or it is not feasible for some other reason. In this case, the contractor sees these as four separate slabs. However, the structural engineer sees these slabs as a single slab with the exception of the application of a standard detail to address the connection.

The system should have the ability to recognize the connection between the various slabs and apply the standard detail for this connection. For example, calculations for the associated keyway, smooth dowel rebar, etc. should be included in the BOM for each slab. Further, the system should relocate lap splices so that they do not occur at breaks in the slab. The system should also allow these connections to be applied at virtually any point in the workflow and by virtually any stakeholder/user. A structural engineer for a self-performing design-build general contractor may want the ability to "tinker" with pour breaks before the structural analysis is complete. The structural analysis should not be adversely affected by the fact that pour breaks have been included. Conversely, if the structural engineer wants to leave this work to the detailer so that it can be done long after the structural analysis has been completed, the system should be able to do this.

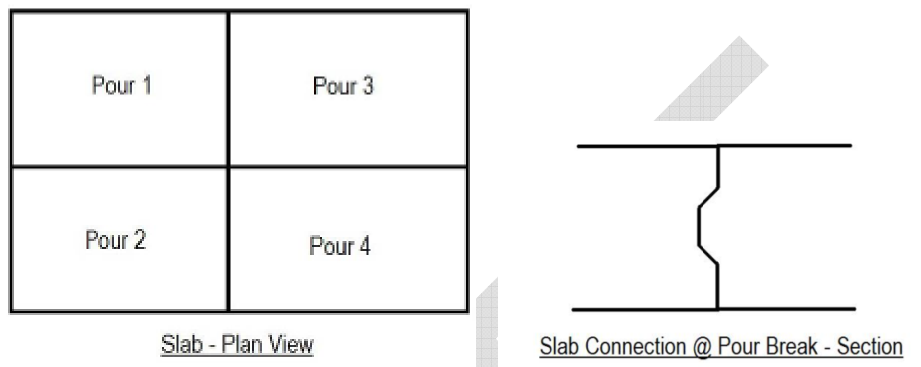


Figure 69. Slab-slab connection

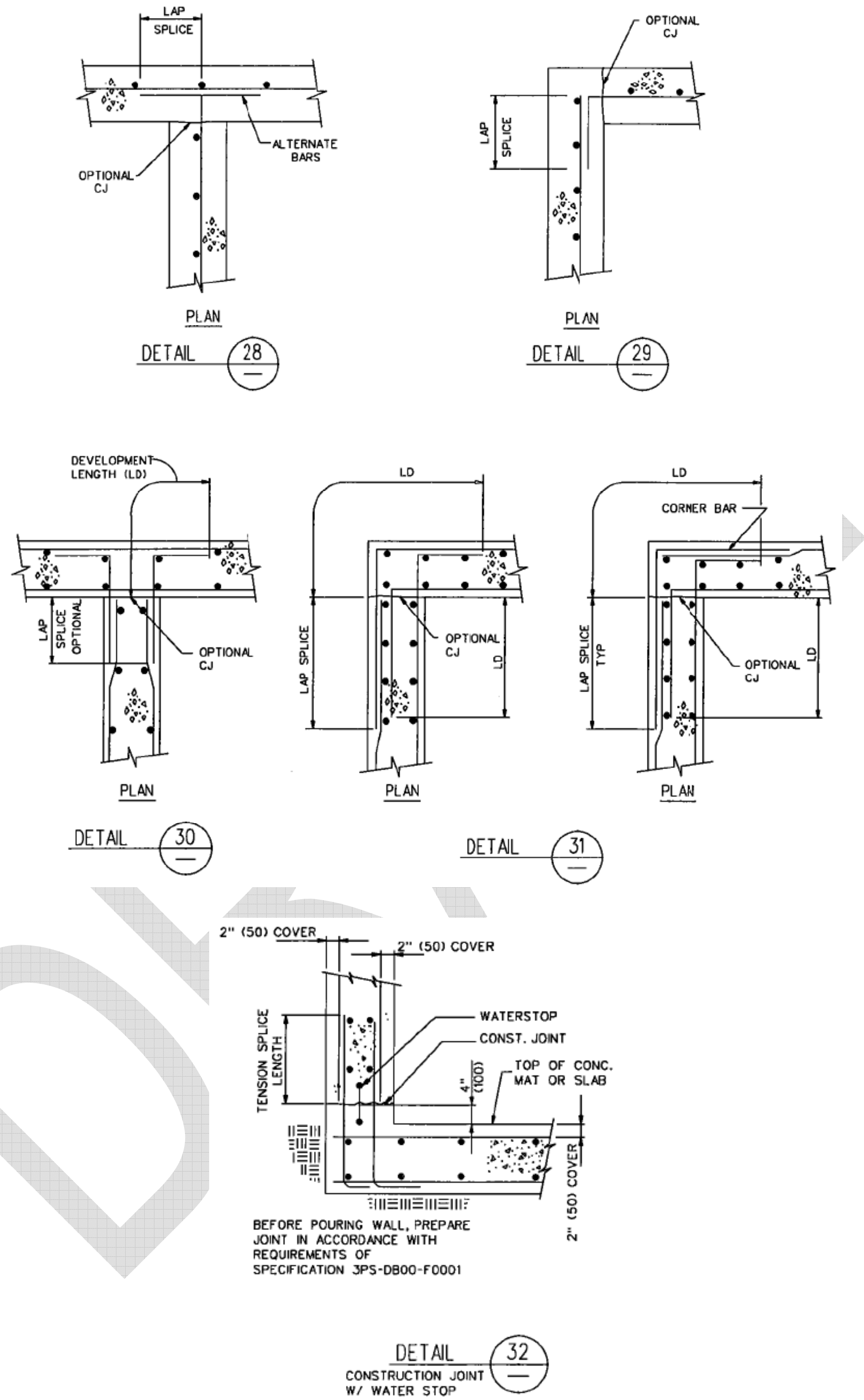


Figure 70. Reinforcing at wall intersections

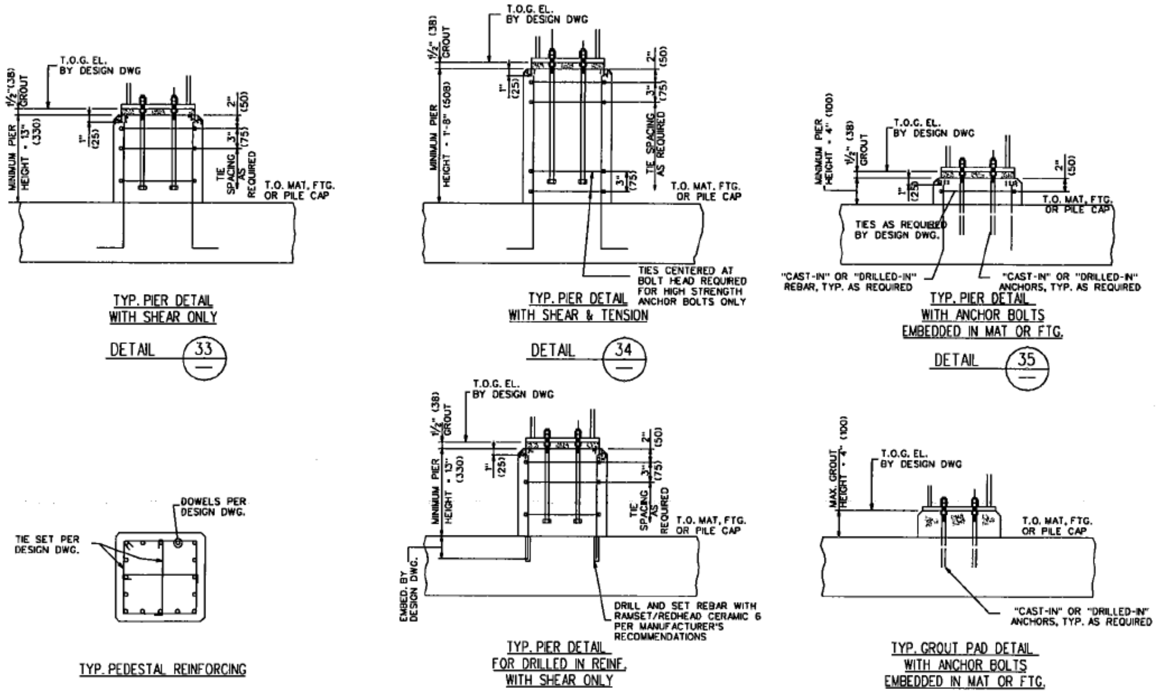


Figure 71. Typical CIP Pier to Steel Column Connection

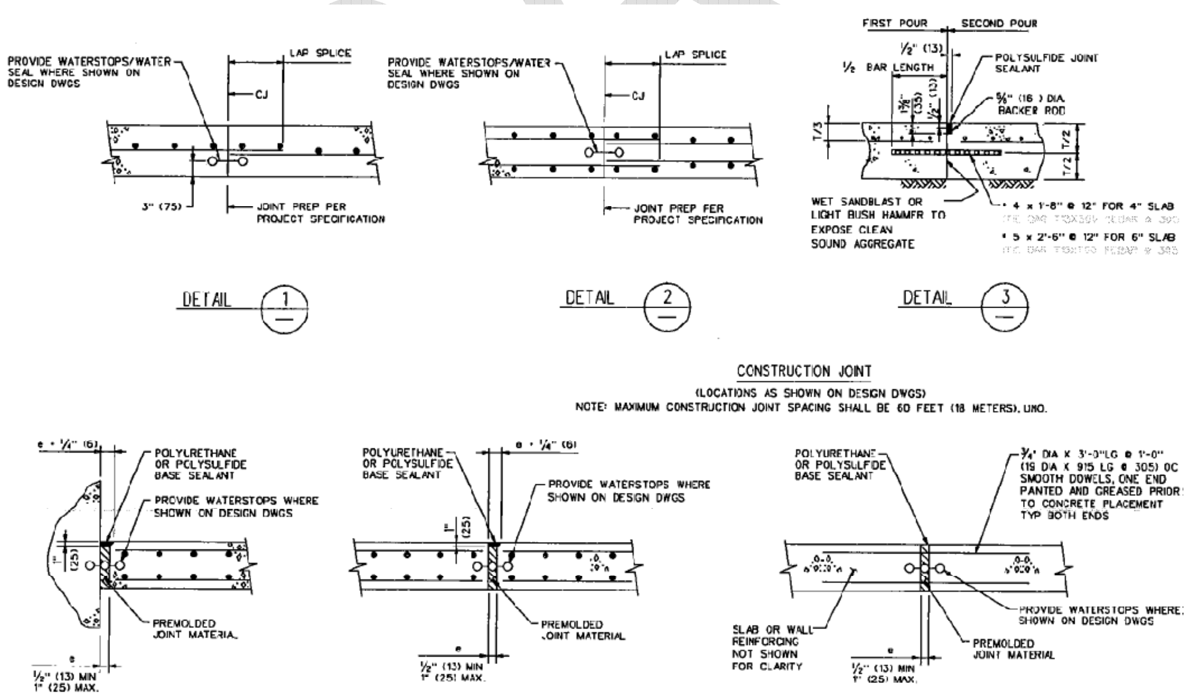


Figure 72. Typical Slab Construction Joint

3.5.4. Hard and soft clash checking

General	• Interface with other disciplines, other building systems (e.g. check whether a hole can be made in a concrete element for a pipe or other system object without interfering with major rebars in the element).
Conceptual Design	-
Engineering	• Checking for rebar congestion
Contracting	-
Pre Const./Detailing	• Constructability analyses: bar spacing vs. aggregate sizes, rat's nests • Checking for rebar congestion • Safety considerations: top mesh spacings, openings, rebar caps
Installation	-

The general cases of hard and soft (or clearance) clash checking were defined in detail in section 3.2.4 above. In this section, the focus is on those aspects of clash checking that relate directly to reinforcement. As described in earlier sections, rebar congestion checking is one of the most required functionalities. Rebar congestion should consider aggregate size, rebar diameter, required minimum covering, minimum spacing, embeds, sleeves, lapped splices, welded splices, mechanical splices and other objects that affect congestion.

In order to achieve reasonable performance the clashing might be broken down into subgroups as follows;

1) External Reference Objects

These objects should not need to be checked against rebar, only concrete volumes, steel, and precast. There might be occasional exceptions to this rule, but it should not be frequent enough to warrant a rule. Time would be better spent adapting a rule penetration feature for when plumbing or other external component penetrates the concrete volume.

2) Concrete objects

- **Piles -** Piles not only clash with the concrete, but may have tension connectors that clash with rebar. The Pile should be a known type that is not clash detected, except maybe to determine embedment depth. The Tension connector perhaps only getting an Embed soft clash rule.
- **Embeds -** Embeds should only receive soft clash consideration. This is because in most cases the bars can be squeezed around the embedded item. In cases where the rebar is inflexible, a smart embed feature should be available that seeks gaps in the rebar cage for anchorage.

3) Rebar to Rebar Groups

- **Longitudinal or Running Clash -** This clash routine should check for parallel runs of bar groups and register a hard clash with a tolerance of -1mm. Such groups may or may not actually work and probably should have their locations adjusted when a clash is found. This should be a complete check of all bars in the group.
- **Tie Bar groups vs. Longitudinal Bar Groups -** Tie groups should be checked against the longitudinal bar groups in columns and beams. Checking the first tie set in the tie group should be sufficient, registering a hard clash with a tolerance of -1 mm if found.
- **Traversing Clash -** This routine should first check by intersecting group ranges. Say typical column-beam-beam intersection. Assuming that there are bar groups forming the column and beam, this range intersection would define the volume for a complete bar vs. bar check (ties and long bars and hooks). This check would register as a hard clash with a tolerance of -1mm.
- **Single Bar Clash -** A single solo bar clash should not have to be considered if it will involve significant CPU time.
- **Rebar Density -** Rebar congestion should be identified by a density function. The two possible ways to define this function would be the number of rebar clashes per volume of concrete or the weight of rebar per volume of concrete.

A threshold density level may be developed that will issue a warning to the user. The threshold level should have a default value that can be adjusted by the user. Alternately a contour plot of density may be displayed. Again this display should have predefined contour levels that can be adjusted by the user.

When the clash runs are made, the stored gap data could form solids with sieve property ranges. For example, in a beam area, the smallest rectangle between bars might be 3"x11". In the traversing clash areas, a smaller sieve property would be expected, say 2"x3". Being able to dynamically display the volume area and report back the property would provide a basis for aggregate size.

3.5.5. Rebar schedule reports

General	-
Conceptual Design	-
Engineering	• Report schematic rebar requirements
Contracting	• Graphical reports
Pre Const./Detailing	• Hard coded shapes • CNC output for production
Installation	-

3.5.5.1. General

Rebar layout design should provide enough information to generate shop drawings and rebar schedules including main rebar and supplementary rebar. Both graphical and textual rebar schedules should be provided. They should show rebar length, size, grade, available size, strength, hook and extension information, bending radius, shape, bend dimensions, and number of bars. This information needs to be available from both the model and the drawings. Schedules should be included for typical construction and customizable for any configuration. All shapes should be included and recognized. It must be possible to define and handle custom shapes with system.

The standards for rebar shop drawings, in the US, are authored and maintained by an organization called the Concrete Reinforcing Steel Institute (CRSI) (http://www.crsi.org/). They offer many publications that outline these standards. Careful review of these publications and adherence to the standards set forth in them is essential to providing a solution that extracts widely accepted data (at least in the US) from a building information model. Drawing layout is just one of these standards.

The user should have the ability to capture standard requirements as defined by other organizations in the same way as CRSI requirements. It should be possible to set defaults for job requirement parameters, such as clear cover, at three levels:

1) At the project level

2) At the level of object types. For example, bottom cover for footings = 3", side cover for footings = 3", top cover for footings = 2". This is key information given that many of the rebar lengths are partly determined by these attributes.

3) At the level of individual objects.

The system should provide detail drawings for rebar details with quantities and should be able to provide g-code for CNC machines. The format of rebar schedule drawings should have catalogs of predefined and customizable graphical representation schemes, so that the user has control of the graphical representation depending on its use. G-code has some variations depending on the manufacturer of rebar fabrication CNC machines, so the system should have flexibility to incorporate variations of G-code.

3.5.5.2. Bar Lists

The user should have the ability to extract bar lists with little effort and a great deal of freedom. The user deals with many constraints when formulating a bar list; the project schedule, the dimensions of the load, the weight of the load, unforeseen schedule changes, etc. The user should be able to specify these constraints as they exist and formulate a bar list accordingly. The user should be able to create a bar list based on object selection including filtered grouping of objects, such as "ALL FOOTINGS, WALLS AND COLUMNS IN AREA B." The user should also be able to modify or split the bar list at will in the bar list view, which will be allow separate shipments to be made.

Concrete contractors are also interested in weight by cost center. There are also extensive CRSI standards with respect to bar lists that must be followed.

3.5.5.3. Shop drawings

The system should allow for easy placement of text and schedules on drawings. Rebar data call-outs are just one example of this. Call-out placement must be automated and meet CRSI standards. The user should be able to toggle between structural engineering and detailing style call-outs on-the-fly or at the outset. Other items like bend chart, bend diagrams, title block and page border must also meet CRSI standards and allow for automated placement.

Reinforcing steel shop drawings have several key components that are almost always present on every sheet:

- Title Block and Border
- Bend Chart
- Bend Diagrams
- Lap Splice Chart
- Reference Inventory
- Drawing
- Call-Outs

The bend chart contains information about all bent bars that are called out on the sheet that it appears. All bent bars are piece marked starting with a letter designation and ending with a three or four digit number. The letter indicates the first bent bar grouping. Bent bar groupings are usually divided on a detailing page by page basis. Most detailing software begins this designation with the letter 'A'. The first number in the numerical portion of the piece mark indicates the bar's size. The last two or three numbers are used sequentially regardless of the size. The sequential portion of this piece mark can never be reused in the event that the bend is modified or this particular bar type is removed. Example:

Bar Mark = A401 = A bent bar that is in the first bent bar grouping, #4 size rebar, and is the first bent bar in the first bent bar grouping.

Straight bars almost never appear on this list, hence the name "Bend Chart". There are a few exceptions. Some government agencies and departments of transportation require that all bars have a bar mark. Thus, all bars straight or bent must go on the Bend Chart.

The Bend Chart also contains other pieces of information. The bend type indicates the shape that each bar in the bend chart is to be bent. These bend types are indicated by numbers or numbers and letters and have been standardized by CRSI. Each segment of the bent bar has its own letter designation. These are also shown in the Standard Bend Diagrams. The length of each of these segments is listed in the Bend Chart for fabrication and validation purposes. These lengths are often verified during the shop drawing review process. Bend diagrams for each bent bar are also shown on each page of the reinforcing shop drawings. Bend diagrams for the bends used on that page should appear on that page. This may seem redundant but it is difficult to memorize all bend types and their letter designations and reference materials are not always readily available in jobsite environments.

Lap splice charts, for only those lap splices that are applicable, are also included on each page of the reinforcing shop drawings. The lap splices to be used are determined by the structural engineer and as a result can vary from project to project. The old method for lap splice determination was bar diameters lengths. These lengths were specifically called out by the structural engineer for every condition. For example:

Lap Footing Bottom Bars 24 bar diameters: If the footing bar is a #4, which is ½" thick, then the bars should be lapped one foot. This could be used to develop a lap splice chart for all bar diameters used in the bottom of footings.

Bar Size	#4	#5	#6	#7	#8	#9	#10
Lap (24D) (inches)	12	15	18	21	24	27	30

This method, while still requested by some structural engineers in some locations, is now rarely used. The new method, initially published by the American Concrete Institute

(ACI), is much more complex. The structural engineer prescribes the concrete mix strength and a lap splice class and the detailer must refer to the lap splice chart provided by CRSI.

Many of the "intelligent" 2D tools allow for reference call-outs. Rather than calling out identical objects repeatedly, an alphanumeric designation can be assigned and placed adjacent to each instance of this like call out. For example, if all F1 footings contain 16 #8's 8'-6" long, this only needs to be called out once but it must be listed in a reference inventory schedule or chart that indicates the call-out, the quantity and the designation.

Most obvious of all are drawings. Reinforcing drawings are straightforward but are different from structural drawings in the way that they are presented. Structural engineers tend to present a wider range of information, especially in plan view, than reinforcing steel detailers. For example, in most structural drawings, the lines that represent the location of footings are often hidden lines. This is because the structural engineer is representing not only the footings on this page but walls, slabs-on-grade, columns, etc. The structural engineer is often also representing the finished product. On the other hand, reinforcing steel detailers tend to separate detailing drawings by cast element type; footings for a given area on their own page, walls for that same area on their own page, and so on. Also, because the reinforcing steel detailer is providing drawings to a user that is seeing an unfinished product (meaning dirt is not covering the footings) the lines representing the location of the footing are solid rather than hidden/dashed. This is similar to setting line types like in 'reflected' views.

Call-outs are another key part of any reinforcing shop drawing. They indicate to the reinforcing shop drawing user what size bar, what bar (if bent), how many, where and how they are to be placed as a supplement to the graphical information already presented. The standards employed for these call-outs are also set forth by CRSI. For example:

Straight Bar

- 10 #4 10-06 @12" OC T&B

Which means:

Quantity	bar size	bar length	spacing	location
10	#4	10-06	12" On-Center	Top and Bottom

Bent Bar

10 4A01 @12" OC

These are just two call-out examples but there are many more. There are also many other abbreviations like those shown above. Another difference between structural engineering drawings and reinforcing steel drawings that should be highlighted is dimension style. Structural engineers tend to use an architectural style whereas reinforcing steel detailers use a different style. For example:

Reinforcing Steel Detailing Style: 10-062

Equivalent Architectural Style: 10'-6 1/2"

3.5.5.4. CNC

Export functionality to standard rebar bending machines is required. User should also have ability to create custom export to future and non-standard machines.

3.5.5.5. Pre Construction/Detailing

Rebar detailing must not be exclusive to model object geometry. Rebar from one element must be easily extended partially into other elements and be parametric. For example, when connecting a beam to a column some of the longitudinal bars extend into the column a certain distance, while others may not. Examples of how traditional rebar schedules and details are shown on drawings for slabs, beams, columns, and walls. Refer to bookmarks in this section for sample details and schedules of the following (see Figure 73 - Figure 75):

- Reinforced Concrete Column
- Reinforced Concrete Shear Wall and Link Beam
- Reinforced Concrete Spandrel Beam (in a high-rise framed tube)
- Two-Way Reinforced Concrete Slab System

- One-Way Reinforced Concrete Slab System
- Reinforced Metal Deck Slab Systems
- Belled Caisson Foundation System

REINFORCED METAL DECK SLAB SCHEDULE

MARK	T1 (IN.)	T (IN.)	SUPERIMPOSED LOAD (PSF)		MAIN REINFORCING						TEMP. REINF.	CONCRETE		HANGING LOAD TYPE (SEE NOTE 2)	REMARKS
					BOTH ENDS CONTINUOUS		ONE END DISCONTINUOUS		BOTH ENDS DISCONTINUOUS						
			LIVE LOAD	DEAD LOAD	BOT.	TOP	BOT.	TOP	BOT.	TOP		WEIGHT PCF	F'C (PSI)		
RS1	2	6	130	120	#6@12"	#6@12"	#6@12"	#7@12"	#7@12"	#6@12"	#5@12"	145	4000	C	LEVEL 17 OFFICE FILE STORAGE
RS2	2	3 1/2	100	15	#4@12"	#4@18"	#4@12"	#4@12"	#4@12"	#4@18"	#4@18"	145	4000	B	LOWER LEVEL 1 SEE NOTE 7
RS3	2	4	250	10	#5@12"	#5@12"	#5@12"	#5@12"	#5@12"	#5@12"	#4@16"	145	4000	B	LOADING DOCK RAISED PLATFORM SEE NOTE 7
RS4	2	5	250	65	#5@12"	#5@12"	#5@12"	#5@12"	#5@12"	#5@12"	#4@12"	145	4000	B	LOADING DOCK (HS20-44 AASHTO LOADING)
RS5	2	8	175	60	#6@12"	#6@12"	#6@12"	#6@12"	#6@12"	#6@12"	#5@12"	145	4000	B	LEVEL 16 MECHANICAL SEE NOTE 6
RS6	2	6	50	120	#5@12"	#5@12"	#5@12"	#6@12"	#6@12"	#5@12"	#5@12"	145	4000	C	LEVEL 17 OFFICE SEE NOTE 6
RS7, RS7A	3	6	50	30	#4@12"	#4@12"	#4@12"	#4@12"	#4@12"	#4@12"	#4@12"	145	4000	B	LEVEL 2 OFFICE ABOVE LOADING DOCK SEE NOTE 5
RS8	2	6	150	80	#5@12"	#5@12"	#5@12"	#5@12"	#5@12"	#5@12"	#4@12"	145	4000	B	LOWER LEVEL 1 MECH. LEVEL 2 MECH.
RS9	2	6	250	50	#6@12"	#6@12"	#6@12"	#6@12"	#6@12"	#6@12"	#4@12"	145	4000	B	ELEVATOR MACHINE ROOM
RS10	3	4	65	95	#5@12"	#4@12"	#5@12"	#4@12"	#4@12"	#4@12"	#4@12"	145	4000	B	COOLING TOWER LEVEL
RS11	3	3 1/4	100	55	#5@12"	#5@12"	#5@12"	#6@12"	#6@12"	#5@12"	#4@12"	115	4000	B	LEVEL 16 MEZZ. CECO ROOM

NOTES: 1. REINFORCED METAL DECK SLABS (RS) ARE TO BE SHORED AS REQUIRED TO MEET METAL DECK REQUIREMENTS. SLABS RS1, RS5, AND RS6 ARE ASSUMED TO BE SHORED FROM THE SLAB BELOW.

2. HANGING LOADS: TYPE A = 50/5
TYPE B = 100/5
TYPE C = 1000/25
WHERE THE HANGING LOAD IS GIVEN IN X/Y. "X" IS THE MAXIMUM PERMISSIBLE HANGING LOAD (IN POUNDS) APPLIED AS EITHER A SINGLE POINT LOAD OR AS MULTIPLE POINT LOADS WITHIN ANY "Y" SQUARE FEET OF SLAB AREA. POINT LOADS LESS THAN 50 POUNDS MAY BE HUNG FROM THE DECK IN ACCORDANCE WITH THE MANUFACTURER'S RECOMMENDATIONS. POINT LOADS GREATER THAN 250 POUNDS ARE TO BE HUNG FROM CAST-IN-PLACE ANCHORS.

3. 'MS' SLAB – NON ELECTRIFIED COMPOSITE METAL DECK SLAB.
'BS' SLAB – 40% ELECTRIFIED CELLULAR COMPOSITE METAL DECK SLAB (5'-0" X 5'-0" GRID OF PRESET INSERTS).

Figure 73. Reinforced metal deck slab schedule

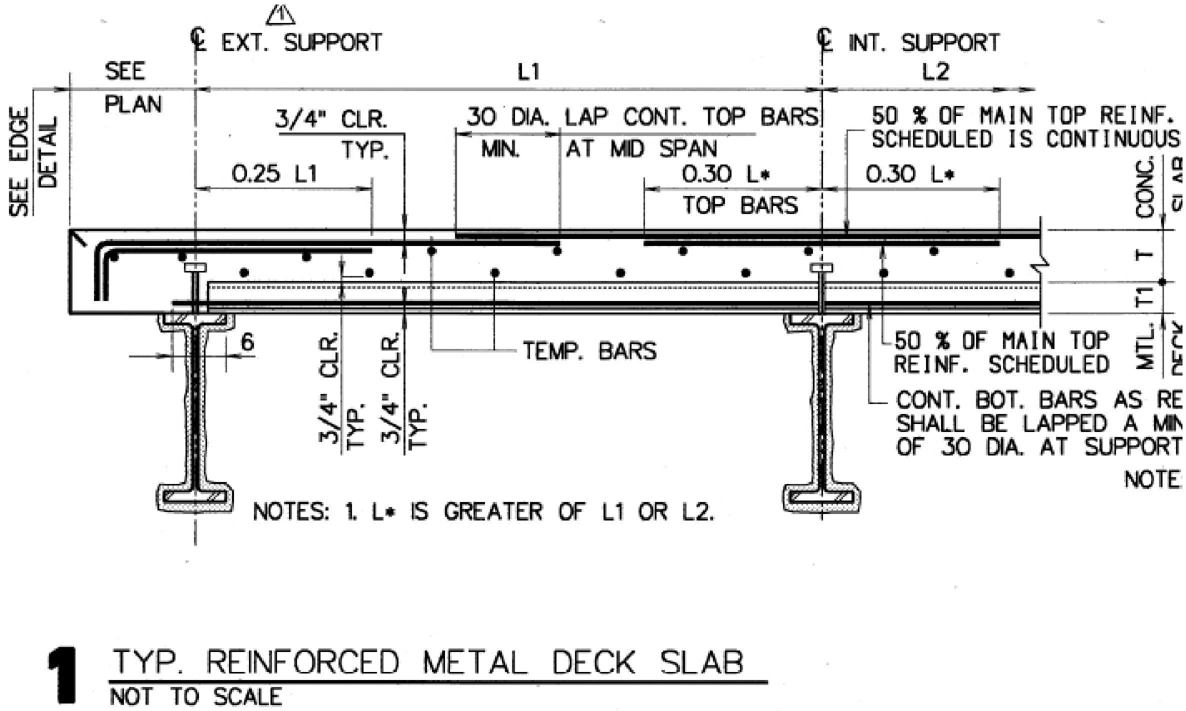


Figure 74. Detail of reinforced metal deck slab

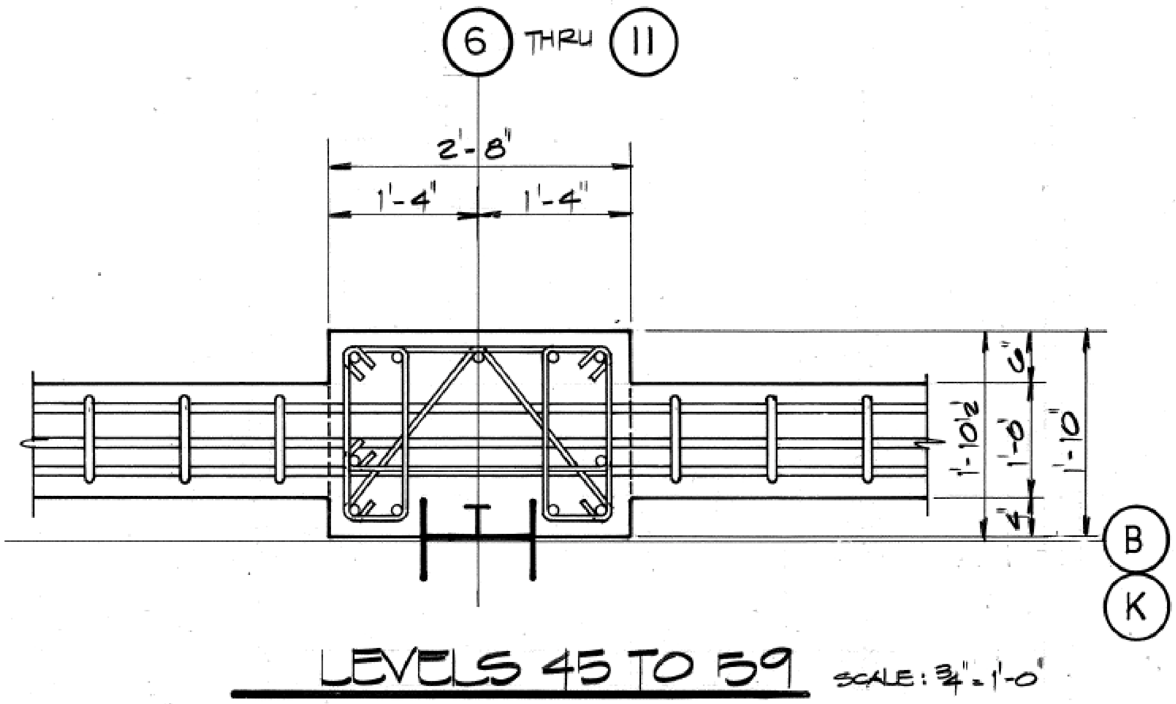


Figure 75. Detail of column

3.5.5.6. Specialized Reports for Construction Safety

Providing safety information for vulnerable locations can help reduce accidents in the field. The specific case of workers falling between rebars occurs when the top mesh spacing is more than a certain width. Openings in slabs, or in top meshes, are also a hazard. Vertical bars should be capped. Providing information on the locations of such dangers to safety managers can be valuable information, so that the manager can take precautions at these areas.

3.6. Concrete Placement

3.6.1. Pour breaks and stops

General	-
Conceptual Design	-
Engineering	• Model pour breaks and pour stops
Contracting	-
Pre Const./Detailing	• Model pour breaks and pour stops • Measure concrete volumes for pours • Schedule pours
Installation	• Report embeds and rebar for each pour according to pour breaks • In-field editing of as-poured breaks

Pour breaks and stops are logical partitions of the CIP structure to represent the pour work package or the actual pour. Pour stops and breaks are often specified through a project's Technical Specification. Here is an example from a typical specification for construction joints:

Construction Joints: Provide where detailed or to limit the size of pour units as follows:

- Slabs-On-Ground: Maximum 60' [18 m] in any direction.
- Framed Slabs, Beams, Etc.: Maximum 90' [24 m] located in middle third of span.
- Walls: Maximum 90' [24 m] in horizontal direction, maximum 20' [6 m] in vertical direction.

Pour breaks are the boundaries of a pour as planned by the construction managers. They are always made in locations that satisfy structural engineering stability constraints. They are not necessarily located at the boundaries of modeled logical objects. Pour breaks are mostly planar, but may be of two distinct types:

1) Breaks defined by a physical shutter, where the slope of the break is larger than the natural concrete slump; in this case there will need to be both attributes and placed components. For slab and wall breaks, the physical stop component could act as both a splicing tool, and is either a purchased item or a general stores consumable.

2) Breaks that are horizontal or near horizontal, where formwork is not needed.

In both cases a) and b) the component could also carry a surface object which can be instructive to surface treatments or preparations.

Pour stops are the actual locations at which concrete pouring was stopped during construction. They may be the same as the planned pour breaks, or they may be in different locations as a result of resource or weather constraints. Except for unusual circumstances, pour stops are almost always horizontal or near horizontal, without forms. As a result, they can follow paths that are not simply planar. But sometimes, curved surfaces are needed for representing complex geometric shape of slab member, as shown in Figure 76,

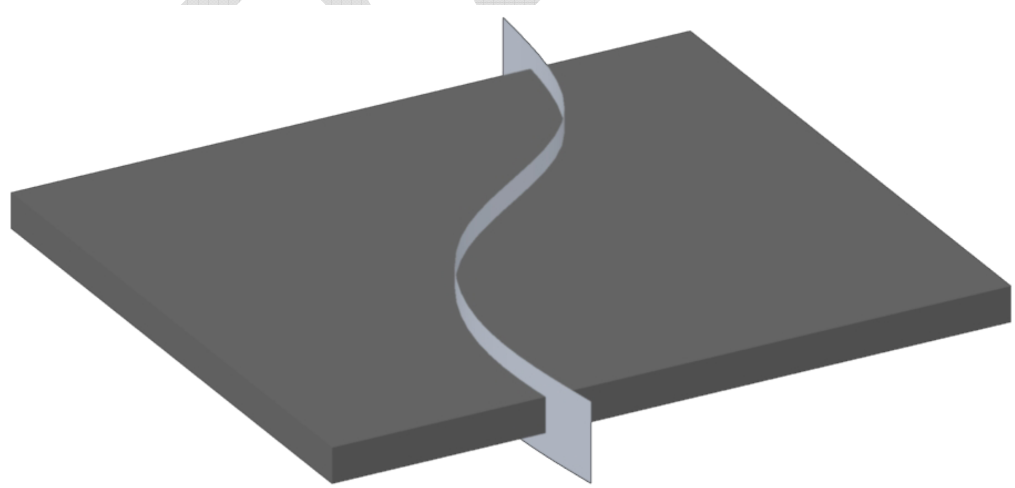
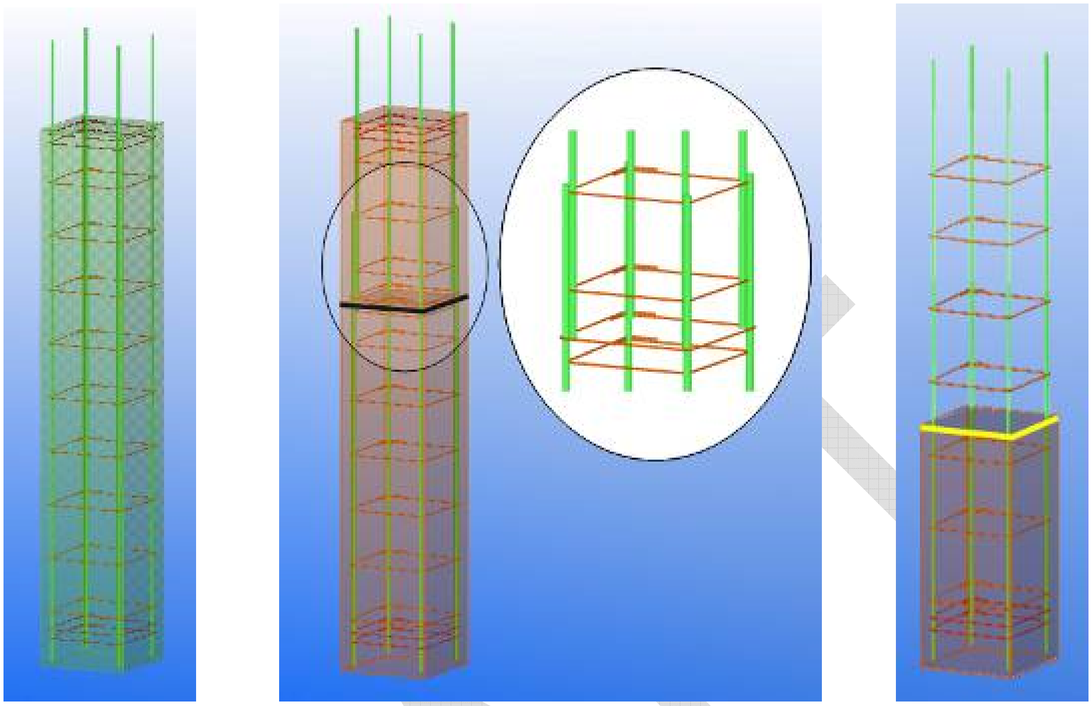


Figure 76. Pour break using curved surface

Pour breaks and pour stops are distinguished by their purpose. Pour breaks serve to delineate the extents of a concrete pour for allocation to a work package. An automated detailed quantity take-off of all of the rebar, embeds and concrete materials should be possible for each work package, which implies that a pour work package should have a volume (a 'planned pour') whose boundaries are defined by the existing concrete as-built to date on the one hand, and the planned pour breaks on the other. In this way a construction manager could extract the list and order the precise set of materials needed by simply defining the pour breaks. Pour stops serve to mark where concrete pouring was actually stopped. They serve as the markers of the 'as-built' condition.

Pour breaks and pour stops can also be distinguished by the way they manipulate the components, like rebars and embeds, within the elements. Pour stops are simpler and influence the concrete volume only (Figure 77c). They have no effect on the internal components. Pour breaks, on the other hand, are more complex; they do require change to the internal components that intersect with the pour break plane. The post-tension and steel reinforcement layouts are affected by pour breaks. The positioning of a pour break should have a cascading effect on rebar detailing (if applicable) and other aspects of detailing. For example, a pour break placed in a long column is often accompanied with a splice in the vertical rebars as shown in Figure 77b. The software should enable the user to opt for automatic insertion of the rebar cage splice arrangement when a pour break is defined. When a planned pour break plane intersects an embed in a model, the plane it must be displaced to either side of the embed in order to ensure that the object can be fully embedded in the concrete.

Depending on the maturity of the system and the interoperability with post-tension structural analysis software, the user could theoretically change the pours of post-tension elements. Users must have the ability to "lock" certain pours or the system must lock certain pours where the work process dictates.



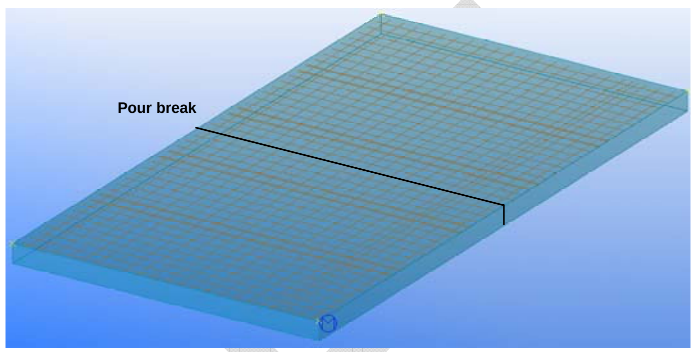
(a)	(b)	(c)
Original column	Column after placing pour break. The original column is now represented by 2 "sub" columns and the rebars are spliced	Column after placing pour stop. The original rebar cage was not changed

Figure 77. Pour break and pour stop principles.

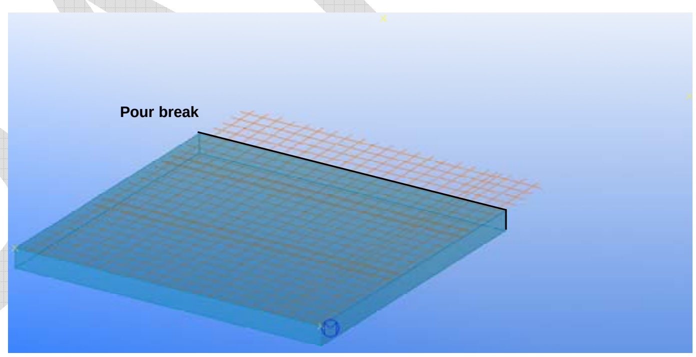
As mentioned, the pour breaks and stops are logical representations, which mean that the original logical model objects are not changed. They still exist in the background. The new elements are "shadow" elements, with all of their properties inherited from the original objects, and their properties cannot be edited directly.

When placing a pour break, the internal components that intersect with it can be associated with the "shadow" elements on both sides or they can be collected 'on the fly' when a report or drawing is generated. The actual association will be made once a work package is defined (see section 4.4.6) and the sequence of work packages is determined. For horizontally planar pour breaks (such as in columns or walls) any rebar splice will invariably be placed above the break and can be done before work sequence is determined. However, for vertical (or near vertical) pour breaks, slice locations can only be determined once the sequence is set. In Figure 78, for example, the mesh intersecting the pour break was attached to the left part of the slab (Figure 78b) because it was sequenced as the first pour. The right hand side was poured subsequently (Figure 78c). If the pour sequence was changed to pour the right part first, the meshes themselves would have been defined differently.

(a)
Basic slab with a vertical pour break



(b)
The logical 'concrete pour' that is planned to be cast first contains existing rebars, with sufficient extension beyond the pour break to ensure structural continuity with the subsequent pour.



(c)
The second logical 'concrete pour' slab without to be cast after the first slab

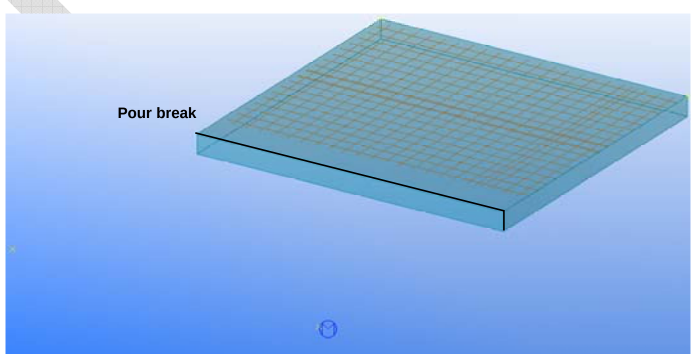


Figure 78. Pour break in a slab

3.6.2. Embeds

General	• Sleeves • Conduits • Anchor plates • Grounding • Water stops • Blockouts • Field Inspection • Documentation
Conceptual Design	-
Engineering	-
Contracting	-
Pre Const./Detailing	-
Installation	• Embedded Parts Installation • Installation Tolerance

The system should offer a wide variety of embeds to choose from or the ability to add these items. Five examples of embedded sleeves and plates are shown in Appendix A.

- Steel plates
- Studded, galvanized steel plates
- Pipe sleeves
- Pipe hangers
- Anchor bolts
- Strap anchors
- Water-stops
- Keyways
- Wall ties
- Block-outs

Five examples of embedded sleeves and plates are shown in Appendix B.

Each of these objects should behave parametrically. Items like the standard reinforcing detail around pipe sleeves over 4" should be automatically applied. The system should allow embeds to be selected from a pick list by manufacturer for easy placement. For some items, the user could input spacing and a selected span or object for the anchors to be placed in and the system should place all of the anchors. Clashes like anchors in doorways should also be detected.

3.6.3. Finishes

General	• Surfaces • Processed finishes • Waterproof
Conceptual Design	-
Engineering	-
Contracting	-
Pre Const./Detailing	-
Installation	• Mark up of patches (segregation)

The system should allow the user, at job setup, to define global finish criteria such as all walls have a 'Class B' finish or floors should have an Ff (floor flatness) of 'x' and an Fl (floor levelness) of 'x'. The system should also allow the user to change this on the fly. Another valuable feature might be to automatically assign finish criteria based on the building user's finished product perspective. For example, all cast-in-place walls in courtyard areas are to be Class 'A' but all cast-in-place walls interior to the parking structure should be Class 'B'.

The system should allow the user to identify areas of cracks, poor finish (where this is of consequence), where honeycomb occurs, where patching is needed, etc. The system should also allow the user to identify when and how these issues were remedied. Many of these issues, such as honeycomb and cracks, have structural significance.

Waterproofing is another finish where the system could play a role. The use of liquid applied waterproofing membranes is more straightforward from a systems standpoint than a fabric-like waterproofing. For liquid applied waterproofing membranes, the ability to apply a waterproofing surface for visualizing the location of waterproofing and deriving the amount of waterproofing is fairly simple. However, fabric-like waterproofing is sold in rolls and requires roll size, lapping rules, etc. to be taken into account. The system should be able to lay out both types of waterproofing automatically and their associative accessories like termination bars, nails, paragranular, etc. These items need not be modeled, but their representation should be in such a way that aids in calculation, and should include attributes such as:

- Cross reference to physical sample ID
- Required surface properties for application

3.7. Formwork and shoring

3.7.1. Detailing

General	• Connection to commercial solutions • Steel Decking • Wall ties • Pans, waffle slab systems • Commercial panel systems • Column systems • Treatment for surface using special type of formwork can also be considered
Conceptual Design	-
Engineering	• Stability analyses
Contracting	-
Pre Const./Detailing	• Permanent formwork • Temporary formwork • Stability analysis for incomplete structures (stability through incremental pours/stages) • Stripping clearance check for formwork (soft clash check)
Installation	• Temporary formwork • Safety/catwalks

Note on terminology: many people use the terms formwork and shoring interchangeably. Three terms are used:

Formwork - is the material or parts of the formwork system that are in direct contact with the concrete poured. This includes primary and secondary support beams.

Shoring - is the temporary structure that supports the formwork whether as vertical posts or towers for a slab, or bracing for a wall.

Re-shoring - is the temporary system of posts or towers or braces that support concrete elements after the formwork is stripped but before they have reached their final strength. Re-shoring is usually installed after shoring/formwork is removed to give additional support for loading above; this is temporary and not permanent.

3.7.1.1. General

Formwork is a fundamental part of the CIP process. For most structures, it accounts for more than half of the cost of the structure because it is labor intensive. There are many types of formwork used to install CIP concrete; from pre-manufactured systems to site fabricated formwork. As a result, the design, manufacture and deployment of prefabricated formwork has its own small industry. Some projects, customers or locales require that formwork drawings be stamped and calculations provided. Other situations only call for detailing where the detailer is using a predefined set of values for the amount, size and type of formwork to be placed.

It would be very desirable for CIP contractors to have automated routines for laying out the formwork configurations provided by them. However, it is not feasible for a BIM provider to provide model objects for detailing all of the available formwork systems on the market. Therefore, the modeling system should provide the basic objects and relationships that enable the major regional prefabricated formwork companies to model their products in the BIM system.

The starting point should be formwork quantity survey capabilities, which means the ability to assign formed surface area by element and/or pour.

3.7.1.2. Formwork Assignment

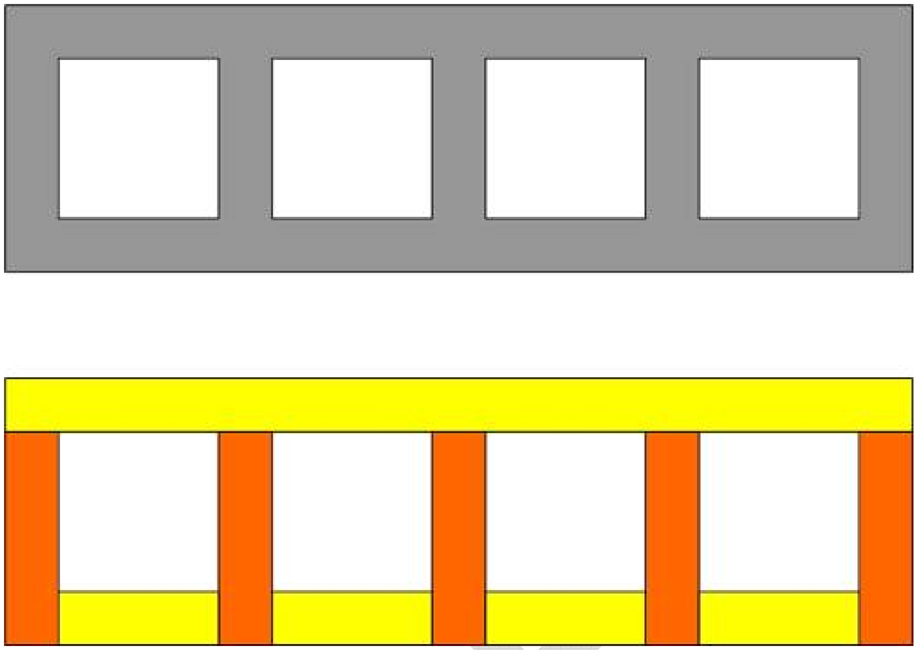
A minimal and effective way to manage formwork is to assign all faces of the structure that are to be cast against formwork with a secondary surface that can be partitioned into formwork elements. These surface elements, called ‘formwork surfaces’, would correspond, one-for-one with a formwork piece assignment. The formwork surfaces would have placeholders for: commercial product name, or “custom”, and a formwork mark number. This is useful for calculating quantities of formwork required during master production planning.

However, in practice, actual formwork is not assigned to RC members directly, but rather to predefined ‘concrete pour sections’. This is apparent in Figure 81. The pour breaks (defined in section 3.6.1) define the extents of the pour and also determine how the formwork will be configured. One of the reasons for this is that the pour sections have the pour break surfaces, while the RC members do not.

It must be possible to partition the surfaces of all RC members or of pour sections into formwork surfaces. A checking program that identifies completeness of coverage for surfaces of members would be useful. Once formwork surfaces are assigned, it should be possible to collect them into work packages, as defined in section 3.8.4. Schedules for formwork preparation and placement could also be automatically generated, and by virtue of their association to work packages, the formwork surface elements would be scheduled similar to machinery or work crews, as temporary assignments, within scheduling applications.

When designing the detailed formwork layouts for a structure, the topology of a structure plays an important role in determining the formwork system that will be used. For example, the panel shown in Figure 79, can be considered in two ways:

- As a panel with window block outs, in which case first one side of the whole panel would be formed, then the blockouts for the windows would be attached to the first side of the form, and then the form would be closed on its second side, again covering the whole gross area of the panel.
- As a set of columns and beams, in which case first the columns would be formed and poured, then the beams in a second pour (it would also be possible to pour the two sets of beams – upper and lower – separately.



•

Figure 79. Defining desired topology and construction method

A similar important aspect of formwork layout that influences how the software should function is illustrated by the decision of where to place the pour breaks at the tops of the columns in any case where beams frame into the tops of columns. Should the columns be poured to the base of the beam and then the beam formed and poured as one long element (as shown in the left side of Figure 80), or should the columns be poured all the way to the top of the beams and then the beams formed and poured between each individual column (as shown to the right of Figure 80)? This will mostly depend on whether the beams or thinner or thicker than the columns and whether the structural engineer requires allows pour breaks in the column or in the beam. In any event, the software 'formwork surfaces' must be able to cover areas of a structure that are not restricted to a single logical member; the relationship between members and formwork surfaces is m:n and the only geometric restriction is that the formwork surface must be co-planar with a member surface.

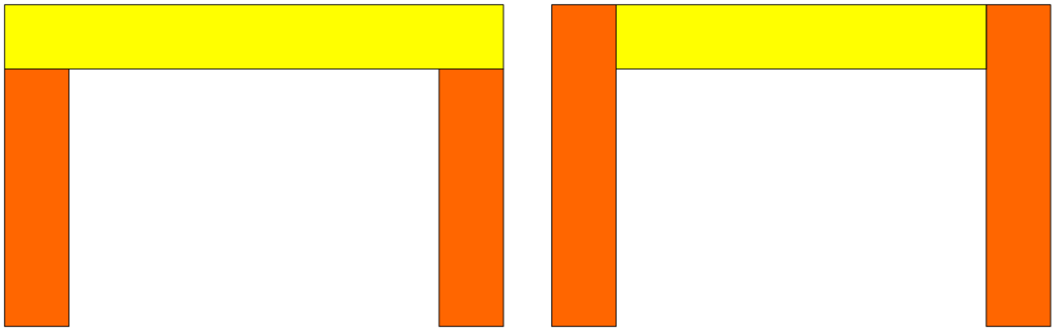


Figure 80. Alternative approaches to pouring columns and beams

Also, the software should allow association between formwork surface son opposite sides of a wall or column, because in practice these are always tied to one another to resist the hydrostatic pressure of the concrete during pouring. Ideally, the software would offer the user the convenience of automatically generating a companion formwork surface on the opposite side when the first side is generated.

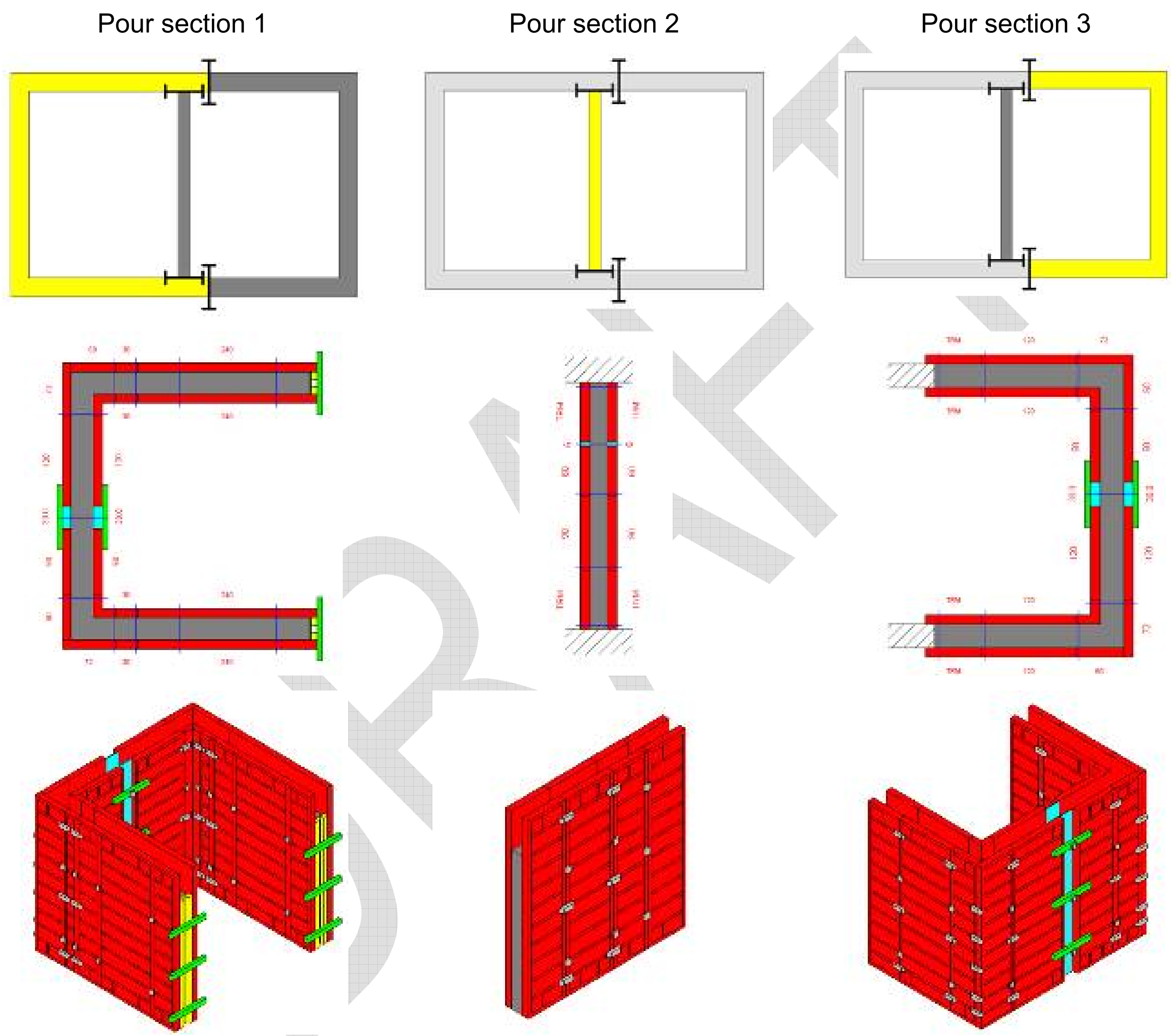


Figure 81. Assignment of formwork to concrete pour sections (courtesy of Peri)

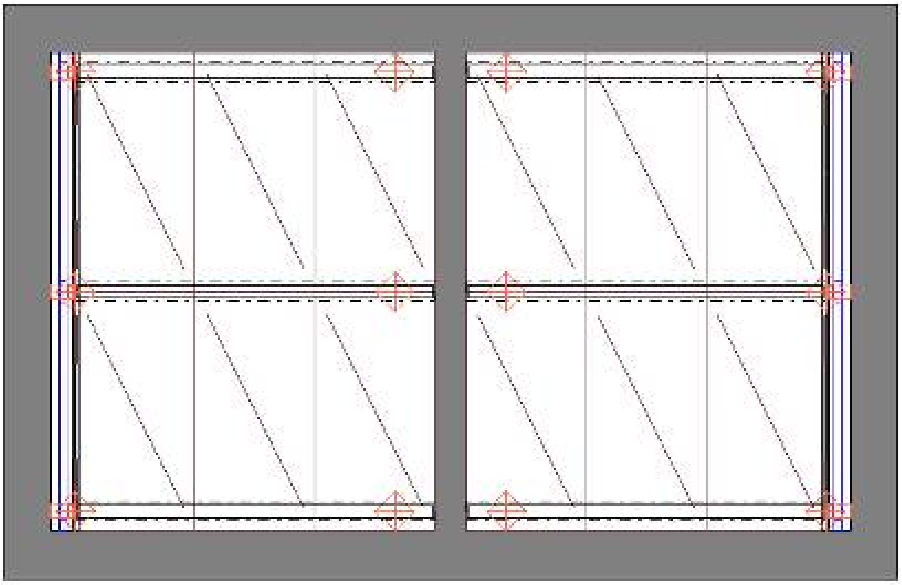


Figure 82. Slab formwork layout

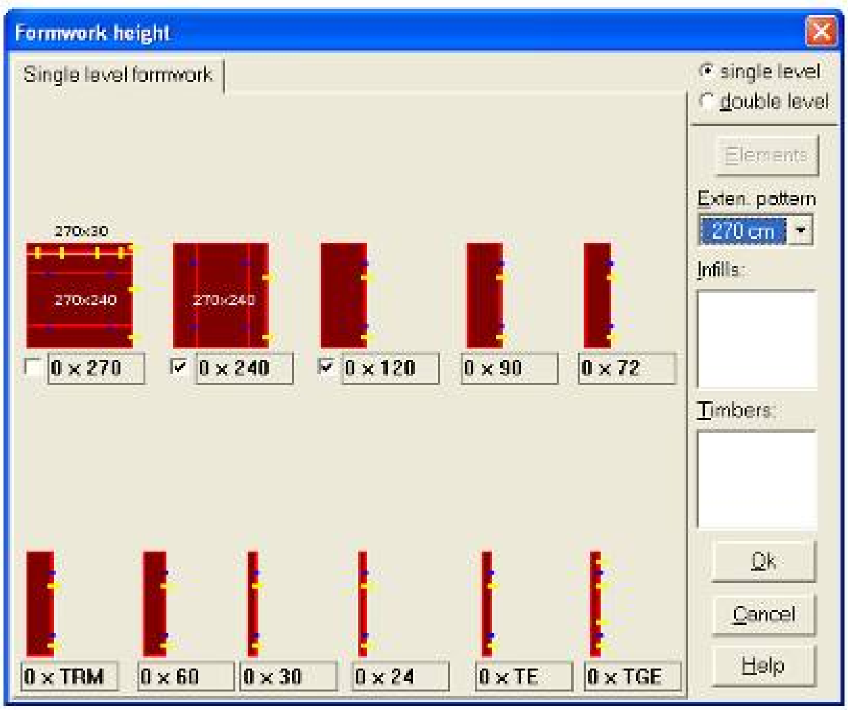


Figure 83. Formwork panels height setting (screenshot of Peri Elpos program)

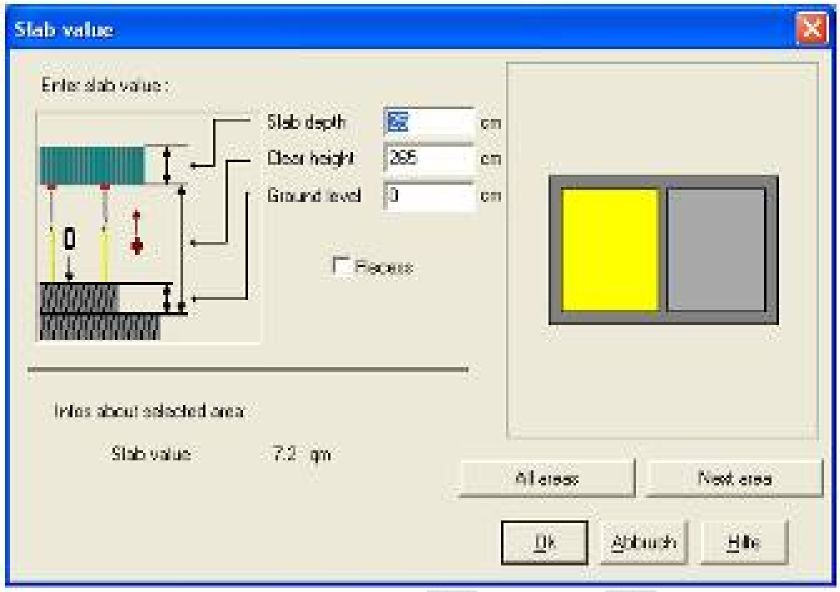


Figure 84. Slab formwork attributes (Screenshot of Peri Elpos program)

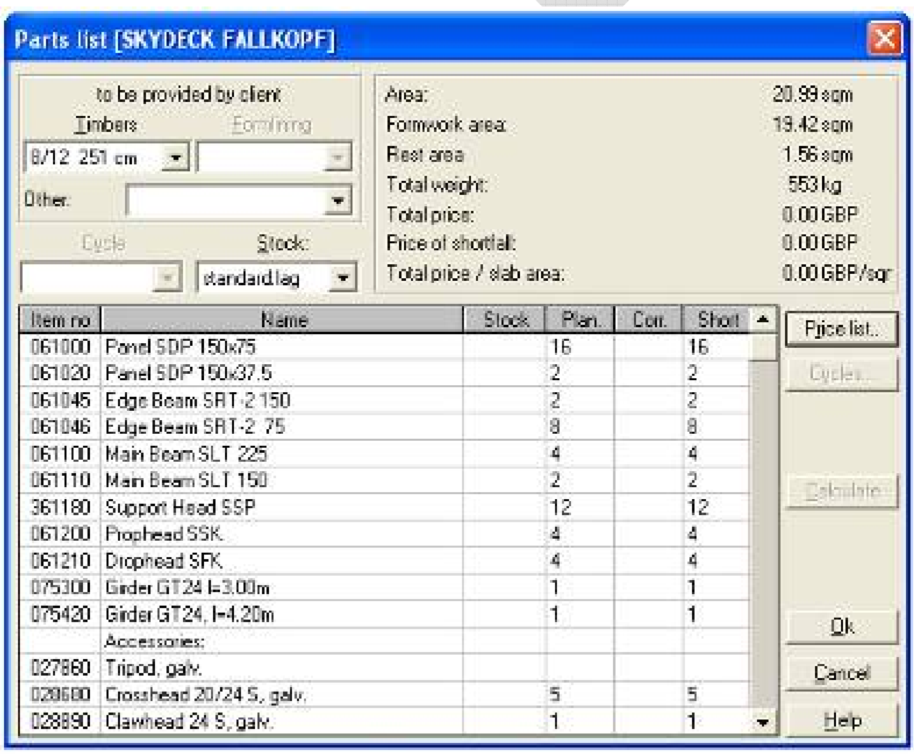


Figure 85. Formwork parts list (Screenshot of Peri Elpos program)

3.8. Project Management

3.8.1. Reporting

General	-
Conceptual Design	• Quantity take-off of gross structural area or volume for high-level estimating
Engineering	• Quantity take-off of gross concrete volumes/element measures/counts for high-level estimating
Contracting	• Quantity take off for detailed line-item bill of quantities
Pre Const./Detailing	• Measure rebar weights and concrete volumes for each pour • Lift drawings
Installation	• Detailed rebar installation lists and concrete details for each pour • Data management to track mix and slump tests

3.8.2. Configuration management

General	• Change tracking • Approvals • Archiving • Assign management attributes to reference files/objects that are imported and displayed • IFC conformance • Status tracking by object grouping to designate status during all design construction stages.
Conceptual Design	-
Engineering	-
Contracting	-
Pre Const./Detailing	-
Installation	-

Configuration management deals with the "who", "what", "when" aspects of project tracking. It involves carrying multiple types of metadata for addressing these issues and for managing the consistency, versioning, and access of the data. The intended goal is to allow a Tekla RC user to manage the evolution of project data, for the system, to be able to tell when derived data is no longer current, and to track as necessary who created, modified or deleted data over the project lifetime.

1) Each object in the model, including all geometry, all pour objects, reinforcing, formwork objects, embeds, etc shall carry a flag telling when it was last acted on, for example by using a timestamp.

2) Most objects are aggregated, such as a pour shape with embedded reinforcing or embeds. Aggregation objects take the timestamp of the most recent update of any of its components.

3) An action invoking a timestamp is Creation, Modification or Deletion. types of actions taken on an object should be reportable.

4) In addition to time and action, it is necessary to track who made the actions taken. This may be accomplished in multiple ways, but one is to carry a reference from all action taken to a transaction entry, designating who made the actions between 2 timestamps. The transaction entries are part of a transaction log, capturing all actions taken within the project.

5) It should be possible for the history of transactions involving any object to be searched .and reported

6) Deleted objects of course no longer exist. However, for interoperability, their GUID needs to be carried for the life of the project.

7) All drawings and reports that are created also have a timestamp. A drawing or report is considered an aggregation object, so if any of a drawing's component objects change, it is flagged as "obsolete". There should be mechanisms that make the user aware of all obsolete reports/drawings, and they should be re-generated.

8) An option often desired by a user is to save a project copy, then undertake one set of changes on that file, and another set of changes to another file. File saved for this purpose should be specially marked. Multiple alternative files should be possible. Any reports from such an "alternative project" should be distinguished in any reports or drawings.

9) Data from one alternative should be able to be imported into another alternative, so that the best of a set of alternatives can be generated. Inconsistencies arising from such merging should be flagged and reported. After such imports are made, the exporting alternative should be deleted.

10) In the check of a project, it should be possible to check whether a project has any alternatives, requesting that the user resolve them.

Objects imported from other systems for use in coordination are called reference objects. Reference objects exported to other systems are described below. Imported objects should carry the same information, if available.

- All reference objects need to carry the following information:
  Building system to which it belongs
  Company or contact person associated with the object
  A GUID assigned by its authoring system
  A timestamp that can be used to manage its currency, in relation to its source system.

- The Tekla system should be able to track all exported files, possibly as part of their transaction manager. This will allow TS to know when an exported reference file is no longer valid.

3.8.3. 4D

General	• Sequences and simulation
Conceptual Design	-
Engineering	-
Contracting	-
Pre Const./Detailing	-
Installation	• Schedule (Concrete, formwork) • Time stamping • Strength gain

4D animation of a phased construction plan is important for devising, analyzing and explaining construction sequences to workers. However, quite unlike steel or precast, simple phasing of parts in the model is not a sufficient approach, because reinforced concrete CIP buildings are not built as distinct complete parts in the way that they are designed. To achieve useful 4D analyses, phase information must be applied through the medium of ‘work package’ scheduling, and the animations must show formwork and pour sections, not basic building design elements.

Aspects that can be examined using 4D include:

- Local animations of bar placement in areas of congestion;
- Phasing of shoring, formwork, pouring concrete, stripping formwork and re-shoring.

3.8.4. Define work/pour packages

General	• Defining pour packages • Defining formwork regions • Defining rebar sets • Methods • Resource planning • Rates
Conceptual Design	-
Engineering	-
Contracting	-
Pre Const./Detailing	• Create work packages in parallel with scheduling software, including sequencing. • Link work packages with components/pours.
Installation	• Work-ready checking • Pull materials for work packages including secondary elements for separate pours (and blockouts) • Lift drawings • Work package data should cover size of pour, delivery plan (crane, bucket, chute,...), pour patching, man power, heating • Monitoring of quality assurance characteristics (temperature, maturity/cylinders strengths of concrete) • Monitoring of management data (delivery notes, actual resource consumption)

In the execution of reinforced concrete works on site, the responsibilities of a construction manager begin with determining exactly which sections of the building are to be cast in any particular concrete pouring activity. The selection will depend on the stage of the work, availability of equipment and labor, the mix type of concrete used for the elements (aiming to collect elements with the same material into the same pours), and structural constraints. Once a set of structural elements to be poured has been determined, the manager must ensure that:

- all of the rebars (cages, meshes, and loose rebar) are available;
- all of the embeds are available;
- the formwork is built, prepared to provide the right surface finishes, and supported adequately to carry the load of the liquid concrete;
- crews and concrete pouring and finishing equipment is assigned;
- any other conditions, such as climate or others, are suitable.

Once the pour is complete, test results from the cylinders taken during the pour for testing, must be monitored to determine the right time for stripping the formwork and removing shoring.

What makes this management activity difficult is that in many cases the decision about which elements to include or exclude form any specific pour is taken fairly close to the time at which the pour is performed. This means that the manager must integrate a large amount of detailed information in a short time. Common practice is to ensure that most of the rebar and embeds, and formwork equipment, is available on site well before it will be needed, is the result of the inability to cope with all of the information that would be needed to pull the materials to the work location 'just-in-time'.

A good software tool that could enable all of the above planning and monitoring tasks to be done quickly and with accurate information could provide a great advantage. The building information model is the most appropriate platform for this functionality. The system would need new objects unavailable in existing BIM software. The first object required is a 'Work Package', which is the basic unit for scheduling construction activity. A work package would hold the activity timing information and would aggregate the different structural elements, or their parts, that are to be built together. Work packages would represent 'formwork', 'rebar fixing', 'concrete pouring', 'curing' and 'stripping' activities.

Another data object – a 'pour section' – is required to collect the structural elements that are to be built as a group, to define the precise geometry of the section of the structure that is to be built, and to collect the rebars, embeds and other parts and equipment that is needed.

Definition of pour sections should be dynamic, so that a manager can indicate directly on the graphic computer model where the extents of each pour are to be. Changes to a pour section should be made simply by moving the 'pour breaks' which represent the physical stoppers applied in the formwork to restrict the flow of concrete, i.e, the software should compile the contiguous sections of concrete that make up a pour section automatically. Multiple pour sections will need to be aggregated into a work package.

It should be possible to generate a complete BOM of all of the parts needed to perform a work package. This will provide managers with the information they need to pull the requisite materials to the right place at the right time without waste.

Detailed definitions of the 'work package' and 'pour section' aggregation objects are provided in the section titled 'Special Aggregations' in Chapter 4 of this report. Definitions of 'pour break' and 'pour stop' objects are provided in the section titled 'Pour Breaks and Pour Stops' in the same chapter.

3.8.5. Control

General	• Live links to database tables of external construction management software. User-friendly mechanism required.
Conceptual Design	-
Engineering	-
Contracting	-
Pre Const./Detailing	-
Installation	• Layout • Survey points • Schedule • Cost • Productivity analysis • Material tracking and handling • Check tolerance in comparison with As-Built model, survey points and inspection after shoring removal

A detailed building model of a CIP structure is an invaluable platform for production control, since it will contain detailed records of the work packages, products and resources. It is also an ideal vehicle for reporting derived management control data, in the form of color-coded or filtered visual reports.

While the majority of the applications are at the installation stage, if a record of actual performance data is compiled for projects, then that data can become a valuable asset for estimating and preconstruction planning of future projects.

The general requirement for effective implementation of any of these uses is that it must be possible to create live links to the databases of companies' external production control/management software.

3.8.6. As Built

General	• Track changes
Conceptual Design	-
Engineering	• Incomplete stage structural analysis
Contracting	-
Pre Const./Detailing	-
Installation	• Mark up actual pour stops • Incomplete stage structural analysis

If a building model is used for production control, which implies updating of production data as the structure is built, then little more is required to accumulate the data for the as-built model. For example, future forensic investigation of a building may require knowledge of the conditions and materials used for a particular pour; if the actual pour stop locations are recorded, and the pours are linked to the concrete mix delivery data, then all the information needed will be present in the model once the structure is completed.

One of the most critical tasks of engineers is to ensure that the work completed up to any point forms a stable structure. This may mean careful staging of the works or addition of temporary works to support the structure until the final supporting elements are completed. To do this, the following functionality is required:

- The ability to define load combinations and loads that reflect a structure during construction. These would include temporary loads imposed by shoring, formwork, fresh concrete and re-shoring. Ideally, a system would be able to generate a loading automatically from a set of simple generic rules. Note that in many cases construction loads should also include the static and dynamic effects of construction equipment that might be attached to a building, such as tower cranes, self-raising hydraulic concrete pumps or formwork systems, etc.
- The ability to deduce the state of the structure at any given planned date as a function of the construction schedule defined for the work packages. The

elements of such an analysis model would be based on the pour sections rather than on the final concrete elements.

- The ability to model temporary structural supports, in particular re-shoring.

If these three functions are provided, then standard or specialized external analysis tools could be applied to determine a) whether the structure is stable, and b) whether it can withstand the applied loads.

3.9. Value to Owner

The process model identified a number of uses that could be made of a CIP building model that would provide added-value to owners. Three specific areas were identified: life-cycle costing, sustainable construction and facilities management. All three are novel uses that will only become applicable once all of the preceding requirements for modeling CIP structures have been met and models become commonly available. None of the three were detailed by the consortium members; this can be attributed in large part to the lack of experience the members have in providing such model-based services.

General	-
Conceptual Design	• Life-cycle costing
Engineering	• LEED evaluation
Contracting	-
Pre Const./Detailing	-
Installation	• LEED evaluation • As-built model for facility management

4. Functional Specifications of New Capabilities

Chapter Three of this report provided general definitions of the user requirements needed to support parametric production modeling of cast-in-place reinforced concrete structures. Much of the identified capabilities can be provided within the modeling software by making additions to existing functions, or even, in some cases, simply by adding appropriate custom components. However, some of the capabilities required cannot be supported without fundamental consideration, and possible adaptation, of the way in which the software functions. This chapter outlines a set of the most important functional requirements that this specification has identified for expanded support for CIP RC engineering.

In some of the sections, EXPRESS-G schemas are provided to illustrate the object relationships described in the text. These are not meant as literal proposals for specific ways in which the software data structures should actually be implemented, but are simply a means to communicate the design intent in a definitive way.

4.1. Connecting Volumes

One of the most basic requirements raised in chapter 3 (discussed in section 3.2.2) was the ability to correctly model objects with intersecting volumes and to resolve those volumes in different ways according to different needs such as structural analysis, reinforcement detailing, material take-off and estimating, etc. Section 3.5.1 (reinforcement detailing) provide one example. Another issue, raised in section 3.5.3, regards the nature of connections between cast in place elements; it deals with the fact that occasionally connecting rebars belong both to the elements and the connection simultaneously. A further fundamental requirement dictates that a cast in place building model should support multiple representations of the same information according to different functional views such as analysis, rebar detailing, pre-construction planning, formwork layout, etc.

Below, we offer one approach that we believe will achieve what we understand is required. It is not meant to be a prescription, but rather to illustrate one means as an exemplar, that can be used as a reference for a Tekla implementation.

4.1.1. Glossary

- *Connected Members:* two or more reinforced concrete basic objects (columns, beams, slabs, etc.) that are to be connected. They will usually have overlapping volumes, but this is not a necessity.

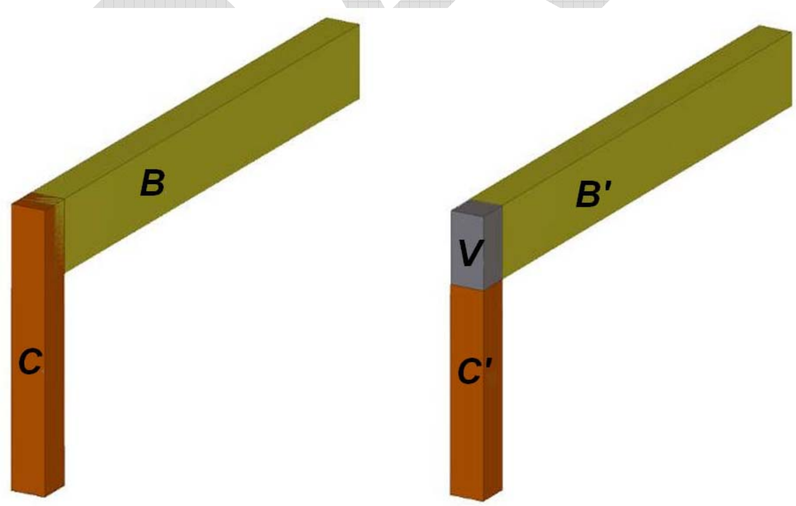
- *Connecting Volume*: an object with a geometrically defined volume of concrete that is created at the intersection of two or more connected members. Part or all of its volume may overlap with the original volumes of connected members.

- *Connecting Face:* an end face of a linear connected member, or an edge of a planar connected member, that falls wholly or partially within the volume of an associated secondary connected member. Connected faces define the vector (perpendicular to them) that defines the direction in which a face of a connecting volume may need to be extended.

4.1.2. Candidate Solution

The proposal generated here is from a structural perspective – that the RC detailed design is generated from the structural members outward to derive the finished forms. It can also be used, we believe, to deal with an 'outside-in' approach where the external target form already exists and structural members are placed inside the given form. Comments regarding this second approach are discussed at the end of the section.

The main objective behind this proposed solution is to transform the intersecting volumes between connected members into actual objects – called 'connecting volumes' – that can be associated in different ways with any of the connected members according to predetermined rule sets for different information processing purposes. A typical situation is shown in Figure 86(a), where a beam (with shape *B*) whose location is determined from a grid intersection intersects with the top of a column (with shape *C*) whose top position is defined by the same grid point. The resulting intersecting connecting volume is shown in gray in Figure 86(b), labeled *V*; the new column and beam shapes are labeled *C′* and *B′* respectively.



(a) (b)

Figure 86. Creation of a connecting volume between a column and a beam.

When the connecting volume object is generated, the connecting members are fitted to the new object. This eliminates the overlapping concrete and allows full flexibility in rebuilding the elements for different purposes. For example, for laying out reinforcing for the beam, the relevant beam shape is the volume $B' \cup V$, and for reinforcing layout of the column, $C' \cup V$. The way the volumes are reconstructed for each function is determined by a set of rules for each kind of connection. The rules should be user-configurable, because they will depend in some circumstances on construction methods and other accepted practices. An example of this is the case of quantity take-offs for a bill of quantities. If the contractor plans to cast the column up to the base of the beam, and then cast the connecting volume together with the beam, then the quantity take-off for the column is simply C' and that for the beam is $B' \cup V$

By the same token, the user should be able to toggle between different 'views' of the model, in much the same way that one can toggle between wireframe and rendered views. A 'reinforcing layout' view would show all the concrete volume as merged without division; a quantity take off view would show the beam and the connecting volume as a single unit. Each view would conform to the general set of rules defined by the user for each purpose and for each set of connected member types. The nature of many of the possible relationships between different connection members is explored below. Note that although the volume and surface area of each of the connected members is altered when the connecting volume is generated, the axis and the definition points need to remain as they were originally defined by the user.

4.1.3. Generating Connecting Volumes

In general, the connecting volume boundaries should cover the *extended* overlapping volume and not the actual overlapping volume. To create a minimal connecting volume object, the system should be able to locate its boundaries automatically. 'Connecting faces' of each of the connected members should be extended to the outer parallel boundary face of the other connected member. This will result in a volume that encompasses the full connection space.

Table 4 is an example of such a case. In any such event, the user could be informed in case his/her actual intent was different. If the operation is done 'on-the-fly' according to user selections of members to be connected, then users will have the opportunity for visual inspection at the time the connecting volumes are generated. However, if they are done in a batch process, then flags could be raised wherever 'new concrete' results.

The general principle for generating a connecting volume between any two connecting members can be stated thus:

1) Identify the connecting members, say *A* and *B*.

2) Determine the physical intersection between the two members ($A \cap B$).

3) For each of *A* and *B* in turn, extend the 'connecting face' of the connecting volume that is perpendicular to the axis of the first connected member to align it with the face of the second connected member that is furthest from the first connected member.

4) Cut the connecting volume from each of the connected members *A* and *B* to generate A' and B' as follows: $A' = A - V$, $B' = B - V$

Table 5 shows a sample of the ways in which connecting faces can be identified for different connected member pairs. As can be seen, the end face of any linear connected member (column, beam) or the edge face of any planar connected member (slab, panel) is a 'connecting face' only if it falls wholly or partially within the volume of the other connected member.

Table 4. Examples of connecting volumes for different pairs of connected members

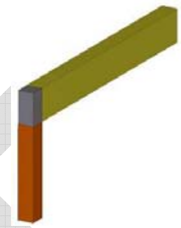
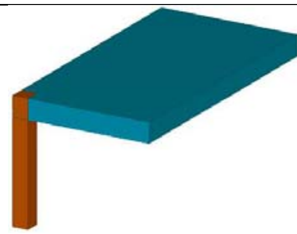
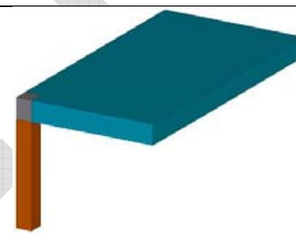
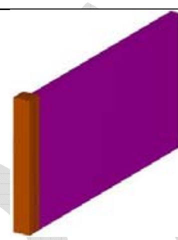
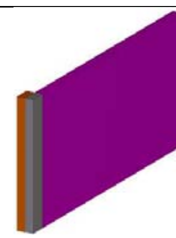
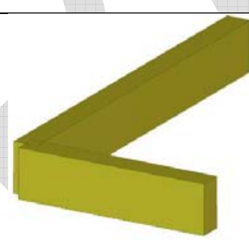
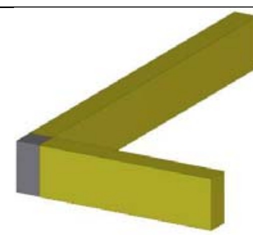
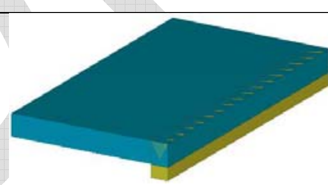
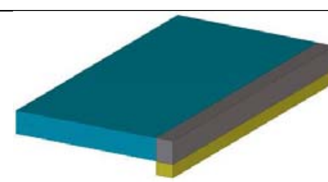
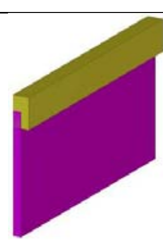
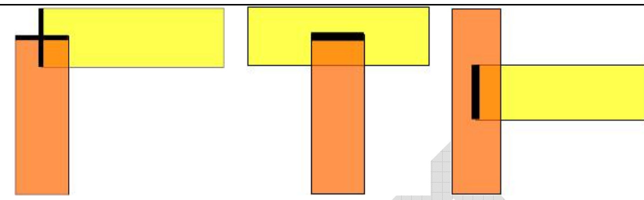
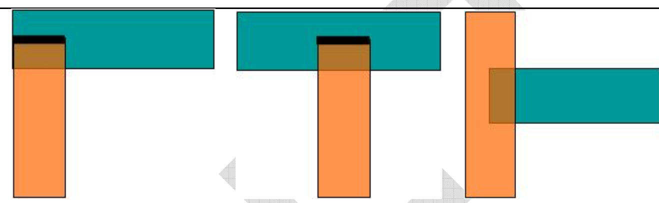
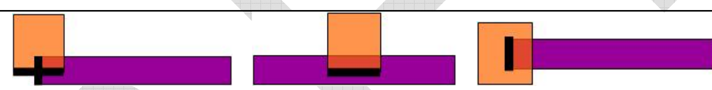
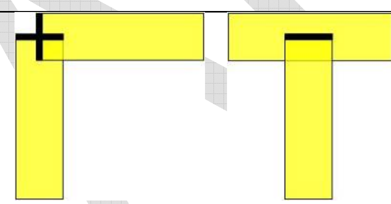
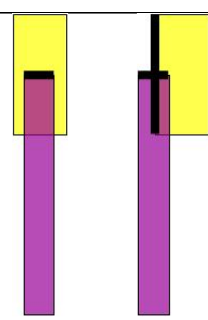
Connected members	Original connected members	Connecting volume and adjusted connected members
Column-Beam		
Column-Slab		
Column-Panel		
Beam-Beam Panel-Panel		
Beam-Slab		
Beam-Panel		

Table 5. Examples of the 'connecting faces' for different combinations of connected members (connecting faces are shown as bold lines).

Connected Members	Connecting Faces
Column-Beam	 Side View
Column-Slab	 Side view
Column-Panel	 Top View
Beam-Beam Panel-Panel	 Top View
Column-Column Slab-Slab	No connecting Faces. Connecting volume should not be created. Intersecting volume should be reported as a clash.
Beam-Slab Slab-Panel	 Side View
Beam-Panel	 End View

In cases where more than two connected members intersect, the system should create a connecting volume object for every intersecting pair of members. After the individual connecting volumes are created, the system should identify any overlaps between the resultant connecting volumes. If overlap is identified between any pair of connecting volumes, then the same logic used to resolve overlap between connected members should be used to divide the overlapping connecting volumes such that the overlap is removed. This may result in three volumes to replace the original two, or, if any one volume is entirely contained within the other, then only two distinct volumes will result. This may be repeated recursively until all overlaps are resolved. Figure 87 illustrates this process with a simple example of a three way connection between a column, a beam and a slab. In the step from Figure 87, three operations between the pairs of connected members generate three connecting volumes V_{SC}, V_{BC} and V_{SB}. In the second pass, from Figure 87, two of the connecting volumes are changed, producing V'_{BC} and V'_{SB} ; V_{SC} remains unchanged.

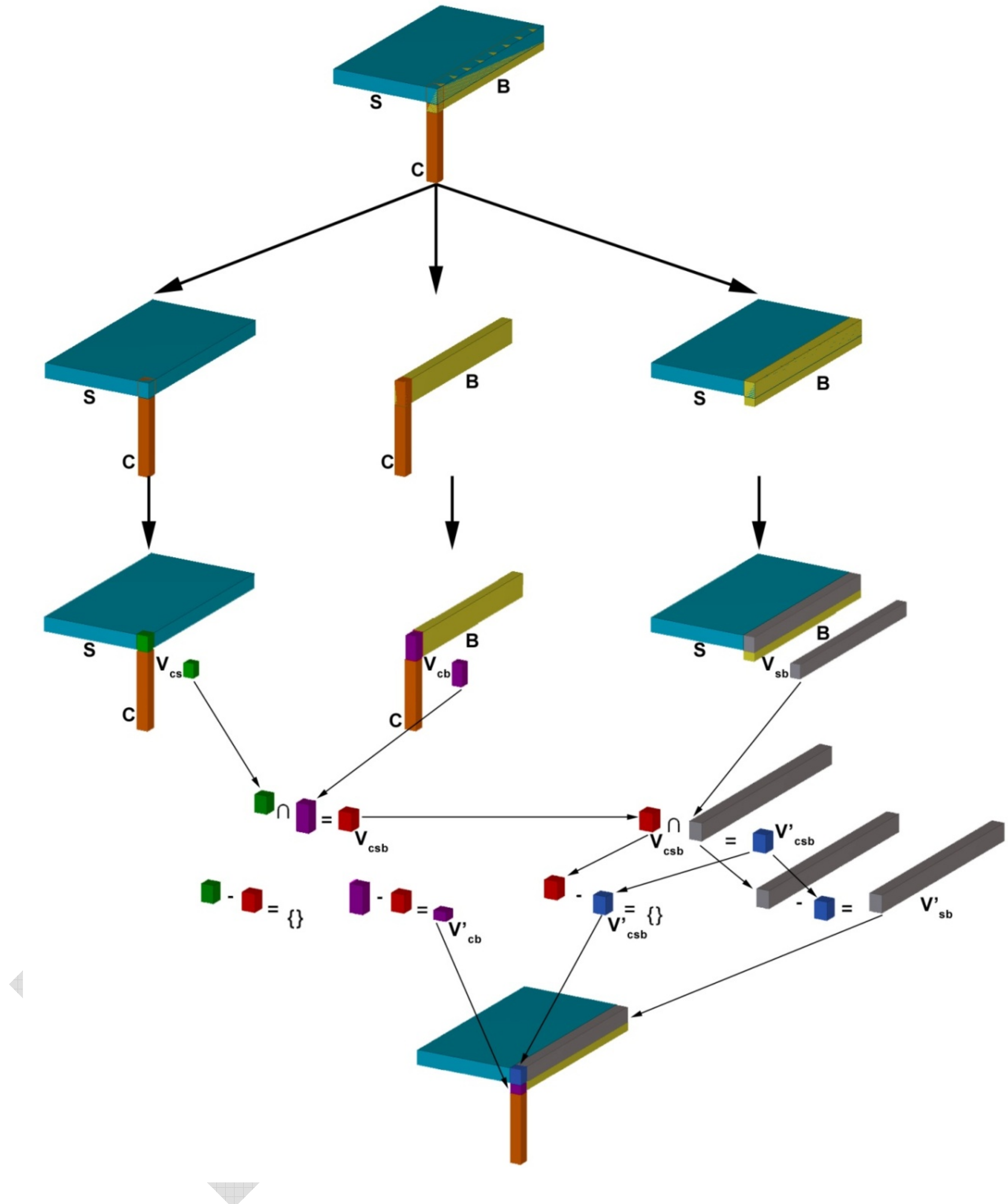


Figure 87. Connecting volumes for three elements

The original connected member objects and the connecting volume objects resulting from this example, and the relationships between them, are shown in Figure 88.

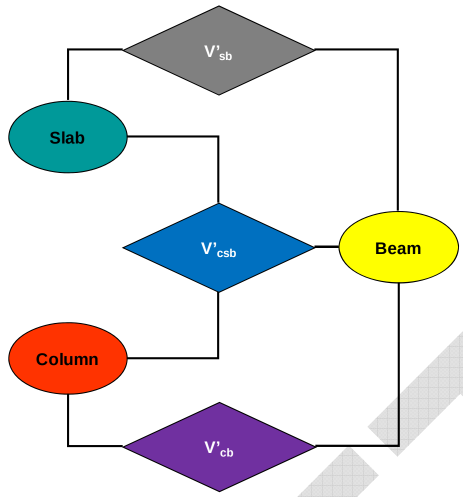


Figure 88. Object connection diagram

The algorithm for generating connecting volumes can now be generalized to treat multiple connected members:

1) **Treat the members**: Identify a set of connected members (*A, B, C…*).

2) Identify the first two members (the sequence can be arbitrary), say *A* and *B*.

3) Determine the physical intersection between the two members and generate a connecting volume V_i, such that $V_i = A \cap B$.

4) For each of A and B in turn, extend the 'connecting face' of the connected member that is perpendicular to the axis of the first connected member, to align it with the face of the second connected member that is furthest from the first connected member, so defining a new location for the appropriate outer face of the connecting volume V_i (only the connecting volume is actually changed in this operation).

5) Cut the connecting volume from each of the connected members *A* and *B* to generate *A′* and *B′* as follows: $A' = A - V_i$, $B' = B - V_i$ (the definition points and axes of *A* and *B* are not changed).

6) Repeat steps 2 to 5 for each and every other combination of connected member pairs (*A* and *C*, *C* and *B*, etc.), generating connecting volumes V_{i+1}, V_{i+2}, ….

7) **Treat the resulting connecting volumes**: Start with the set created in step 6.

8) Identify the first two connecting volumes in the set, say V_1 and V_2.

9) If they have an overlapping volume, continue – if not, cycle to the next pair.

10) Determine the physical intersection between the two members and generate a connecting volume V'_i, such that $V'_i = V_1 \cap V_2$.

11) Cut the new connecting volume V'_i from each of V_1 and V_2 in turn, to generate V'_1 and V'_2.

12) Add the new connecting volumes V'_i, V'_1 and V'_2 to the set (note that one of V'_1 and V'_2 may be null, in which case they are discarded).

13) Remove V_1 and V_2 from the set.

14) Return to step 8 and repeat until no volume in the set overlaps with any other volume.

4.1.4. Relationships between connecting volumes and creating members

When a concrete feature or external form has already been defined by the architect or other designer, the procedures outline above identify <u>minimal</u> connecting volumes. These should be entirely sub-joint (inside) of the externally generated form. Adjustments to the external form should be made when they are not completely contained.

As mentioned in section 3.2.2, there are different functional needs from the model, each of which requires that the volumes be resolved in different ways, according to different sets of rules. In this proposed solution, the set of rules should define the ways in which the connecting volume objects are related to the connected members.

All connecting volume objects have a 1:*n* relationship (n≥1) with a set of connected members. Each member in a building model may have 0 to *m* connecting volumes associated with it. When a connected member is changed, all its dependent connecting volumes should be updated accordingly. The representative shape of any connected member that is shown in any model view is always its own basic cut volume unified with any associated connecting volumes, according to the rules defined for that view. By the same token, the non-geometric properties of a connecting volume can be inherited from any one of its associated connected members according to the same function-dependent rules.

For example, when applying a reinforcement macro to members, the user should decide whether rebar should be extended into the connecting volume or not. This can be done *ad hoc*, or, preferably, can be implemented by a default set of rules. Another example is a bill of quantities. In this case, the actual areas or volumes should be calculated according to the rules for that function, defining the members with which the connecting volumes should be unified. Table 6 provides a matrix of different possible rule settings for each pair of connected member types.

The relationships between the connecting volume and the elements are also useful for different visualizations. Different sets of rules can be applied to specific views to create several alternate representations of the same model.

In practice, it may also be necessary to enable users to locally override the application of a rule for any particular connecting volume object and any of its connected members, allowing more precise control over a design.

Table 6. The range of possible rule selections for the connecting volumes between different pairs of connected members.

	Column	Beam	Slab	Panel
Column	• Not allowed (report clash)	• Union with beam • Union with column • Conditional on relationship between column width and beam width; If <condition> then union with beam, else union with column • Union with both	• Union with slab • Union with column (unlikely) • Union with both	• Union with panel • Union with column (unlikely) • Conditional on relationship between column width/height and panel width/ height; If <condition> then union with panel else union with column • Union with both
Beam		• Conditional on relationship between 1^{st} beam width/height/length and 2^{nd} beam width/ height/length; If <condition> then union with 1^{st} beam else union with 2^{nd} beam • Union with both	• Union with beam • Union with slab • Union with both	• Union with beam • Union with panel • Conditional on relationship between beam width/length and panel width/ length; If <condition> then belongs to panel else belongs to beam • Union with both
Slab			• Not allowed (report clash)	• Union with panel • Union with slab • Union with both
Panel				• Conditional on relationship between 1^{st} panel width/height/length and 2^{nd} panel width/ height/length; If <condition> then belongs to 1^{st} panel else belongs to 2^{nd} panel • Union with both

4.1.5. Benefits

This proposed solution addresses one way of defining the intersecting volumes and presents a unified way to resolve those volumes in reports and views. This allows one, fairly simple user interface to create different sets of rule for each functional need. The idea that the definition points of the actually modeled elements that are not modified but maintain throughout the process makes the implementation fairly easy and straight forward.

Note: In the description above, the process for generating connecting volumes was defined to include cutting of the shapes of the connected members to exclude any overlap between them and their associated connecting volumes. An alternative implementation would be to avoid performing this step at the time connecting volumes are generated. This would not change the final result of any representation, because the result of the union between overlapping shapes is the same as the result of the union between non-overlapping shapes: $A' = A \cup V = (A - V) \cup V$. Given that the representation is dependent on the view, the display list would likely be regenerated frequently in any event.

4.2. Manipulate Cross-sections

The geometry of cast-in-place structures is flexible. Most structural elements have rectilinear and planar geometry only because the formwork needed to cast them is easier to fabricate, but in many cases non-planar warped and curved surfaces are used. There are also no set restrictions on profile shapes for linear elements.

For these reasons, in many cases it is difficult to generate the geometry using simple prismatic extrusions. The geometry of the basic logical elements in the building model on the computer, such as columns, beams, panels, slabs etc., often needs to be easily edited to fit the desired design geometry. There may be many ways to achieve the desired geometry and it is often difficult to prescribe in advance what formwork will be necessary. No predefined catalog of standard profiles or shapes can suffice. Thus, the ability to control and manipulate the geometry of logical elements 'on the fly' during editing is essential. This section identifies some functionality that will enable effective geometry manipulation for cast in place reinforced concrete structures.

4.2.1. Glossary

Almost all BIM systems define primary definition points for the logical elements. In the following, we call these 'primary handles'. The following definitions introduce the additional concepts of 'segments' of elements and 'secondary handles' to manipulate them. All the definitions relate to Figure 89 and Figure 90.

- **Handle**: a point that defines the geometry of an element that can be moved/dragged by the user in order to manipulate the geometry.
- **Primary handles**: a set of handles that defines the location of a logical element, located at its ends or edges. Linear elements (beams, columns) have two primary handles representing the definitive start and end points of the element (called the 'start handle' and the 'end handle' respectively). Surface elements (slabs) have three or more primary handles (chamfers) that define the surface geometry.
- **Secondary handles**: handles that can be added to the element between the primary handles. For linear elements, all secondary handles will be located along

the element's axis. For surface elements, secondary handles can be added anywhere on the surface or the boundary.

- **Secondary boundary handles**: handles along the edges of a two-dimensional element. These are generally automatically generated from secondary handles.
- **Secondary area handles**: handles within the area of a two-dimensional element and within its defining plane.
- **Element segments**: a part of an element between handles. For a linear element, a segment may be defined by any two primary or secondary handles. For surface elements, each segment is defined by three handles (secondary or primary) in such a way that the element's surface is triangulated to favor generation of acute-angled triangles.
- **Cross-section**: the geometry that defines the extruded shape of a linear element. The cross-section is a planar poly-line placed at right angles to the axis of the element at a given point in the extrusion's centerline
- **Segment start cross-section** (for a linear element): the cross-section that defines the start geometry of the segment at its side closest to the primary start handle.
- **Segment end cross-section** (for a linear element): the cross-section that defines the end geometry of the segment at its side closest to the primary end handle.
- **Cross-section handles**: a set of points on a plane that define a poly-line which serves to define the cross-section geometry of a linear element.
- **Segment Thickness at Handle** (for a two-dimensional element): the physical distance measurement of the thickness of a slab or wall panel, measured perpendicularly to the plane that contains the referenced handle.
- **Segment Offset at Handle** (for a two-dimensional element): the offset from the plane to the start of the physical thickness of a slab or wall panel, measured perpendicularly to the plane that contains its handles.

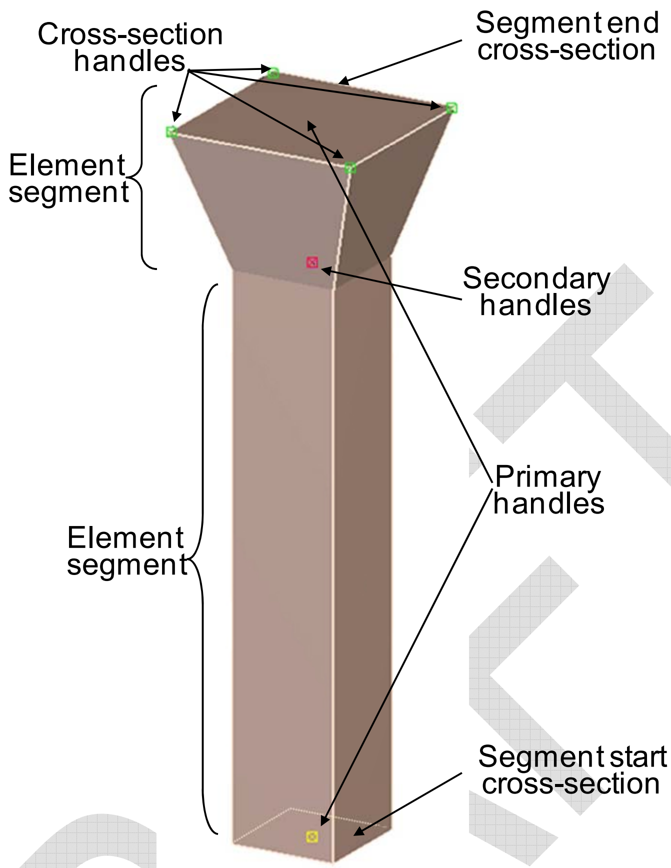


Figure 89. Glossary Terms

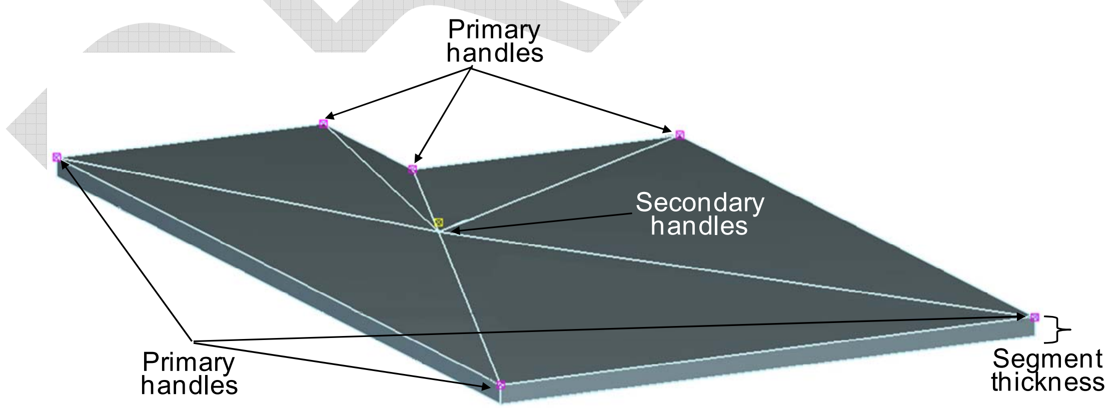


Figure 90. Glossary terms on

4.2.2. Proposed Shape Editing Extensions

The proposal has two main features:

- Editing of sketch profiles in place within a 3D extruded object.
- Manipulation at any handle point of a linear element's profile cross sections or a two-dimensional element's thickness and offset;
- Editing of an element's cross-section handles in place.

To manipulate a linear element's profile cross-section, the user should have a means to enter a 'profile edit mode' that will show the cross-section handles at both start and end handles, and any others created, as shown in Figure 91A. Once in profile edit mode, any cross-section handle can be moved within the cross-section's local working plane. Handles cannot be moved off the cross-section plane in the Z direction. The cross-section geometry can also be rotated create warped shapes.

Another function that should be available is addition of cross-section handles. The "adding" function will ask the user to pick two points, one on each cross-section Figure 91 B, to maintain correspondence between the number of vectors in each of the opposite cross-sections. Once those points are added, they can be moved independently of one another (Figure 91C).

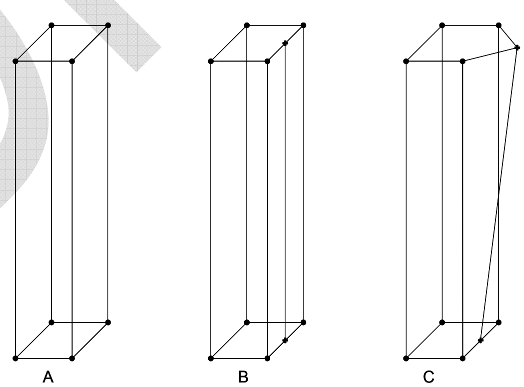


Figure 91. Adding points to a cross section

The ability to manipulate primary and secondary object handles consists mainly of the ability to add one or more secondary handles to an element. This feature will enable creation of complex geometries. For example, a column capital can be created as part of the column itself, instead of as a separate object. Another example will be the ability to warp the top surface of a slab for drainage.

The first step of the process is adding the secondary handle. As mentioned before, a secondary handle can only be created within the boundaries of an element. For linear shapes the secondary handle will be placed along the element axis anywhere between the primary handles (start and end points), as shown in Figure 92B. The location of the secondary handles should be stored either as a fixed distance from start or end point or possibly as a percentage of the overall distance. With the second method, the logical location of the secondary handle will be maintained with any change to the element's primary handles. For example, if a secondary handle was located at midpoint between the start and end primary handles, it will stay at the middle even if the overall height of the element is changed.

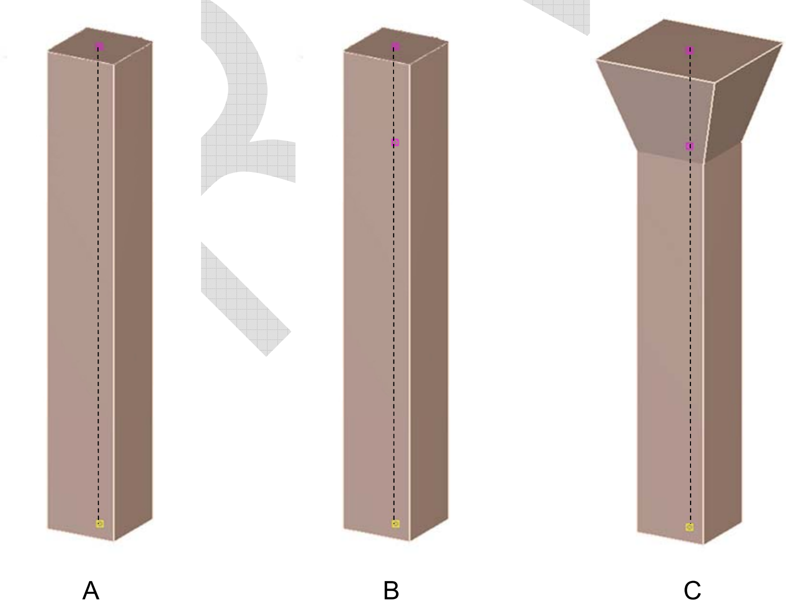


Figure 92. Adding a secondary handle

With the placement of a secondary handle along a linear element, a new segment is generated and the existing one modified. Each segment has a start and end cross-section.

The cross-sections on either logical side of the new secondary handle are by default equal to the cross-sections at the opposite ends of each segment. The new cross-sections can be manipulated, by entering profile edit mode and manipulating their cross-section handles, exactly as an edge cross-section would. The user should be able to choose to edit each cross-section separately or together as one cross-section. To simplify the process the user should be able to choose to change the geometry cross-section to be the same as that of the next cross section or the previous one. A generic way to accomplish this is to allow a switch for each segment cross-section to optionally be 'the same as' any other cross-section, whether a named catalog cross-section or a numbered cross-section belonging to this or a different segment of the element'.

In Figure 92C the end cross section of the first segment remained identical to the start cross-section of the segment (i.e. the start point cross-section). The start cross-section of the second (upper) cross-section also remained identical to the previous cross-section. The end cross-section of the second segment was edited to be a larger cross-section. The result is a single column object with a gradient capital.

In Figure 93 the start cross-section of the second segment was made equivalent to its end cross-section, both of which were changed to have a new profile. The end cross-section of the first segment remained equivalent to its start cross-section, creating a singular change in the shape.

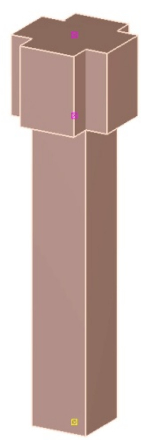


Figure 93. Example of secondary handle manipulation

The secondary handle can be moved, after it is placed, only along the shape's main axis. When a primary handle is moved, even after a secondary handle has been placed, the main shape axis will remain the axis between the start and end primary handles of the shape. The secondary handle should be moved with that axis according to its setting. If the secondary handle's location was entered as a fixed distance, this distance will remain the same after the change. If the location was entered as a percentage, the new location of the secondary handle will be proportional to its original location along the axis.

For surface elements, the secondary handles (or secondary chamfers) will be placed on the surface anywhere within or along the element's boundaries. There are thus two kinds of secondary handles: boundary and area. Boundary secondary handles do not define new geometry; they simply allow additional control points for raising, lowering or adjusting the thickness of the slab or wall. An area secondary handle will generate new segments that are defined by triangulating the surface of the face in which the handle lies with edges (Figure 94B). For simplicity, it the edges intersect with the boundary of the face (because of a concavity) the operation is denied and the user will have to partition the face differently. All three types of handles (primary handles, area and point secondary handles) can be modified with two values: the offset and thickness of the segment at the handle point. This functionality will enable creation of complex surfaces, such as warped top surfaces of a slab for drainage.

Primary handles of a two-dimensional element can be moved within the plane defining the element; area handless can be moved anywhere within the shape’s face structure; and boundary handles can only be moved along the boundary between two primary handles.

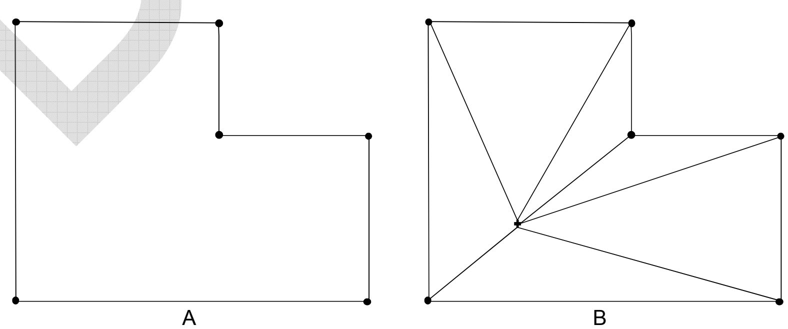


Figure 94. Surface element sub handles

If properly implemented, the above capability, with multiple partitions of faces, would allow the definition of fairly complex features, such as those shown in Figure 95 and Figure 96. An additional capability would be to be able to select a set of faces within such a construction for a particular instance and name the face set as a ‘feature’. This feature could then be copied to other slabs. In such a case, the feature must reside completely inside of a face within the slab. The edge faces (those not attached to another face in the feature) will have a set of vertices. These identify points to which the face hold the feature is triangulated when it is attached.

An extended implementation of ‘feature’ would be the storing of such face sets in a library, allowing them to be applied to new projects. Drop panels for a slab are a good example of a feature. When applying features, the user will first pick a face on the parent object and a location point where the feature will be placed. This can be defined either by picking an actual plane, or by using a relative position like “Front” or “Back” in relation to the local work planes of the parent element. Figure 95 shows a rectangular drop panel added to a slab, where the base and secondary profiles are identical. Figure 96 shows a pyramid shaped pedestal feature for a slab, such as might appear in the floor of a water reservoir where a column raises from the floor; in this example the base and secondary profiles are different to one another.

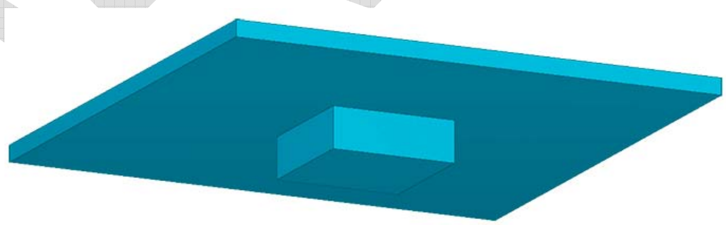


Figure 95. Drop panel for a slab

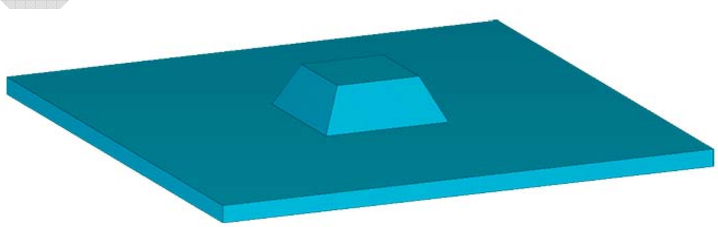


Figure 96. Pyramid column base

Once created, features might be parameterized, allowing sets of points to be edited together. For example, the features in Figure 95 and Figure 96 might be parameterized so the feature has a thickness, which defines the distance between the base and secondary 'profile'.

Because Boolean operations can be applied to a face, the capabilities outlined here can be used to define a wide variety of 'features' that would have wide application of RC modeling.

4.2.3. Benefits

By enabling the kind of manipulation described here, the user can easily create almost any desired shape. The elements resulting from the manipulations will still be one single object, with the intent that all other system functions should require only minimal changes. For example, the structure analysis model will reflect the desired structure (no extra nodes are added), and the distinct BOQ items will remain intact since we still dealing with one object. Reinforcement can still be applied to distinct logical element shapes, although they will have to be able to cope with multiple segments.

4.3. Pour breaks and pour stops

Pour breaks and stops are objects with surfaces that logically partition a CIP structure into sections that are cast in a single operation. They are logical representations, which mean that the original model elements – beams, slabs, columns, etc. – are not changed.

4.3.1. Glossary

- **Pour breaks** (PB): the physical boundaries of a pour as planned by the construction managers. They are always made in locations that satisfy structural engineering stability constraints, and must always completely divide a parent element into two distinct segments.
- **Pour stops** (PS): the actual locations at which concrete pouring was stopped during construction, where they are different to the planned pour breaks. In the majority of cases pour stops will be defined by horizontal planes, because that is the natural form of the surface of any concrete pour. Exceptions to this rule may occur when a physical barrier is introduced during casting if a problem arises and an unplanned stop is needed. In these cases, pour stops are defined by one or more planar surfaces.

4.3.2. A Solution

The geometry of a pour break or a pour stop is defined by a surface consisting of one or more planar sections (see Figure 97). The surface divides the parent element(s) into segments. The segments inherit properties from the parent elements with the exception of their physical shapes. A pour break must extend to the edges of the parent element to make a complete separation of the element it cuts into two distinct volumes.

There is a 'one to many' relationship between basic model elements and pour breaks; any concrete element may be split by any number of pour breaks. The first PB/PS placed will divide the parent element into two segments, and subsequent pour breaks will divide the segments in turn.

For example, for a linear element like a column, the first segment generated starts with the start primary handle cross-section plane, and its end plane will be the PB/PS plane itself. The second segment will start with the PB/PS plane and will end with the end primary handle cross-section plane. Each additional PB/PS will divide one segment into two new ones that eliminate the original segment.

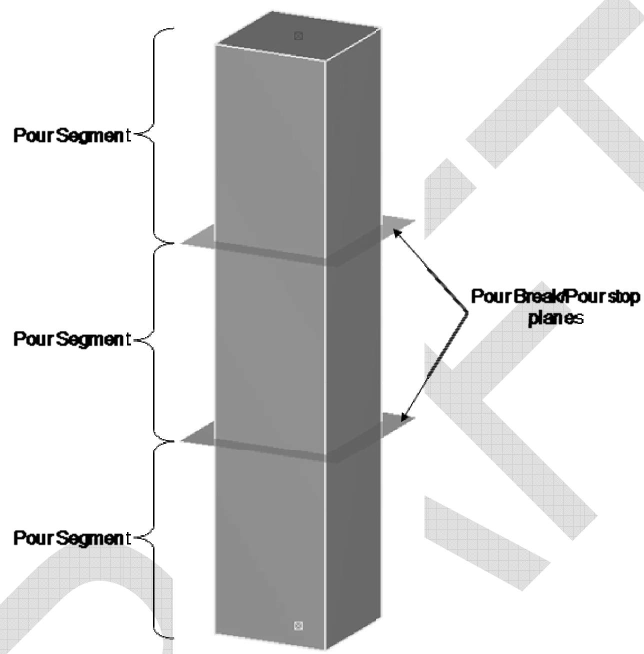


Figure 97. Column with two PB/PS planes

A collection of contiguous segments belonging to different elements is called a 'pour section'. Pour sections are defined in the 'Special Aggregations' section of this chapter. If a segment has no neighboring elements that will be cast with it, then it will be the sole member of a pour section.

PB/PS geometry is based on a polyline path that placed on a designated plane surface. The multi-plane surface is orthogonal to the designated surface along the polyline path. (see Figure 98). The PB/PS surface will extend through the entire element. The PB/PS line is editable, meaning that the user can change the location of the points creating the original path.

It should be possible to translate the PB/PS surface anywhere within the object's borders, which may be outside its original reference surface. However, some domain specific restrictions can be thought of that cloud restrict the freedom with which a PB can be moved. It would not be reasonable to expect the pour break surface to move beyond the extents of a reasonable pour break limit. For example, a column break should not create a segment of a size so short that the protruding bars, which will be included in a subsequent cast, cannot have the minimum length required to develop their full allowable stress. Similarly, it would be reasonable to limit upward movement of a PB in a column to the bottom level of any intersecting beams. For slabs and walls, the same would apply considering underlying beams, columns and transitions. For paving slabs, there would also need to be consideration of both and x and y breaks moving simultaneously. Note that no such restrictions can be applied to a pour stop position, as the placement of an 'as-built' divide cannot be predicted.

A PB/PS surface need not necessarily be perpendicular to the surface on which its' defining path is modeled. It needs to be possible to rotate a surface about any user-defined axis.

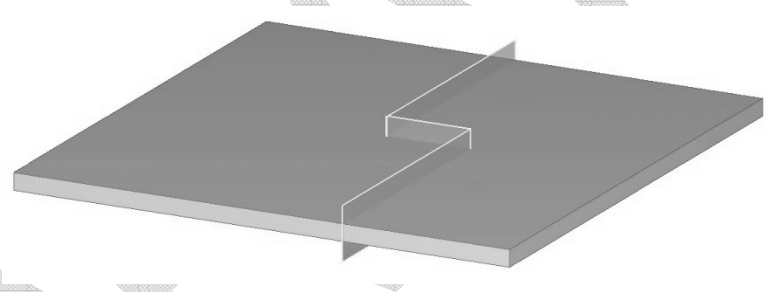


Figure 98. A slab with a multi-plane pour break surface

The user should define whether the plane is a pour break or a pour stop. The distinction is important because it determines how the system will behave with regard to any rebar or embeds that cross a PB surface when construction planning is done. The PS object is fairly simple and its only purpose is to divide the concrete volume of the parent elements into segments. The PS doesn't have any effect on the internal objects like rebars or embeds. PB, on the other hand, is more complex because its presence has an effect on the behavior of the internal objects.

As shown in section 4.3.5, pour breaks divide concrete elements into segments, and segments define the geometry for aggregations of them, the pour sections. When one or more pour sections are assigned to a work package, and the work package is scheduled with a casting date, the sequence of casting on either side of the defining pour breaks becomes known. At this point, any internal objects should be divided (rebar or mesh) or assigned (embed) to one or other segment. This means that the system should determine the order in which the segments are poured by assigning '*First*' and '*Second*' tags to each side of the PB surface as shown in Figure 99. Note: a special case occurs where a PB surface is horizontal, the lower segment is to be cast first and the upper second. This order is technically highly unlikely, if not impossible; if a user determines a work package sequence in which the order is reversed, the system should alert the user to the situation.

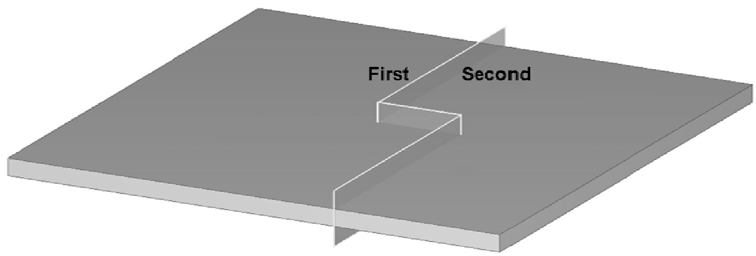


Figure 99. Defining the order in which segments are to be cast

After casting order is assigned, the user can define to which segment the internal objects belong. This definition affects only those objects that intersect with the PB plane. The internal objects should be divided into classes like rebars, embeds, etc. and each class could be associated with a different segment. Another property for each class is the clearance distance from the PB plane. The PB will relocate the specific object according to that clearance.

A more sophisticated function is '*Auto-splicing*'. If enabled, the user can define a splicing distance (as a constant default value or as a parametric value to be calculated using a formula); the rebars that intersect with the PB plane will then be spliced automatically according to that distance as shown in Figure 100.

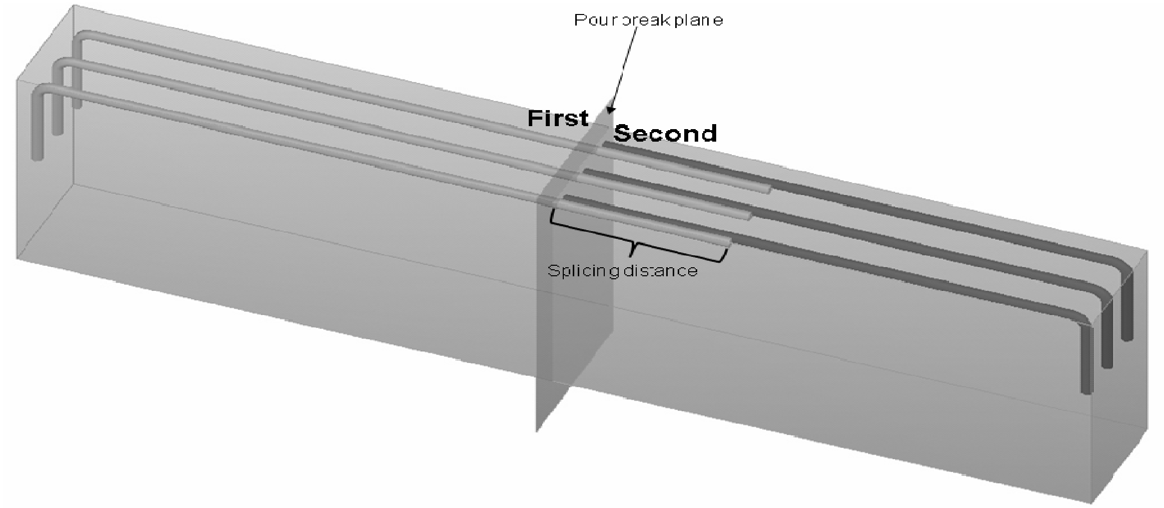


Figure 100. 'Auto splicing' in a beam

When moving a PB plane the 'Auto splicing' function and the clearance distance for embeds should be recalculated and move or change the internal objects accordingly.

4.3.3. Assigning physical parts to pour breaks

In many cases, pour breaks are implemented by building physical barriers to the flow of concrete. Where necessary according to user choice, it should be possible to assign physical parts to the surfaces of pour breaks. These parts would be included in any BOM prepared for a pour work package.

4.3.4. Visualization

The user should be able at any time to request the system to display the structure as pour sections, which may or may not be color-coded. As stated in the 'Special aggregations' section of this chapter, the pour segment geometry is derived automatically by the system and it follows that contiguous segments can be collected automatically into pour sections. The user should then be able to select pour sections and assign them to work packages.

Once pour dates have been assigned to work packages, the system should provide an automated 4D animation display of the pouring sequence for the structure as a whole.

4.3.5. Object Schema (Express-G)

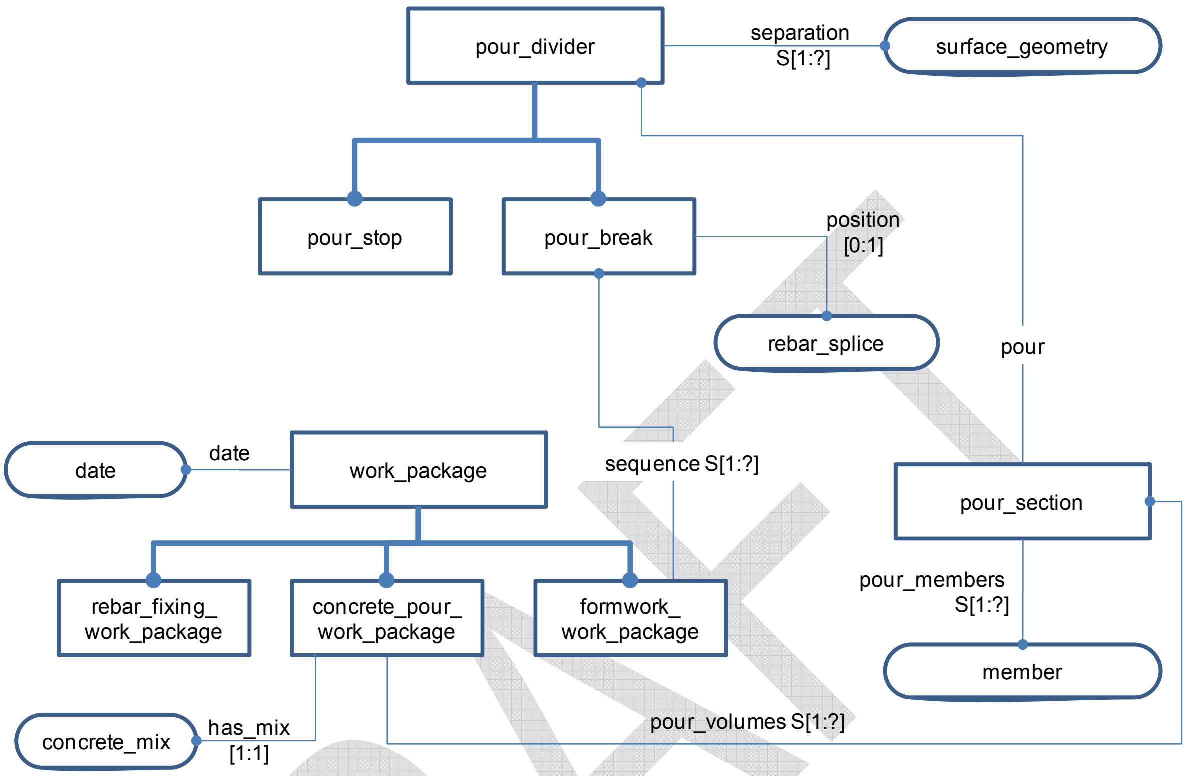


Figure 101. EXPRESS-G diagram for representing pour breaks and pour stops

4.3.6. Benefits

Pour breaks are the basic geometric definition that is needed to define pour segments, and consequently pour sections. They are fundamental constructs for a system that enables scheduling, procurement, logistics and production control for cast-in-pace reinforced concrete structures. Once pour breaks have been defined, the system can identify pour segments and pour sections, leaving the user only the task of defining the work packages and their sequence. With these basic objects in place, it should also become possible to consider an 'intelligent' function capable of proposing a reasonable casting sequence for an entire building, even placing sensible pour breaks, according to a set of rules.

Pour stops are needed because construction does not always proceed exactly as planned. Where pours are not completed as planned, placing pour stops should enable construction managers to very quickly calculate the quantities of concrete and other materials needed for the remaining pour. If needed, pour breaks could then also be adjusted, resulting in immediate feedback about the new construction plan to be followed.

4.4. Special Aggregations

4.4.1. Motivation

In earlier chapters, the fundamental difference between cast-in-place concrete, on the one hand, and almost all other structural types (steel, precast concrete, timber), on the other hand, was described. It is that CIP structures are monolithic, without any clear break between its basic structural elements. That continuity has unique expression in the ways in which CIP structures are conceived, analyzed, reinforced, planned for construction, and built. Many of these operations are performed on specialized groups of objects. However, the standard collection objects available in BIM software cannot fulfill all of the needs, because each type of group has distinct behavior. For example, a continuous beam is not simply a chain of co-linear beams with common consecutive end and start points. Rather, it has its own distinct behavior in terms of structural analysis and reinforcement detailing.

Dedicated and specific software objects are needed for modeling these constructs. They include objects to represent continuous beams and columns, steel-concrete composite sections, rebar fabrication cages, contiguous volumes of concrete that are poured together, and work packages.

4.4.2. Continuous Beams and Columns

Both beams and columns have different structural behaviors when they are continuous than when they are distinct objects. For structural analysis, it is sufficient to define that the 'connections' between consecutive members of a column stack or a row of beams carry moments across the joints. For reinforcement detailing, however, different patterns of rebar are used when consecutive beams or columns are either present or absent. As a minimum, it should be possible to group a stack of columns or a row of beams in a group construct that carries an identity as a column stack or as a continuous beam. Ideally, the system should automatically determine such groups and apply a grouping object.

The group object should be available internally for functions like 'reinforcing layout', or exported to external analysis and possibly other routines. In the case of reinforcing layout, this approach would allow correct placement of top rebar over supports to cope with negative moments, without having overlap of separate rebar sets, each placed independently for neighboring beams. Similarly, a column reinforcing function could then easily determine the correct relationship between the extensions of longitudinal column starter bars above the end of a lower column and the main rebars of the column above, ensuring by design that they don’t clash.

4.4.3. Steel and Concrete Composite Elements

As described in section 3.2.6, composite elements are generally made by casting steel profile sections into concrete elements or by filling tubular steel profiles with concrete. They may also be two-dimensional, with the most common application being casting of concrete slabs on steel decking that becomes part of the slab's reinforcement. Here too, the behavior of the whole is different to the behavior of the parts. While it is possible to model the physical geometry of these elements by modeling the concrete and the steel components separately, separate modeling cannot capture the correct behaviors and is also inefficient.

One possible approach to modeling linear composite elements is to enable cross-section definitions with heterogeneous materials. A steel profile cross-section can be embedded in a concrete beam cross-section, and then the composite cross-section could be extruded together. However, this would have the drawback that both the concrete and steel parts would have to be of the same length. Greater freedom and flexibility would be offered by an alternative second approach, in which a 'composite cross-section group' could be defined. The group could be extruded and manipulated together, but the main steel and concrete elements would retain their individual geometric parameters. Thus they could also have different lengths if needed, with the steel member fully encased in the concrete, or protruding from it. Another way of achieving this is to allow internal elongation of an element, anywhere in its midsection. This would allow member ends to be defined, then the middle elongated and easily edited without affecting end detailing.

Regardless of which approach is taken, specific analysis functions should be associated with the composite object rather than with the individual objects. In an object-oriented programming environment, only the composite objects' physical properties should be exposed to analysis methods. As suggested for continuous beams and columns, the BIM software should automatically identify composite steel and concrete elements, automatically generate the composite object, and expose its properties rather than those of the individual elements to functions that reflect the aggregate behavior.

4.4.4. Rebar Cages

A special aggregation object is needed to collect rebars into 'cages' that can be pre-assembled. Rebar cages do not necessarily have a one-to-one correspondence with the concrete elements to which they belong. Cages can be tied that cover multiple elements, or that fulfill only a part of a whole element's reinforcing. Some of the bars within the general volume of a cage may need to be excluded from its' prefabrication, for reasons of constructability.

Thus a rebar cage for prefabrication is different to a set of rebars applied by a detailing function. It must be possible to define a group of rebars as a 'rebar cage', and report bending schedules and BOM for each cage. Not only logistics and management functions need to operate on a cage; engineering calculations are also needed. For example, it may be necessary to calculate the center of gravity of a cage in order to properly locate crane lifting points.

4.4.5. Pour Sections

A 'pour section' is another aggregate object that BIM software should be able to identify automatically and represent in a collection object. A pour section is a contiguous section of concrete that may include all or part of the physical concrete volume of one or more building elements. As shown in Figure 102, the boundaries of a pour section are made up of:

1) the physical surfaces of the member elements,

2) pour break and pour stop objects. Both of these new objects are defined in the section 'Pour breaks and stops' earlier in this chapter.

A pour section is different to a 'cast unit' in that it is not simply an aggregation of basic model objects. Cast-in-place structural elements may be poured in a series of distinct concrete pours, and so they cannot be modeled as belonging exclusively to any one pour section. Unlike the basic elements, the extents and boundaries of a pour section may be determined late in the process and the definition may be subject to change as construction site conditions develop.

The software should be able to evaluate the extents of pour sections whenever required. If a pour break is added or removed, the new pour sections that result should be re-evaluated. Pour sections should be visible to the user on request, for example as shaded sections of concrete. The pour section must be able to collect and report a list of all of the rebar, embeds and any other parts needed for its fabrication, including the volume of concrete. The properties of a pour section are its volume and identity. The project management attributes, such as production dates, are not stored in the pour section object; instead, one or more pour sections are associated with a work package, which is the basic unit for scheduling.

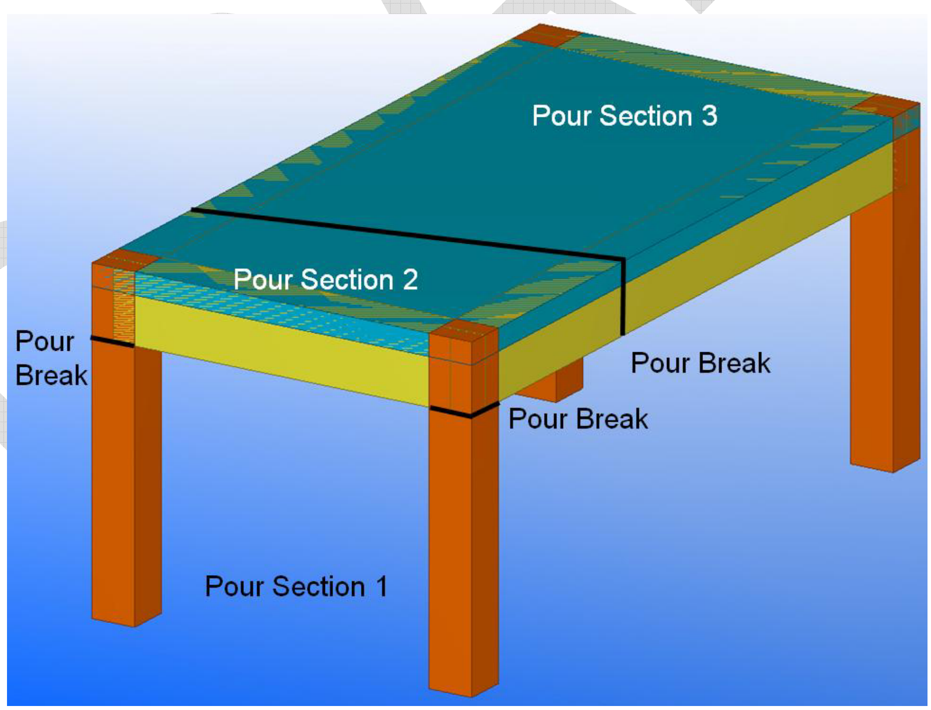


Figure 102. Work packages for concrete placing, composed of construction object sets defined by pour breaks.

A concrete pour section within the RC BIM model carries significant data. It includes:

1. A pour ID
2. The concrete material quantity (volume), type, slump and admixtures.
3. Additional materials needed (BOM of rebars, embeds, waterproofing etc.)
4. Surface preparation (may be stored alternatively with the formwork surface objects)
5. Finishes for formed surfaces (may be stored alternatively with the formwork surface objects)
6. Curing process.

Attributes 4-6 and the material type should be references to entries in look-up tables that are compiled for the project as a whole, or at the level of a company's standards. Additional properties that relate to the concrete pour will be associated with the work package, rather than with each individual concrete pour section. These are:

- Pour date
- Air and concrete temperatures
- Reference to cylinder test sample or other test information
    - Standard Compression Test
    - Slump Tests
    - Air Content Tests
    - Core Test
- Delivery tickets

Attributes such as finishes for unformed surfaces should be stored with the physical members attributes.

When pour breaks are moved, all of this data is regenerated except for the pour ID, because the pour ID may already be associated with a work package. Thus the pour ID must continue association with the volume between the changed pour breaks. It also must be possible to insert a new pour break between existing ones, in which case a new pour ID would have to be generated.

When a work package (see below) for pouring concrete is determined, the user should select which pour sections are included in it. It should then be possible to query all of the objects for all of the volumes (pour sections) included in that work package for BOM purposes; embeds, reinforcing, concrete CY, waterproofing, etc. The concrete BOM should tell the user what concrete and how much is to be used for each part of that pour. For example, walls are 3,000 PSI, columns are 5,000 PSI, slabs-on-grade are 3,500 PSI and so on. Further, if engineer-approved mix designs are to be used, the approved mix numbers should be provided on the report, e.g., Carolina Ready Mix #N563J. This number will be used when dispatch is called and the order is placed. The need for any admixtures should also be identified if this need is not already denoted by the use of a mix number.

The user should also be able to track pour performance on a pour-by-pour and by object basis. The system must allow the user to input the amount of concrete used, the date, the delivery ticket number, the labor-hours used to complete that pour, the weather during the pour, etc. It may also be necessary to link to images of the delivery tickets to the objects for record keeping purposes. The system should also be able to generate over/under reports and visualize the pour performance. For example, all pours that were less than 2% over the quantity survey will be in green, 2 - 4% in yellow and 4% + in red. As the incremental cylinder break results are received, the user should have the ability to associate those results with model objects and then generate reports based on the break results. The user should also have the ability to visualize the break results. In the event of low test results, this will aid in devising an alternative work path for subsequent work activities.

4.4.6. Work Packages

Simply associating 'construction date' or other similar user-defined variables with physical structural elements is insufficient for project management, because building construction is not managed 'piece-by-piece'. The relationship between physical elements and their management attributes must be objectified in order to aggregate elements, parts of elements, and temporary equipment such as formwork. This new aggregation object is a 'work package' identifying the elements that are built together.

Work packages are essential for detailed construction planning and management because they aggregate project information at the level for which work is actually assigned and performed by individual crews. They are defined at a finer degree of detail than tasks in a CPM master plan or monthly look-ahead plan. Conceptually, they aggregate activities performed on individual components or parts of individual components. For example, all of the formwork for a specific set of concrete columns and fixing the steel for them may make up a single work package. Work packages are assigned to particular crews, and are also the basic unit for 4D CAD animation and visualization. The defining attributes of a work package are:

1) A '**Construction Object Set'.** For cast-in-place concrete, a construction object set is a group of RC building objects (columns, walls, beams, etc.), parts of building objects, faces of objects or parts of faces of objects, and sets of reinforcing bars. Construction object sets allow allocation of crews to the different stages of work needed for their production (formwork and shoring, reinforcement, placing concrete, stripping, etc.). Formwork is assigned to a set of faces or regions of faces (see Sec. 4.5), steel fixing is assigned to discrete sets of rebars, placing concrete is assigned to regions of a building defined within pour break boundaries, etc. A graphical interface is needed for collecting objects or parts of objects into construction object sets, and the user should be able to confirm visually what has been selected. Note that 'construction object sets' can be considered roughly equivalent to 'cast units' in precast concrete construction.

2) One or more **'Work Types'**. These are categories of similar undertaken by a crew. Work types include formwork and shoring, Steel Fixing, Concrete Placing, Stripping, etc.

3) One or more **'Resources'**. These are the human and equipment resources needed to execute the work package, such as formwork, crane time, and work crews. They are measured in hours (or days). At a minimum, the software should compute the raw quantity of work for each, so that if base work rates are provided, the amount of time required for each can be computed. Different resources have different types of cost: crews consume both time allocation and dollars; rebar is consumed, formwork is a time allocation. These need to be associated with the objects in the work package.

4) A list of **Materials**. Materials include concrete, rebar and any embedded components that are consumed. The software should compute and compile the list of materials needed for any work package automatically, so that they can be ordered and delivered to the work face.

5) A set of **Preconditions**. Preconditions cover all of the previous work, space, access, permits, climate, information, and any other conditions that must be met before the work package can be performed. A standard checklist could be provided according to work types, and/or users should be able to add their own specific conditions.

Work packages are the basic unit needed for production planning and control. Once they are defined, software – whether external to or embedded in the BIM modeling software[5] – can support weekly work planning (assigning work to crews) and procurement of materials directly to site in exactly the quantities and configurations needed. An important aspect of this is versatility, in terms of empowering a production manager to respond rapidly to change. For example, a decision may be made to delay pouring concrete for a particular area of a slab, because some electrical fitting that must be embedded was not delivered, or if notification of a design change was received for a particular column among a set of columns. In these cases, the user should be able to simply move a pour break, remove a face or region of faces from the construction set, or remove or add rebars to the construction set, and expect the software to update the work packages accordingly. In this way the software would exploit the BIM model to accurately and automatically change procurement orders and quantities, work assignments, etc.

[5] The degree of detail of production planning that can be supported by the work packages as defined here is compatible with the Last Planner™ method of production control for construction. This method is the most common implementation of Lean Construction principles. Although there are some software applications available for this method, none of them enable any kind of link with a building model, which means that all of the various quantities for work packages have to be calculated and entered by the user, and that any changes to the extents of a work package demand recalculation of the quantities in the same way. Linkage with the project model facilitates lean construction.

A work package that includes pouring concrete may have one or more 'volumes' of structural members, or parts of members, associated with it. Each of these volumes is called a 'concrete pour section' and is demarcated by the physical boundaries of the members, by pour breaks and stops. Pour sections are defined above; Pour breaks and pour stops are defined in section 3.6.1 and in the section 'Pour breaks and pour stops' in this chapter.

4.4.7. Object Schema (Express-G)

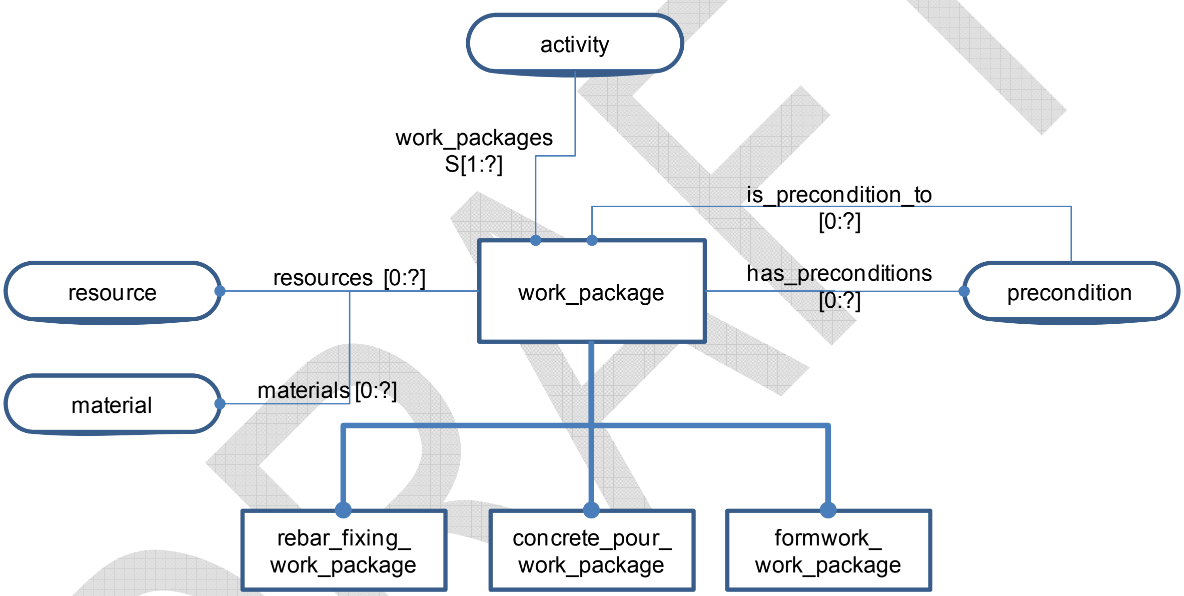


Figure 103. EXPRESS-G diagram for representing work package

4.5. Formwork Planning

4.5.1. Objective

Formwork and finishing of cast in place concrete structures is an important part of any work package and an intrinsic aspect of planning and execution of RC. Like other aspects of this task, formwork and finishes need to be objectified so they can be acquired or fabricated, placed, stripped and scheduled.

Some formwork is modular and manufactured. This includes Symons, Gates, SIA, Peri, Titan and others. These systems are often leased for a project and include both a formwork system and finish options for the exposed concrete surfaces. Where possible these systems should be entered as library objects into the Tekla system. Providing a service (possibly through a third party software vendor) would facilitate development of such libraries. Some form of guide to use these pre-formed elements within Tekla would facilitate their use. It might provide guidance of how to set up grid lines or other methods to make such layout simple for users. Often the formwork also includes other support systems such as shoring or cross ties.

Custom formwork is often used, whether the requirements are simple planar surfaces or more complex cylindrical, lofted (straight in one axis) or multi-curved. Custom formwork is leased or fabricated for the project and is designed to support its requirements. The design of formwork is outside of the scope of this specification, but Tekla should anticipate that the development of a formwork design system will be an indicator of success in the RC market.

All RC surfaces in a project must receive formwork, except:

- ground embedded RC, where earth serves as the form
- self formed construction where permanent walks or steel decking provide the formwork
- top surfaces that remain open

The formwork needs to be defined as distinct objects capable of being allocated and tracked over time. These objects may range from modular pre-defined forms to shapes extracted from the RC design by subtracting a surface shape from a block using the Boolean operations.

4.5.2. Formwork Scheduling

Formwork scheduling is an important aspect of RC construction. It identifies when the formwork is placed and must be coordinated with reinforcement and embed placement. It also identifies curing periods and then stripping activities. Management and scheduling requires existence of a formwork project library. The formwork library identifies what formwork pieces are available to be assigned to work packages.

Formwork scheduling identifies which formwork piece is allocated to each RC surface in each work package. It also manages the temporal allocation of each formwork piece in the formwork library to make sure that there are no overlapping temporal assignments. A checking algorithm could geometrically validate that every required RC surface had an associated formwork piece. Alternatively, formwork allocation could be reviewed graphically, by color coding the formwork assignments to the RC surfaces, with a different color identifying surfaces with no assignment. In some cases spatially overlapping assignments may be an issue.

The functional implementation of formwork scheduling is the allocation of formwork from the project formwork library to the required RC surfaces. The formwork should match the complement of a segment of the RC surface. That is, the formwork is supposed to match a portion or all of one or several shape faces. A proper assignment check is required. That is, some level of surface-to-surface check is required upon allocation. We believe this check can be approximate, for example edge-of-form matching to the surface of the RC.

Formwork scheduling must address both coverage of all required surfaces and also the allocation of re-usable forms over time. We envision an effective scenario to be carried out in the following steps:

a. allocate forms to a RC surfaces in a large section of work, without consideration of schedules

b. then coordinate schedules, based on construction sequences and rate of work within a single pour-strip period.

The formwork library needs to be defined as distinct objects capable of being allocated and tracked over time.

4.5.3. Finishes

Finishes in RC have different requirements:

1. defined by surface molds within the formwork. such as wood strips, metal corrugations or boulder-shape molds
2. defined as a process, such as sand blasting
3. the addition of additional material to the RC, such as surface finishes and painting/

Each of these types of finishes must be supported.

Formwork elements have coherent relation with pour breaks and pour stops because formwork configuration is determined by schedule and pour section of pour breaks and stops. Therefore, formwork definition should be linked to definition for work package.

Benefits

Currently, formwork is normally planned by coloring building elevations and doing the layout and scheduling manually. We are not aware of any software package that allows the planning and management of general formwork. This is an extremely time intensive task and the development of such a package would have large productivity benefits. There are no competitors in the marketplace.

5. Summary and Conclusions

This specification has defined the issues needed to develop an advance BIM tool for CIP RC structures. Reinforced concrete structures have many conceptual differences from a building modeling perspective compared to steel and precast concrete structures. The most important differences between CIP and steel/precast structural systems include the physical characteristics, structural connections, integration with other building systems and the method of fabrication.

Implementation of the functional requirements in building information modeling (BIM) tools would provide the CIP industry, designers and contractors alike, with the ability to model its structures from the early stage of conceptual design through detailing and up to and including construction and maintenance. The specified capabilities would enable this classical example of on-stie fabrication with the full advantages of building information modeling, improving reinforced concrete productivity, and quality.

These functional requirements were derived through three stages which are composed of (1) definition of process model, (2) definition of user requirements, and (3) a functional specification that addresses new capabilities identified in the user requirements. Especially, consortium members such as major construction contractors, engineering companies and research institutes took part in order to reflect requirements of the various roles of different actors within the RC construction field.

This specification provides functional requirements for specific areas that require definition and development for BIM of CIP RC structures. The areas include work packages, pour breaks stops, overlapping connecting volumes, object manipulation beyond extruded profiles and management of formwork surfaces for production. In particular, monolithic shape data can be handled by their embedding structural volumes. This specification defines various kinds of connecting volumes for different structural functions. It then deals with member connections. In addition, the specification provides an algorithm for derivation of connecting volume.

APPENDIX A

Embed sample drawings

EMBEDDED PART NO.	QTY	A	B	C	D	E	F	G	H	J	K	L	M	N	[redacted]SI NO.
10086	1	3'-4 5/8"	4'-0"	2'-0"	1'-0"	3/4"	2'-0"	1'-4 3/4"	1'-0"	1'-0"	1'-6"	1'-6"	1/2"	1/2"	38310

NOTES

1. REF. TO 21203-102-01-SS-[redacted] FOR SPECIFICATION.
2. UNLESS OTHERWISE SPECIFIED TOLERANCES ± 1/8.
3. CONTAINMENT WELD.
4. DIMENSIONS ARE IN INCHES WHERE NOT OTHERWISE SPECIFIED.
5. TOLERANCE ON ANGLES ± 0°-30'
6. FLANGES ARE ALL SAME SIZE.

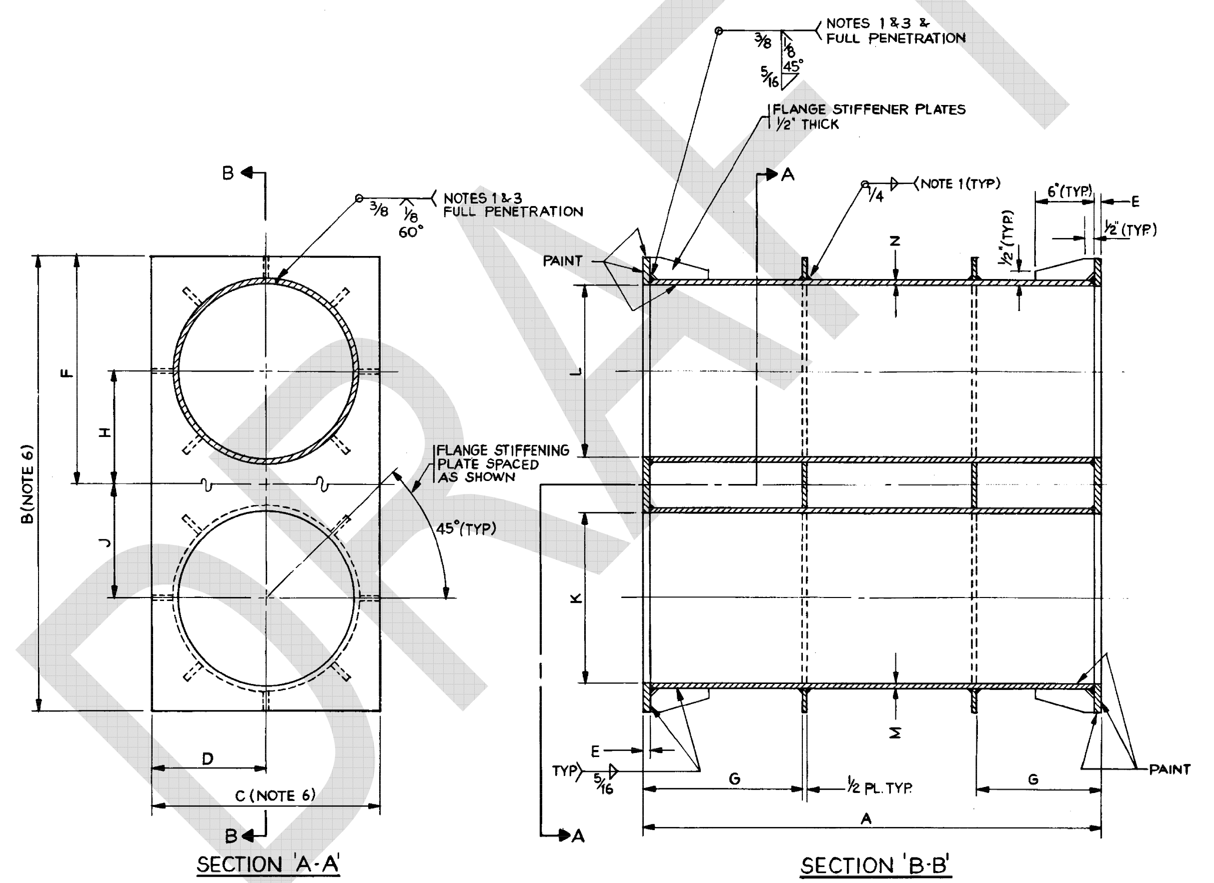


Figure 104. Sleeve penetration in containment wall

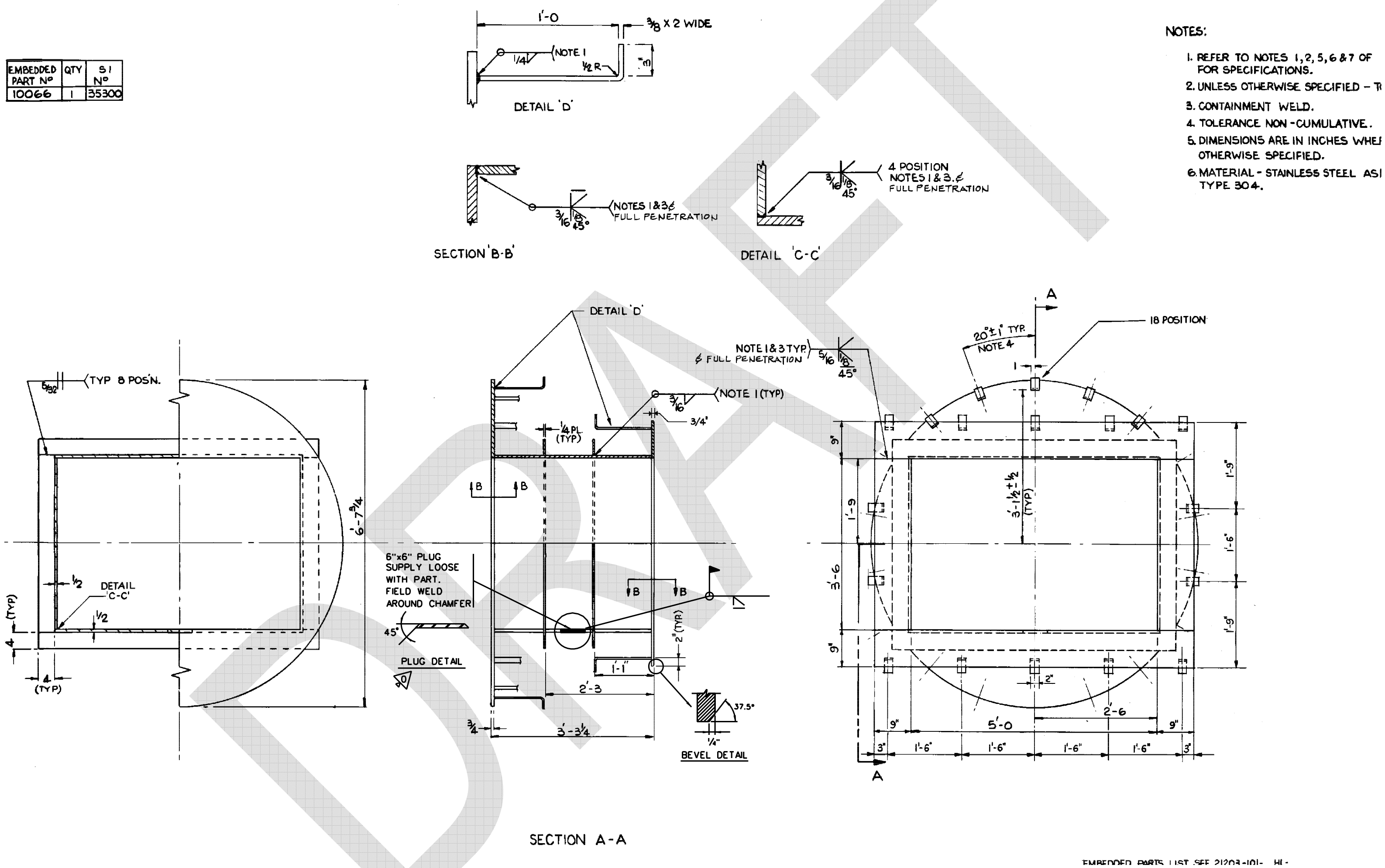


Figure 105. Embedded frame

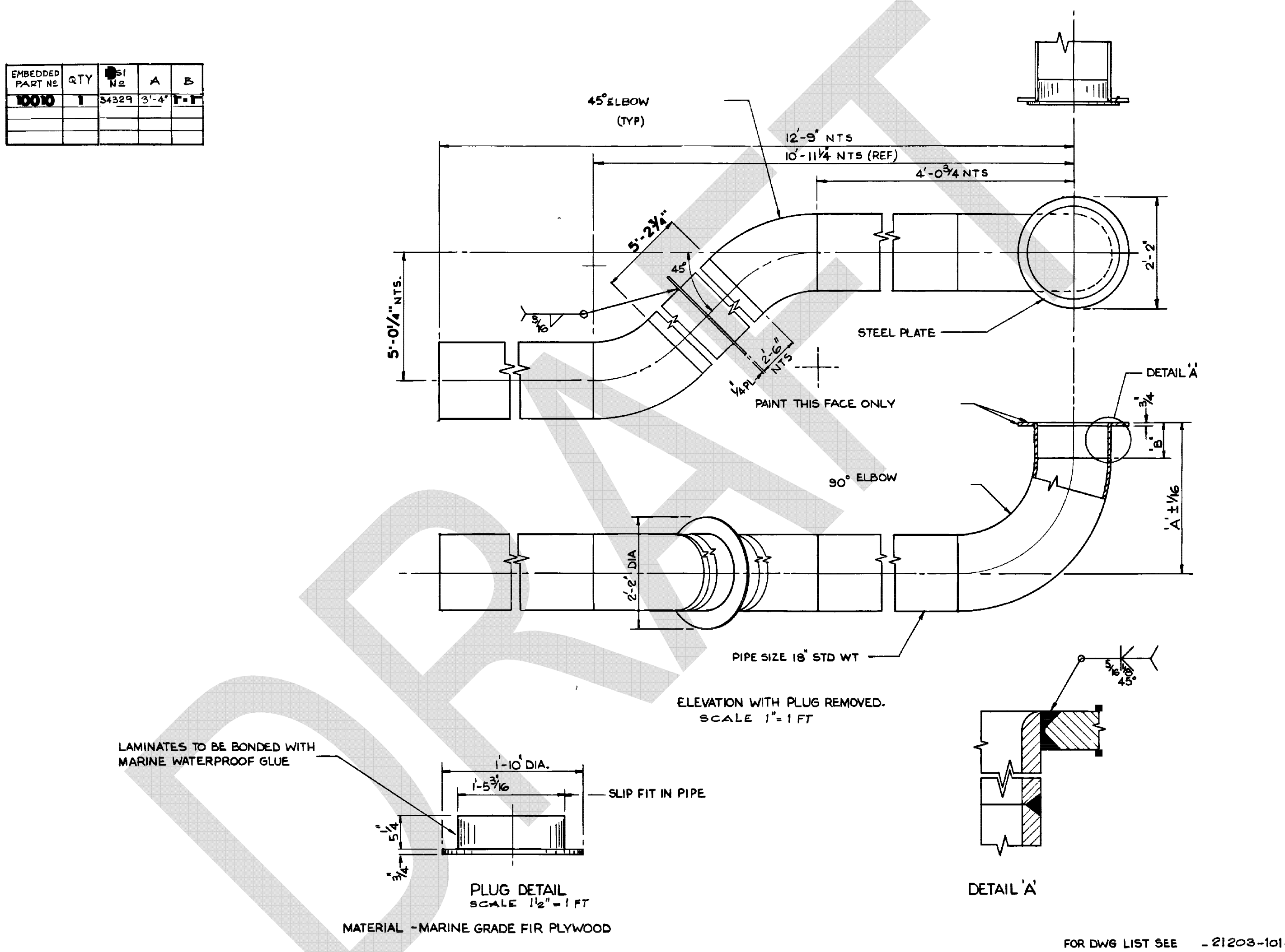


Figure 106. Embedded pipe

EMBEDDED PART Nº	QTY	A	B	C	D	STRAP ANCHORS QTY.	[redacted]SI Nº
10542	1	9'-8	3×3×1/2	10	1/4	9	31710
10543	1	8'-8	3×3×1/2	10	1/4	8	31710
10544	1	9'-3	3×3×1/2	7 1/2	1/4	9	31710
10545	1	9'-3	3×3×1/2	7 1/2	1/4	9	31710

NOTES:

1. REFER TO 21203-102-1-SS-[redacted]. FOR SPECIFICATION
2. UNLESS OTHERWISE SPECIFIED - TOLERANCES ±[redacted]
3. ANCHORS TO BE STAGGERED AT 1'-0 (APPROX) CEN
4. UNLESS OTHERWISE SPECIFIED - DIMENSIONS ARE IN INCHES.

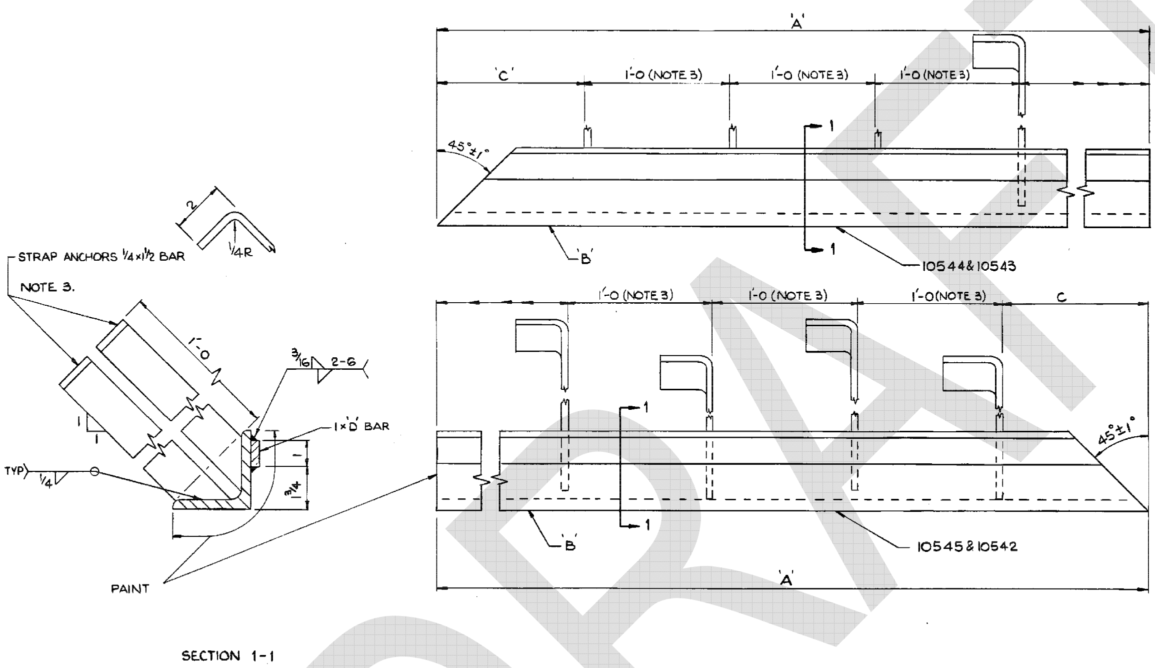


Figure 107. Edge angle

EMBEDDED PART NO.	QTY.	BSI NO.	QTY. BOLTS	BOLT SIZE	BOLT TYPE	DIMENSIONS ARE IN MILLIMETERS 'A'	'B'	'C'	'F'	'G'	'J'	'N'	'O'	'T'
12018	3	71340	16	M24x305	A307 GR.A	810	50	810	270	50	270	910	910	60
12019	1			M24x305		1110		1110	370		370	1210	1210	60
12361				M24x265		630		690	210		230	790	730	50
12362						795		795	265		265	895	895	
12363						630		690	210		230	790	730	
12364						795		795	265		265	895	895	

NOTES:

1. REFER TO 21203-102-1-SS-D FOR SPECIFICATION
2. UNLESS OTHERWISE SPECIFIED - TOLERANCE ± 3 (1/8").
3. UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN MM.

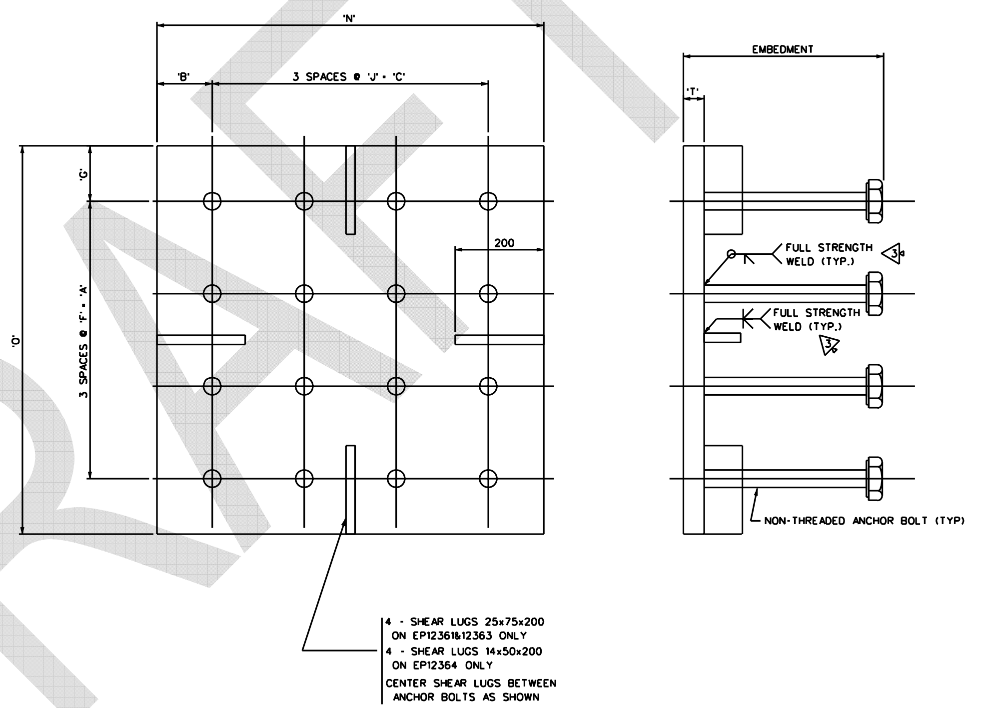


Figure 108. TYP embedded plate

DIMENSIONS ARE IN MILLIMETERS																
EMBEDDED PART NO.	QTY.	BSI NO.	'A'	'B'	'C'	'D'	'G'	'H'	'K'	'L'	EMB. 'M'	'N'	'O'	'T'	PROJ (MIN)
51077	1	38420	114	15	150	12	15	14	20	415	235	180	144	20	175
51078	1	38420	114	15	150	12	15	14	20	415	235	180	144	20	175

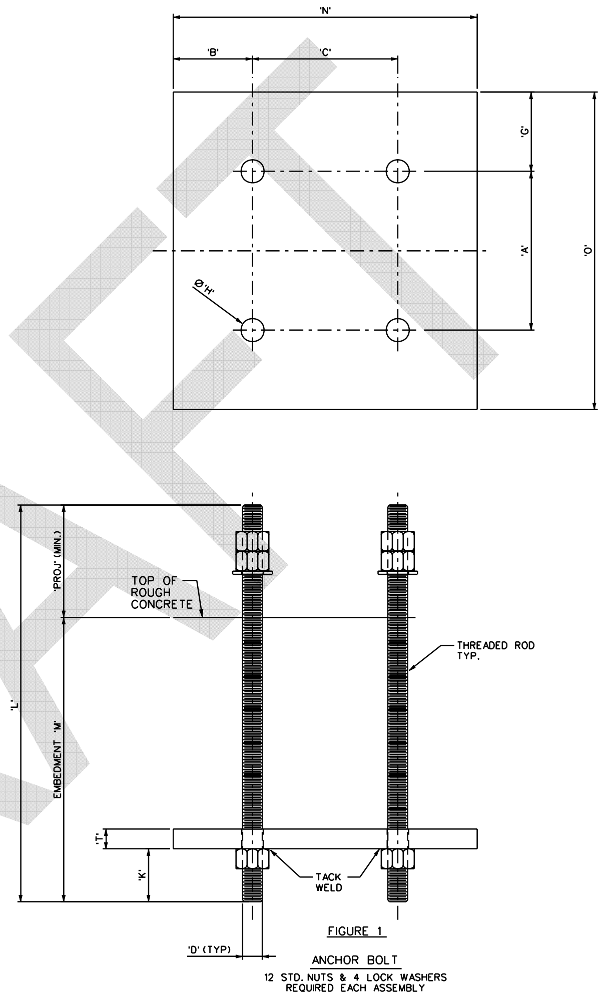


Figure 109. TYP anchor bolts

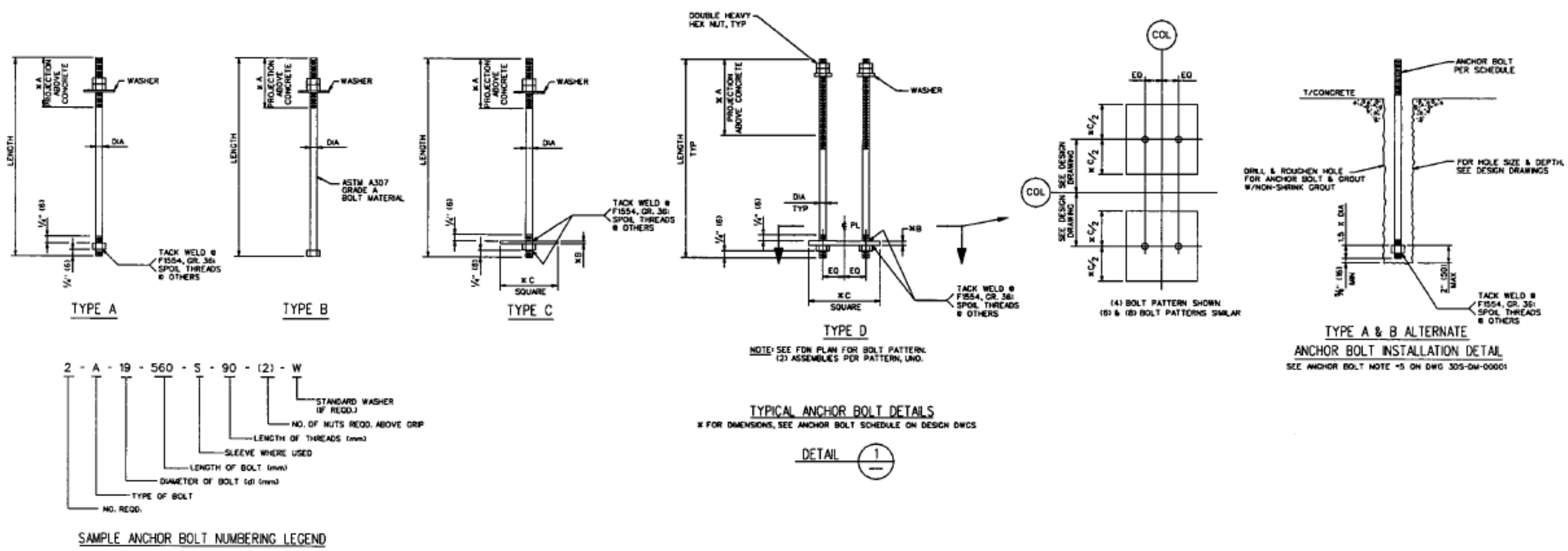
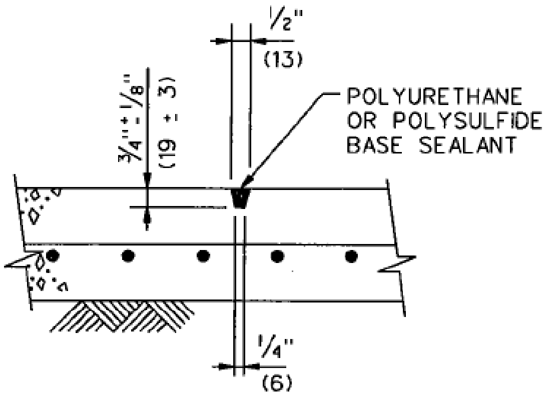
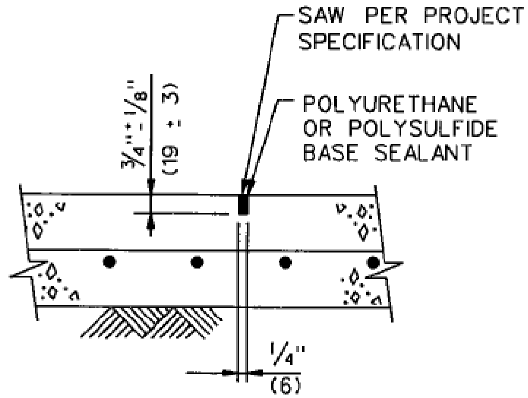


Figure 110. TYP embedded plate



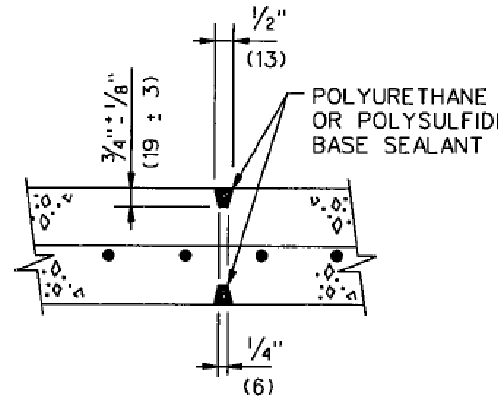
FORMED

DETAIL 9



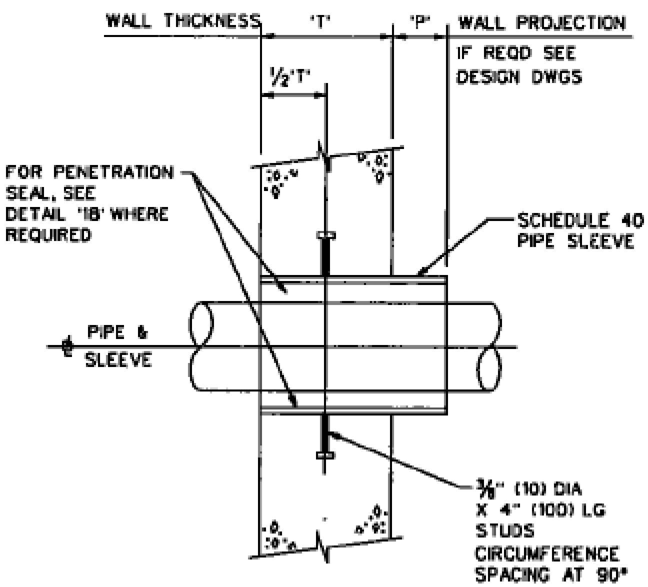
SAWN

DETAIL 10



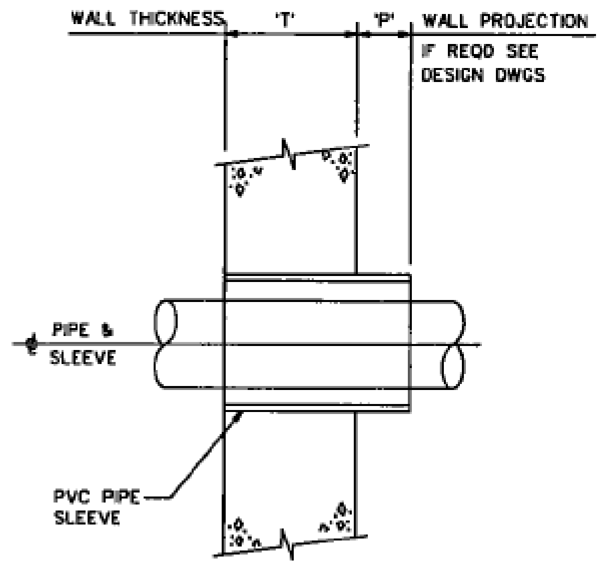
FORMED AT WALL

DETAIL 11



PENETRATION SLEEVE

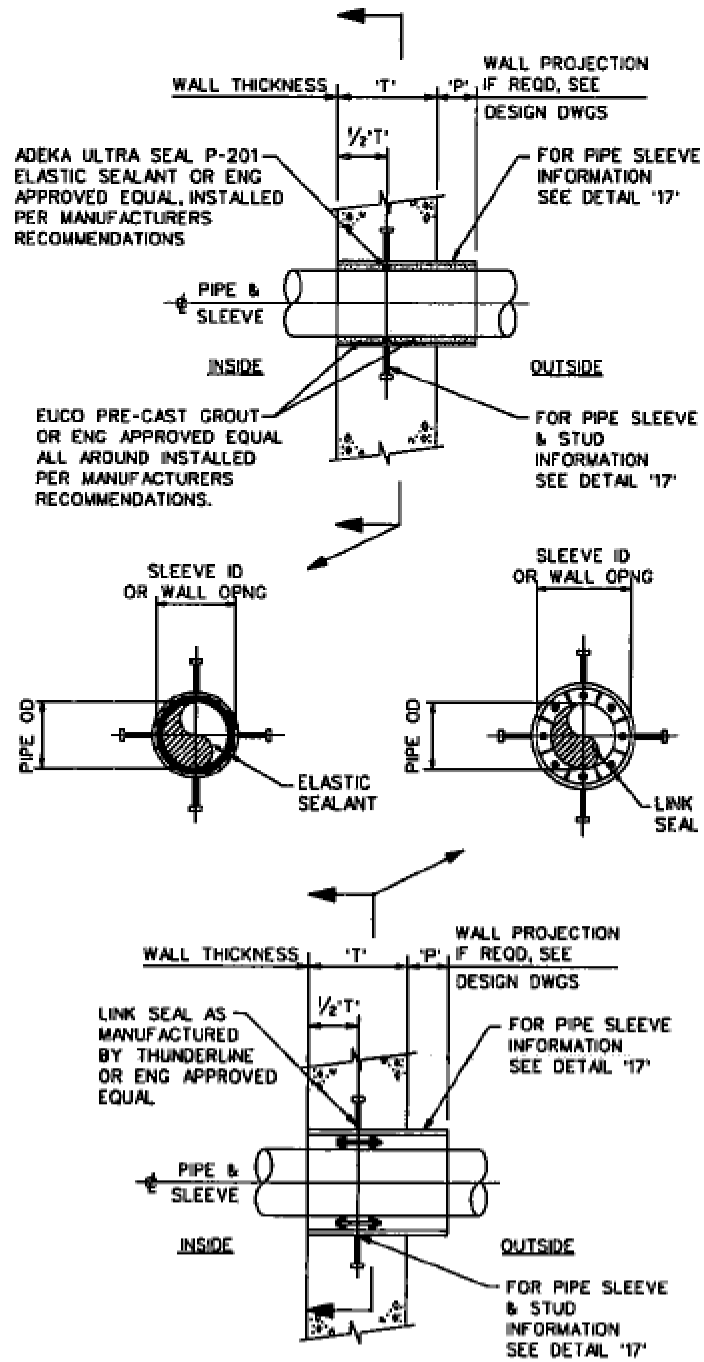
DETAIL 17



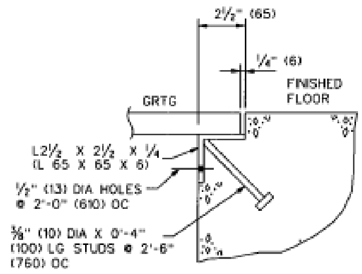
PENETRATION SLEEVE USING PVC PIPE FOR ELECTRICAL CABLE & CONDUIT

DETAIL 18

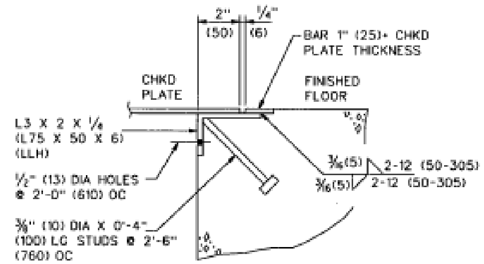
EMBEDDED SLEEVES



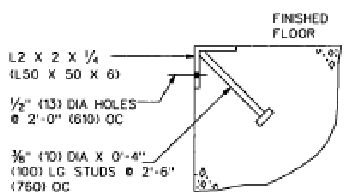
WALL SLEEVE & PENETRATION SEAL



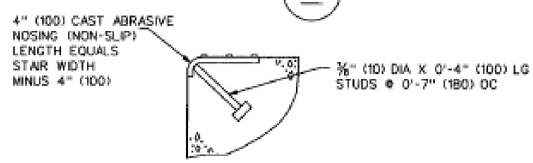
DETAIL 20



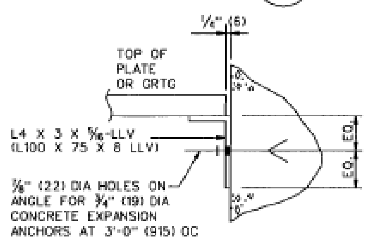
DETAIL 23



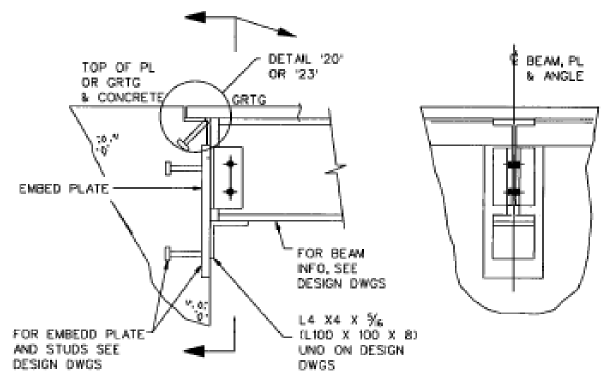
DETAIL 21



DETAIL 24



DETAIL 22



DETAIL 25

References

Fanella, D. A. (2000). *Concrete Floor Systems - Guide to Estimating and Economizing*, Portland Cement Association.

ISO TC184/SC4 (1997) Industrial automation system and integration - Product data representation and exchange - ISO 10303 - 42: Integrated Generic Resource: Geometric and Topological Representation, ISO Central Secretariat.