Analysis of efficient planning processes with BIM technologies in the planning of precast concrete elements

(Analyse effizienter Planungsabläufe mit BIM-Technologien in der Planung von Betonfertigteilen)

Master thesis

Approved by the Faculty of Civil engineering
of the Technische Universität Dresden

Written by

Konstantin Platonov

Supervisors:

Prof. Dr.-Ing. R. J. Scherer

Dr.-Ing. F. Jesse

Participating company:

Hetschke Bau GmbH

Dresden, March 2016

TECHNISCHE UNIVERSITÄT DRESDEN

Faculty of Civil Engineering

Appendix 4

Declaration of originality

I confirm that this assignment is my own work and that I have not sought or used inadmissible help of third parties to produce this work and that I have clearly referenced all sources used in the work. I have fully referenced and used inverted commas for all text directly or indirectly quoted from a source.

This work has not yet been submitted to another examination institution – neither in Germany nor outside Germany – neither in the same nor in a similar way and has not yet been published .

Dresden,

……………………………………………

(signature)

DRESDEN concept Exzellenz aus Wissenschaft und Kultur

Please note the Institute's information on writing a Master's Thesis.

Acknowledgements

I would like to sincerely thank my supervisor Prof. Dr.-Ing. Scherer for giving me the opportunity to do thesis work with the topic which is related to BIM and design automation.

Special thanks to Dr.-Ing. Jesse for his help in finding agreement between Chair of Construction Informatics in TU Dresden and participating companies Hentschke Bau GmbH and Precast Software Engineering GmbH, and also for encouragement, suggestion and help during my work.

I also would like to thank all of staff in the office Hentschke Bau for their support. Especially I would like to thank the design engineer Lars Wiesner for assistance in usage of software Nemetschek Allplan Precast and his guidance during the thesis work.

TECHNISCHE UNIVERSITÄT DRESDEN

Fakultät Bauingenieurwesen Institut für Bauinformatik

Anlage 1

Definition of the Master's Thesis

Name: Konstantin Platonov

Topic: Analysis of efficient planning processes with BIM technologies in the planning of precast concrete elements
(Analyse effizienter Planungsabläufe mit BIM-Technologien in der Planung von Betonfertigteilen)

Description:
For efficient processes in the work planning and work preparation, the market provides novel tools which are also support BIM technologies. This potentially allows replacement error-prone and time-consuming steps in the planning and plan creation through partial automation by using so-called intelligent design elements. The aim of this study is to investigate the performance of two software modules and to demonstrate and evaluate the particular performance on the basis of prototypical usage scenarios on a BIM model. These are the software modules "Smart Parts" and "iParts" of Nemetschek Precast AG.

Specific tasks:

1. Comparison of conceptual differences and technical performance parameters of the software modules
2. Selection and implementation of typical precast plans with the possibilities of the construction base module
3. Identification and validation of automatable design capabilities and prototypical implementation using the analysed software modules
4. Draft of a concept or design of methods and techniques for locating insufficient parameterized building elements
5. Qualitative assessment and recommendations for use of the software modules for practice

Supervisor TU Dresden: Prof. Dr.-Ing. R. J. Scherer
Supervisor industry: Dr.-Ing. Frank Jesse
Concerned company: Hentschke Bau GmbH

Handed out at: 24.11.2015

Submitted on: 24.03.2016

Prof. Dr.-Ing. R. J. Scherer

Die besonderen Hinweise für die Anfertigung der Diplomarbeit des Instituts sind zu beachten.

Abstract

Last years many construction companies consider use of BIM technologies in the planning processes. Nowadays the market provides novel BIM-based tools which are able to further increase the efficiency in the precast design. Time-consuming and error-prone steps are to be replaced using advanced parametric modelling tools. The aim of this study is to investigate the performance of two software modules and to demonstrate and evaluate the particular performance on the basis of prototypical usage scenarios on a BIM model. These are software modules "SmartParts" and "iParts" of Nemetschek Precast AG.

Keywords: BIM, Precast design, parametric modeling.

Table of Contents

List of abbreviations and symbols

Symbol/Abbreviation	Description
BIM	Building Information Modeling
BIM Model	Building Information Model
AEC	Architecture, Engineering and Construction
SP	SmartPart

1 INTRODUCTION

Aim of work

Aim of this study is to investigate the performance of two software modules and to demonstrate and evaluate the particular performance on the basis of prototypical usage scenarios on a BIM model. These are software modules "SmartParts" and "iParts" of Nemetschek Precast AG. The emphasis is on the rapid and effective implementation of parametric model and detailed plans of precast concrete elements.

.

Scope of work

First of all, the performance of modules SmartParts and iParts is to be analyzed. Secondly, the usage scenario for basic design modules is to be formulated. Thirdly, typical precast plans are to be selected and corresponding parametric 3D model is to be implemented using Allplan's basic design modules. Based on that, the automatable design capabilities of analyzed modules are to be identified and validated. In particular, partial automation of design of transport anchors is to be implemented. In addition, the techniques for locating of insufficiently parameterized elements are to be described. Furthermore, usage scenarios for analyzed modules are to be formulated and corresponding parametric 3D model is to be implemented. Based on that, the performace of analyzed modules is to be evaluated and recommendations for their use in practice are to be formulated.

2 STATE OF ART

2.1 What is BIM

The term Building Information Modeling is described as method of optimized design, implementation and management of buildings with the help of software. All relevant building data is digitally recorded, combined and networked. Building Information Modeling is used both in the construction industry for building design and construction as well as Facility Management.
The term Building Information Modeling is defined by Autodesk as three-dimensional, object-oriented, AEC-specific computer-aided design process.
"From scientific point of view Building Information Model – digital image of an existing or designed building while Building Information Modeling – process of its creation.
Also, Building Information Modeling can also mean the use of the digital model throughout the entire life cycle of building. In other words, one BIM model can be used beginning from stages of conceptual and detailed design and finishing with demolition of building or beginning the new cycle with renovation (Figure 1).

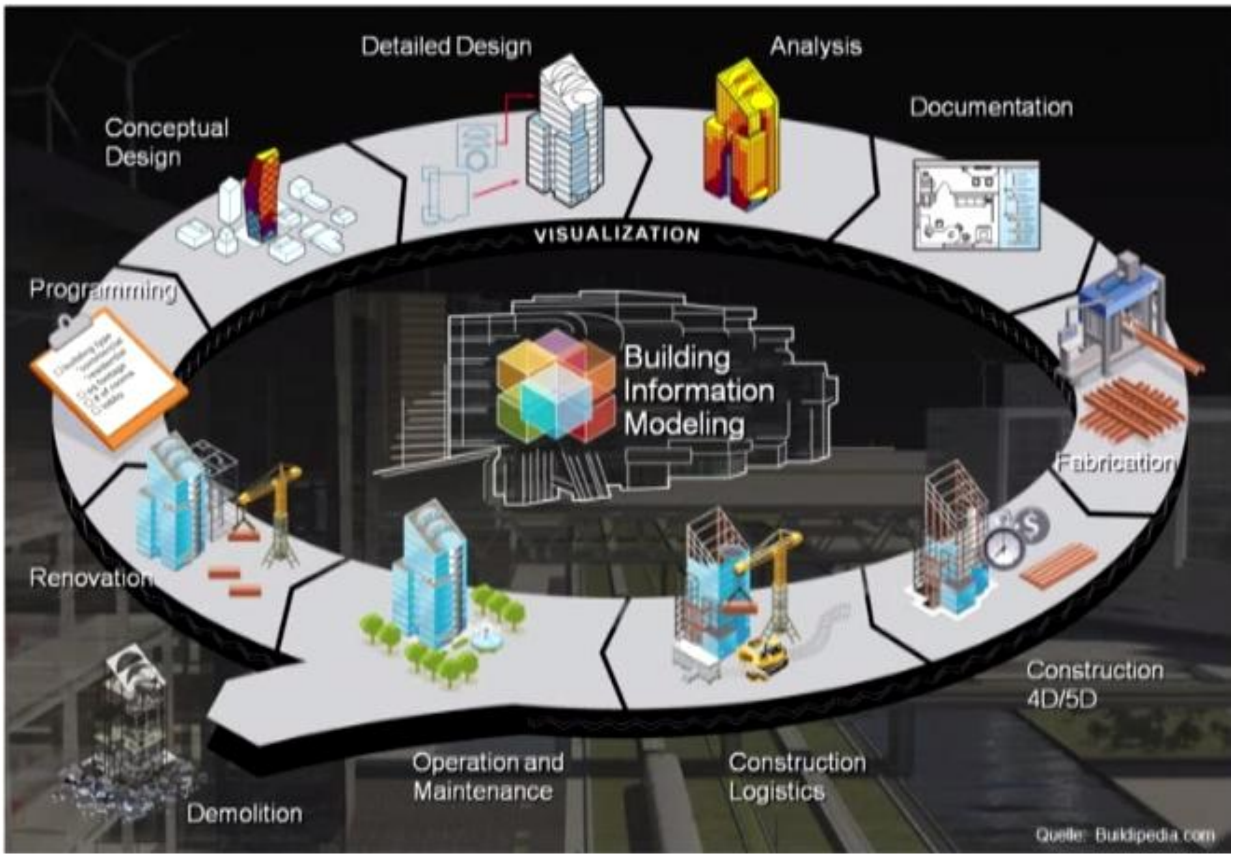


Figure 1. Life cycle of building

2.2 Motivation

“Comparing to convenient planning process the BIM has significant advantages:

- no repetitive data entry,
- better coordination between departments
- less mistakes,
- cost savings

Figure 2. Problems in coordination between the parties involved in the building of structures

All stages of life cycle are implemented by various departments. The data exchange between them is still frequently manual which is costly and error-prone. Hence, the expenses due to changes in a project increase along its stages.

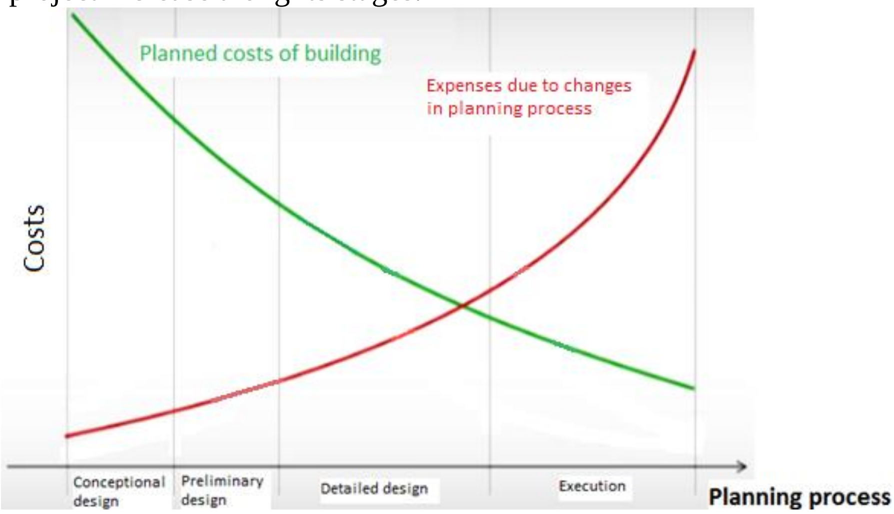


Figure 3. Costs of building at different project stages.

Application of BIM leads to less expense due to changes in project – for complex projects up to 30%. On the other hand, BIM requires more effort at stage of preliminary design while in convenient planning the effort reaches its peak closer to the end of design process. ”[1]

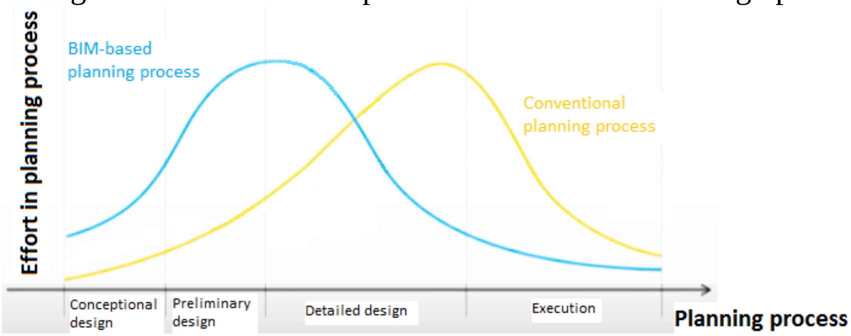


Figure 4. BIM: influence on efficiency in planning process

2.3 Design process

„Structural engineer

- Tool: static program,
- Defines static system and loads,
- Makes calculation / dimensioning / proof,
- Resulting with reinforcement and / or cross sections dimensioning.

Design engineer

- Tool: CAD program
- Generates reinforcement and general arrangement drawings,
- Based on requirements / work results of structural engineer. "[2]

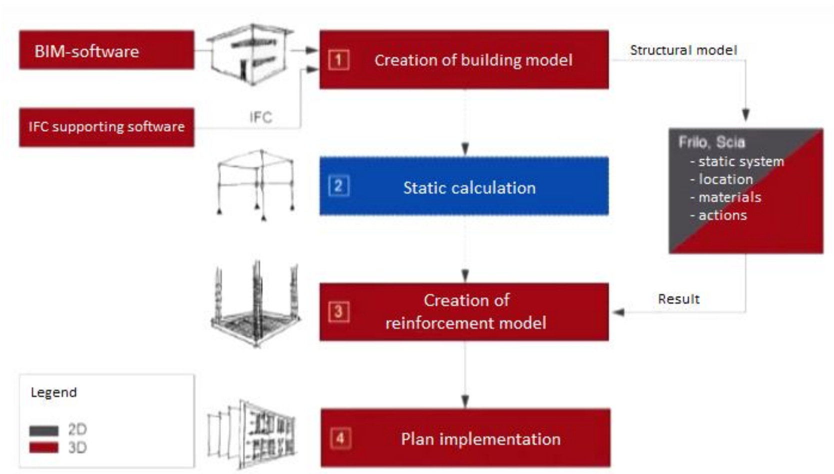


Figure 5. BIM optimal workflow using software Allplan

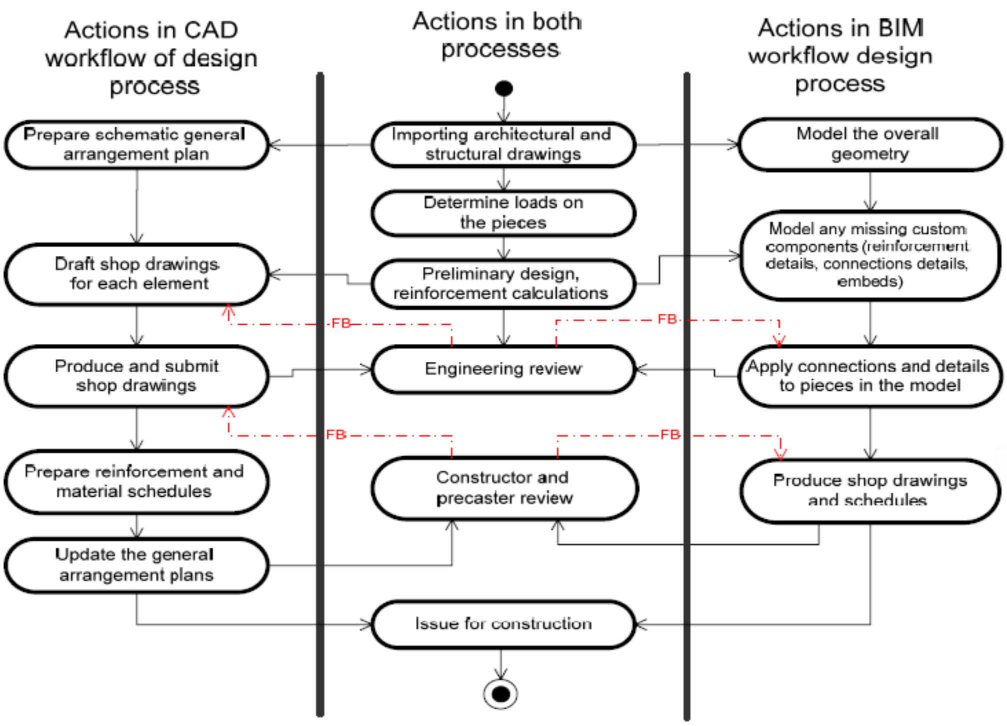


Figure 6. Actions in CAD / BIM design processes [3]

2.4 BIM-Model types

“Following the cycle, a model can be represented in different ways.

- Architectural model
    - contains all objects (non-bearing walls, windows, doors, shading)
- Engineering model
    - view of design engineer
    - contains bearing elements (non-bearing elements are hidden)
- Static model
    - view of structural engineer
    - 3-D objects are degenerated to lines (bar elements) and surfaces (formwork elements) "[2]

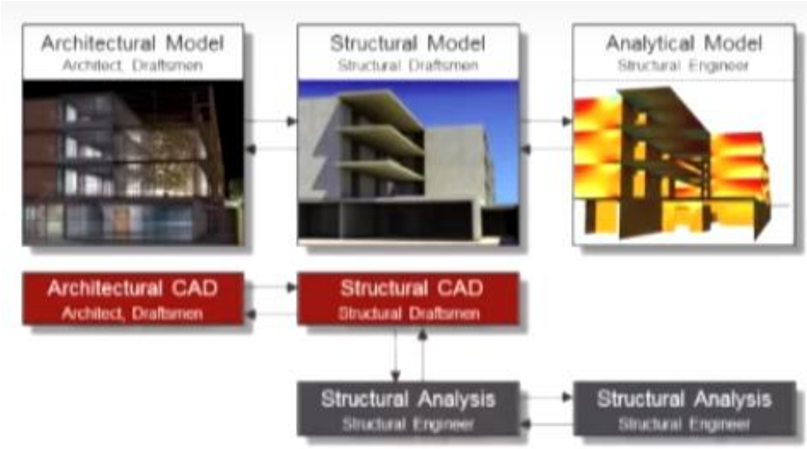


Figure 7. BIM-Model types

“BIM-Model includes not only 3-D geometry but also additional information – features – such as types of building elements, material properties and relations between building elements.”[1]

2.5 Parametric modeling

“Feature-based parametric CAD is currently the industry standard technology to create geometric models and assemblies, and is widely used across many engineering fields. In a parametric model, the geometry is mainly controlled by non-geometric features called parameters [4], which can be defined by dimensional, geometric, or algebraic constraints. "[5]

A distinction is made between a parametric building model and an intelligent building model. In parametric building model, all elements can be mutually brought into dependencies (walls, ceilings, dimensions, annotations, objects, cutting lines, etc.), while the intelligent building model is limited to individual objects.

Parametric modeling is obvious and intuitive. But for the first three decades of CAD this was not the case. Modification meant re-draw, or add a new cut or protrusion on top of old ones. Parametric modeling is powerful, but requires more skill in model creation. Skillfully created parametric models are easier to maintain and modify. Parametric modeling also lends itself to data re-use.

2.6 Concepts of top-down and bottom-up modeling

“Parametric solid modeling can be structured to support top-down or bottom-up modeling. In this context, the following definitions are used:
– Bottom-up modeling starts with explicit representation of distinct parts, with parametric relations defining the details and components of those parts, followed by successive association of the parts to form aggregated assemblies.
– Top-down modeling is the explicit definition of a total product, and then refinement of the product design by iteratively replacing objects that represent whole assemblies with successively finer grained parts until the level of detail required for production is achieved” [6]

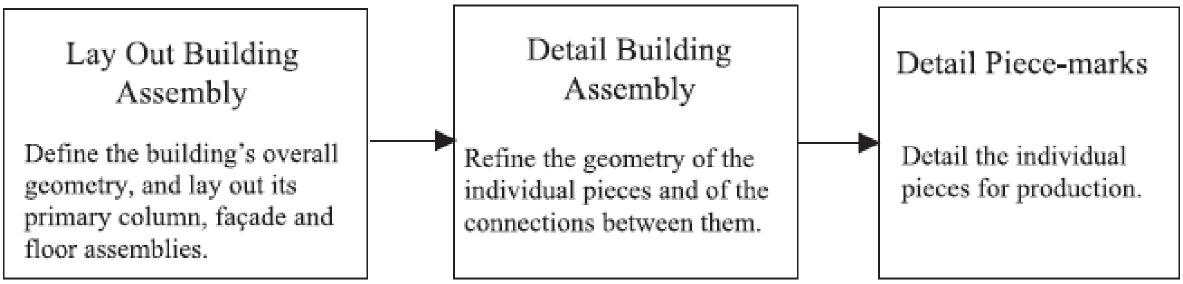


Figure 8 The process of precast design.

2.7 Software overview

BIM-based design is offered by all major CAD vendors. The strategies, reactions and terminology vary from manufacturer to manufacturer.
Nemetschek Group is one of the biggest vendors of software for architects, engineers and the construction industry. The CAD system Allplan is one the main Nemetschek subsidiaries. Allplan supports 2D design, 3D modeling to object-oriented building model with quantity takeoff and cost determination.

“Allplan Precast supports a variety of additional modules, i.e. Structural Precast Elements. These modules are integrated program packages for entering process planning data of precast concrete elements, such as slabs, wall panels, columns, girders etc. The concept facilitates a seamless flow of data - from the initial floor plan to automatic slab, roof and wall element design to plan generation to quantity takeoff operations to invoicing.

2.8 Design modules

Software Allplan Precast has a modular structure. In other words, it features individual program modules, each of which contains the necessary tools for a specific discipline. The modules themselves are arranged in families, such as

- Basic family,
- Bonus Tools family, including
  - 3D Modeling:
  - SmartParts
- Architecture, including
  - Basic: Walls, Openings, Components
- Engineering, including
  - Bar Reinforcement

    - Concrete Construction
- Views and details, including
    - Associative Views
- Precast Elements, including
    - Precast Slab
    - Precast Wall
    - Structural Precast Elements
    - Catalogs and Configurations” [7].
- Others

Tools classification

Mentioned earlier tools are to be divided into three main groups

- Basic tools,
- SmartParts,
- iParts.

Tool Concrete Construction uses the same principle as SmartParts and, hence, also considered as part of module.

Table 1. Tools classification

Tools group 1 “Basic tools” or “Basics”	Tools group 2 Module “SmartParts”	Tools group 3 Module “iParts”
- Associative Views - Bar Reinforcement - Basic: Walls, Openings, Components - 3D Modeling:	- SmartParts - Concrete Construction	- Structural Precast Elements

Tools

Tools group 1: Basic tools

- “Associative Views
    - Module contains tools for creating self-updating views and sections for 3D general arrangement drawings and 3D reinforcement drawings drawn using the 3D model. The reinforcement is created directly in the 3D floor plan and the corresponding associative views or in the 2D general arrangement drawing.
- Bar Reinforcement
    - Module contains tools for placing bar reinforcement
        - FF Components tool uses component groups to create bending shapes and their placements in a single step.
- Basic: Walls, Openings, Components
    - Module contains tools for creating three-dimensional building drawings
- 3D Modeling:
    - Module contains tools for creating three-dimensional drawings

- Concrete construction - 3D object tool provides a convenient way of creating predefined engineering components (e.g. pre-cast columns). The components can be selected in a catalog and customized using dialog boxes and shortcut menus.

Tools group 2: Module SmartParts

- SmartParts
  - Module contains tools for creating parametric 3D components which can be modified and updated later.
    - Element plan – tool
- Concrete Construction
  - Module contains tools for creating parameterized reinforcement components. Element plans can be derived from these components.

Tools group 3: Module iParts

- Structural Precast Elements
  - Module contains tools for creating parameterized precast columns, foundations, beams, girders, stairs, walls, slabs and other elements."[7]

3 COMPARISON OF SOFTWARE MODULES

3.1 Concept of SmartParts

„Module SmartParts represents the concepts of convenient bottom-up (part-by-part) parametric modeling. It is designed as addition to 3D Modeling module. Script language enables users to create parametric components and content which makes it possible to develop and exchange their own parametric objects. A SmartPart is a parametric CAD object. Parametric information is controlled by a script, which is attached directly to the object. SmartParts can be edited using handles (graphic modification) or a dialog box (alphanumeric modification) or SmartPart editor tool.

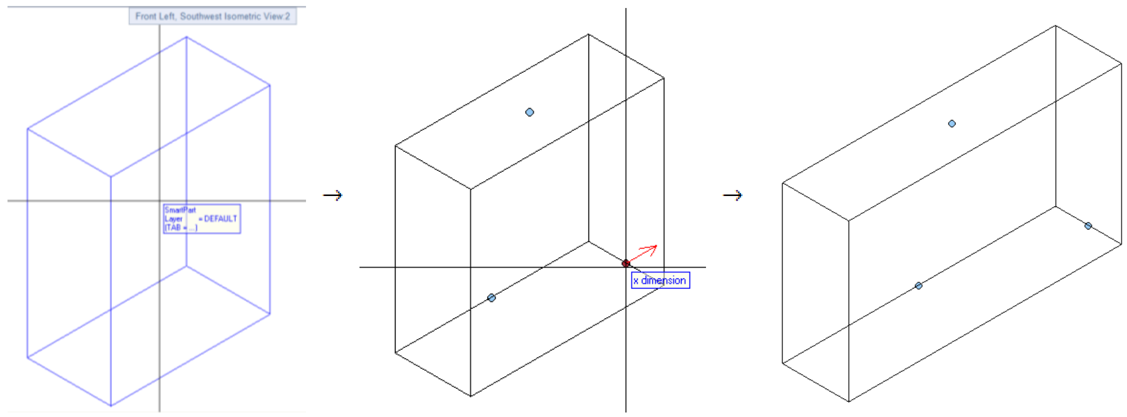


Figure 9. SmartPart graphic modification

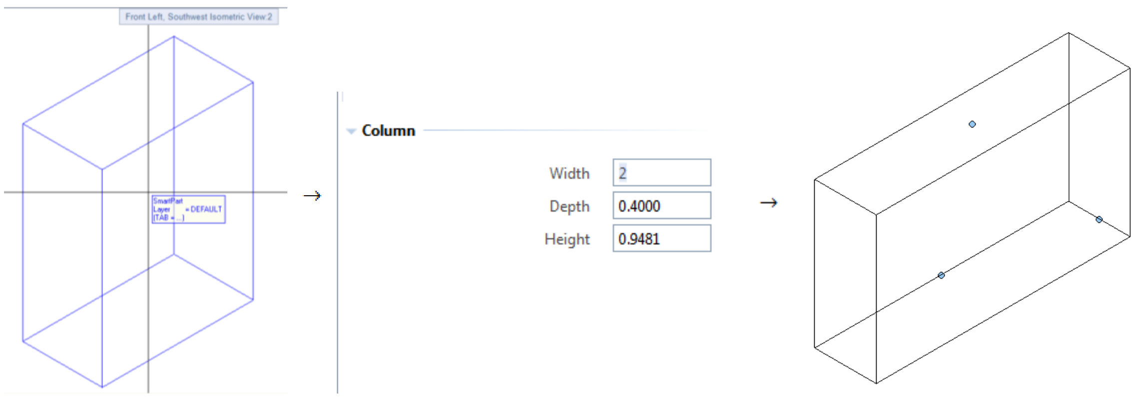


Figure 10. SmartPart alphanumeric modification

Parameter list and scripts are defined using SmartPart editor tool. Parameter list stores independent parameters. SmartParts scripts can be divided into groups depending on command types. Groups are Parameter script, Dialog script, 2D script and 3D script. Parameter script allows to operate with parameters, i.e. to set limitations on parameter input, as well as to create the new ones which depend on existing parameters. Dialog script is used to set dialog box user options. 2D and 3D scripts are used to create parametric geometry and to set graphic modification options for user.

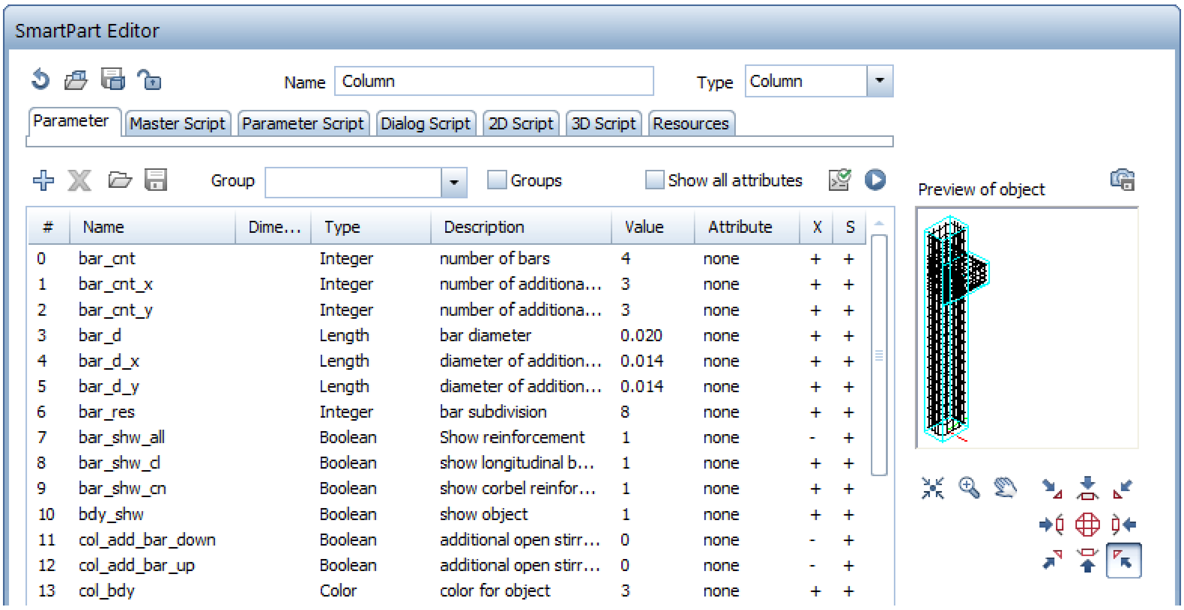


Figure 11. SmartPart editor

3.1.1 SmartParts in precast design

Script commands allow creating reinforcement. Using dependencies between geometry of reinforcement and element geometry it is possible to create reinforced precast elements.

Using Element plan tool the reinforcement plans can be automatically generated in accordance with predefined patterns of plan – layouts. Module SmartParts doesn’t offer user modification options for layouts.

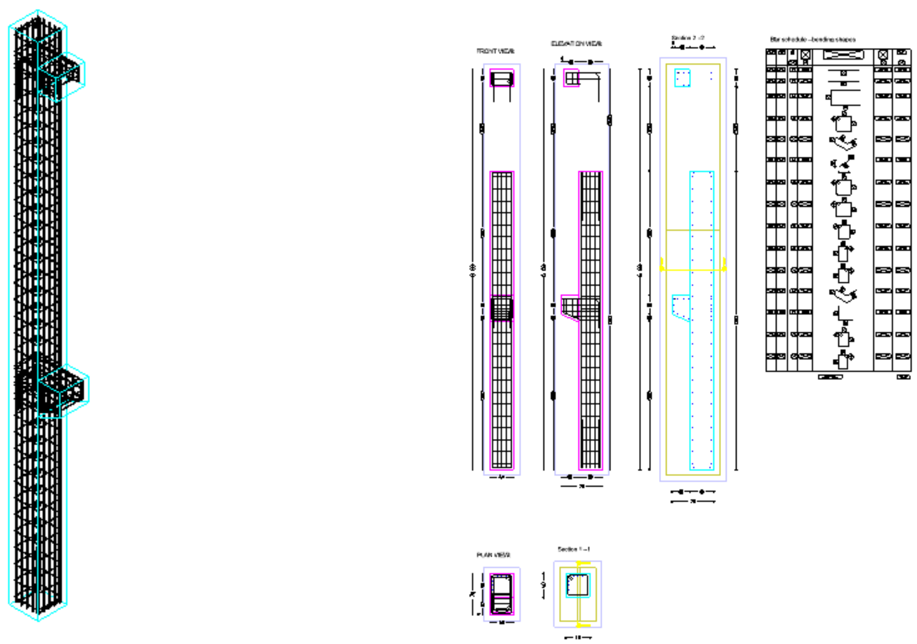


Figure 12. SmartPart Precast column with reinforcement plan

3.1.2 Element format transformation

Any SmartPart element can be transformed into convenient Allplan 3-D object including its reinforcement. Format transformations of SmartParts-reinforcement in 3D format is possible.

On the other hand, 3-D object can be included into a SmartPart as resources using 3-D script option. External objects are not based on SmartPart script. Included into the SmartPart objects can't be modified, either graphically or alphanumerically. The only way to edit 3-D object inside of SmartPart is to replace it with another – correctly defined 3-D object.

3.1.3 Data reuse options

Module SmartParts offers to user the opportunities to create own databases or to use databases of other users. File format is .smt. Parameter set also can be saved (file format .smv).

3.1.4 Workflow

Process of modelling of a SmartPart is described using flowchart (Figures 13, 14).

3.1.5 Supported parameter types

- Boolean, integer, decimal
- String
- Length, angle
- Color, pen, line, layer, surface, pattern, hatching, area style, drawing type

3.1.6 Script commands

Modeling of SmartPart elements is based on script commands which use parameters. Number of parameters is not limited. Nevertheless, the number of commands is limited:

- Non geometrical commands
    - Parameter commands (define dependent parameters in parameter script)
    - Dialog commands, Reinforcement commands
      (define user options for the alphanumeric modification of parameters set)
    - Palette commands
- Handles (define user options for the graphic modification of an element)
- 2D shapes
- 3D shapes
- Transformations (define the location for the modeled by script elements)
- Attributes
- Reinforcement functions (define reinforcement parameters: bar diameter, concrete strength, steel grade, bar length, hook length, hook angle)
- Expressions and functions
- Control Commands
- Environment
- Global Variables

3.1.7 Example of SmartParts parameter set

Available by default SmartPart element Sleeve foundation is selected as example demonstrating complete SmartParts parameter set.

- General
    - Standard
        - Concrete grade, Steel grade
    - Reinforcement
        - Layer options, Visualization
    - Foundation
        - Foundation type (rough, smooth), concrete cover for foundation and for sleeve
    - Object settings
        - Visualization, Layer options, Color settings etc.
- Geometry
    - Foundation geometry in its entirety, Illustration
    - Sleeve geometry in its entirety, Geometry of sleeve wall, Illustration
- Longitudinal reinforcement
    - in X direction, in Y direction, Illustration
        - Bar diameter, offset, hook length
- X-sleeve wall
    - in X direction – Vertical stirrup, in Y direction – Horizontal stirrup, Illustration
        - Bar diameter, bar spacing in center and from edge, hook length / overlap length, bending pin diameter
- Y-sleeve wall
    - in X direction – Vertical stirrup, in Y direction – Horizontal stirrup, Illustration
        - Bar diameter, number of bars in corner, bar spacing in and from center and from edge, hook length / overlap length, bending pin diameter

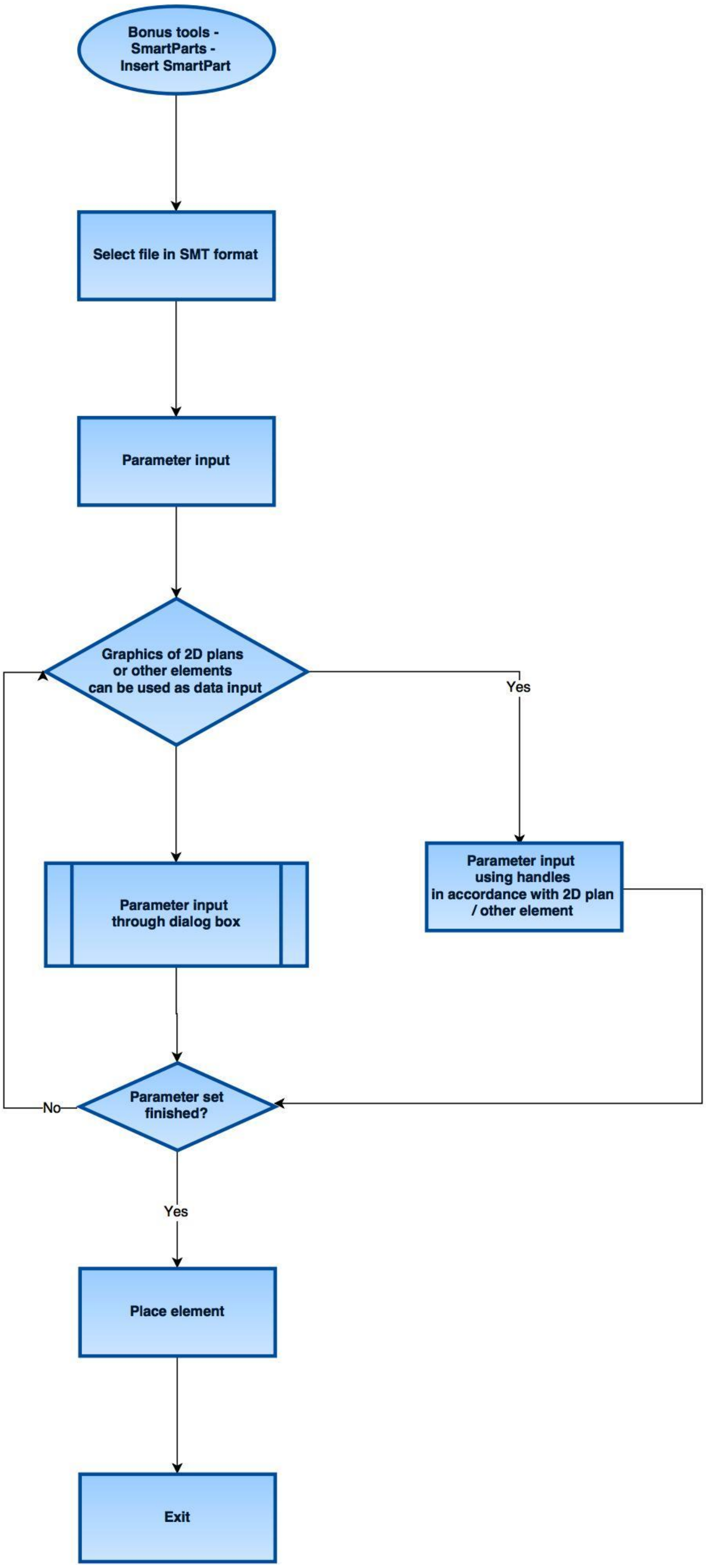


Figure 13. SmartParts: parameter set

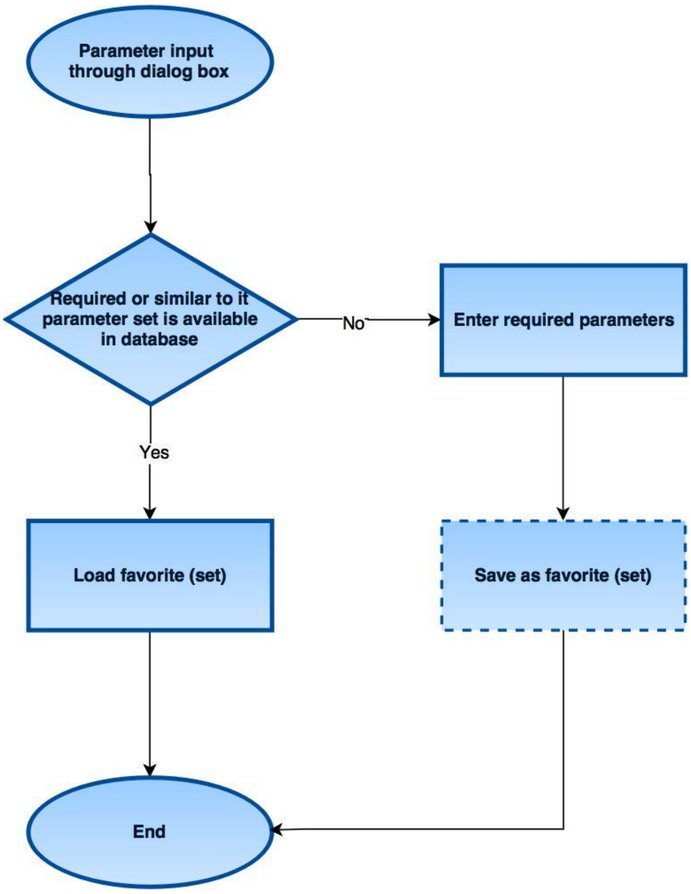


Figure 14. Parameter input through dialog box

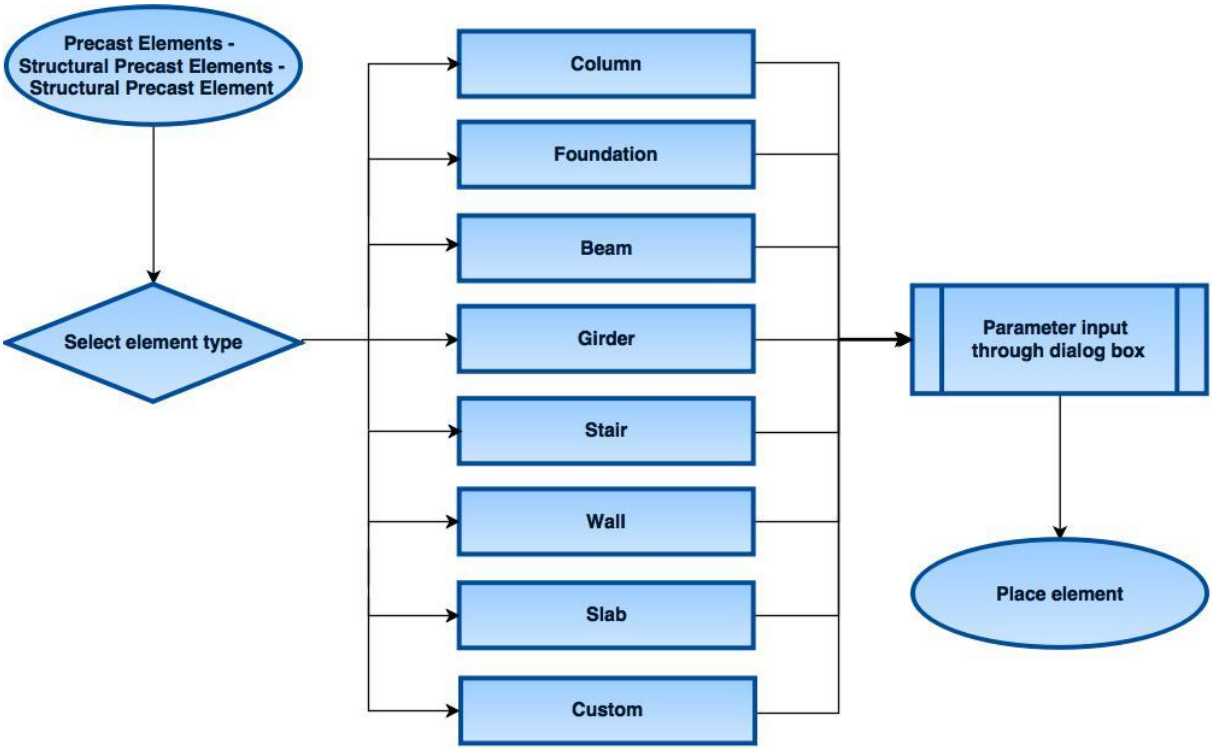


Figure 15. iParts: parameter set

3.2 Concept of iParts

Aim of modules family Precast elements is to automatize design of precast elements and to simplify the data transfer to further stages such as production, logistics and invoicing. Depending on element type different modules are to be chosen – Precast slab, Precast wall or Structural precast elements. Module Structural precast elements is also known as Structural precast units or iParts. Module iParts is oriented on design of foundations, beams, girder and stairs as well as slab, wall or other 3-D objects. Module iParts represents the concepts of top-down parametric modeling (stepwise detailing of model).

Elements of already existing architectural or 3D object model can be transformed into iParts. Also modeling can be performed by entering the geometry parameters. Following the principles of parametric modeling a model updates automatically to reflect changes made to the design.

Using the Join Structural Precast Elements tool the connections of types column-beam, column-girder and column-foundation can be performed based on parameter input.

iParts can be edited only using dialog box (alphanumeric modification).

Module iParts offers reinforcement options only for modeling of stairs. Otherwise it can be done either with SmartParts or with other tools.

Using Element plan tool the reinforcement plans can be automatically generated in accordance with predefined plan patterns – layouts. Module iParts offers a variety of modification options for layouts.

3.2.1 Element format transformation

Any iPart element can be transformed into convenient Allplan 3-D object as well as any 3-D object can be transformed into iPart. Options for further geometry modifications of such based-on-3D iParts are not provided. In this case iPart should be transformed back into 3-D object, edited and again transformed into iPart.

Any architectural object can be transformed into iPart without the removal of original object.

3.2.2 Data reuse options

Module iParts offers to user the opportunities to create own databases (catalogs) or to use databases of other users. Following data can be stored, imported/exported and further used:

- Parameter set
- Details of elements (consoles)
- Connections
- Element plan layouts

3.2.3 Workflow

Process of modelling of a iPart is described using flowchart (Figures 14, 15).

3.2.4 Supported parameter types

- General (Input option, Cross-section/ Shape)
- Dimensions (element-type-based)
- Attributes tab (Select factory, Label, Classification - Concrete grade, Exposure class ,Fire resistance classification, Additional attributes)
- Dimensions tab (Local coordinate system, Alignment, Production dimensions, Loading dimensions)
- Invoicing tab"[7]

3.2.5 Example of iParts parameter set

iPart element type Foundation is selected as example demonstrating complete iParts parameter set.

- Geometry
    - General
        - Input option (parameters, 3-D object)
        - Shape (Rectangle, Hexagon)
    - Dimensions
        - Widths, thickness, socket (without, attached, cut out), piles settings
    - Socket position
        - If not centered – reference point, eccentricities
    - Socket geometry
        - Widths, enlarge at top, wall thicknesses, wall height, depth, profiling
- Attributes
    - Factory
    - Label
        - Label, mark number, additional text, component name, text anchor point
    - Classification
        - Concrete grade, Exposure class, Fire resistance classification
    - Element states and actions
- Dimensions
    - Alignment
    - Loading dimension
    - Production dimensions
- Invoicing
    - Invoice item
    - Invoice for customer
    - Invoice for factory

3.3 Conceptual differences

Advantages and disadvantages in comparison between modules SmartParts and iParts are marked green and red.

Table 2. Conceptual differences

Comparison criteria	SmartParts	iParts
Initial purpose	Addition to 3D Modeling module	Precast design automation
Parameters input options	-Dialog box -Graphic -Script	-Dialog box -Based on other objects
Transformation options	-SmartPart → 3D	-3D ↔ iPart
Variety of available element types	-Default catalog has a few elements -Catalog can be refilled by user	-8 types of elements available
Data reuse options	-Same format for data storage, -Online catalogs available	-Format depends on data type
Options for reinforcement design	-Provided -Possible by basic tools	-Only with other tools - basic tools, SmartParts
Automatic plan generation	-Provided, with poor parameter set	-Provided

3.4 Comparison by performance parameters

Table 3. Comparison by performance parameters

Comparison criteria	SmartParts	iParts
Parameters for modeling of element	-Defined by modeler / programmer -Limited only by available types (length, string, integer)	-Defined by software -Limited by list of available parameters
Parameters for connection design	-Available, considered as part of single element modeling	-Available, parametric relations between elements supported -Limited by available connection types
Parameters for design of reinforcement, 3D build-in parts	-Available	-Not available
Element plan parameters	Limited by available plan layouts	Numerous layout settings are to be set by user

Conclussion

In terms of detailed design of elements module SmartParts has more opportunities than iParts and other tools. All options provided by iParts and other tools can be programmed with SmartParts. Nevertheless, the default catalog of SmartParts is poor.

On the other hand, module iParts supports automatic plan generation and parametric relations between modeled elements.

4 SELECTION AND IMPLEMENTATION OF TYPICAL PRECAST PLANS USING BASIC DESIGN MODULES

4.1 Selection of typical precast plans

Typical precast plans are chosen based on requirements:

- Precast plan is used to demonstrate the tools and features of modules SmartParts and iParts
- Precast plan should reflect the particular actual problems in precast design based on information from the cooperating concrete factory

Pattern drawings for precast concrete products provided by organization German Precast Concrete Construction (orig. Deutscher Betonfertigteilbau e.V.) are chosen as the basis for model. The pattern drawings are based on Eurocode 2 [8]. Drawings are provided in Appendix 1.

Drawings contain elements plans of precast column, slab and girder as well as arrangement plan. Arrangement plan differs from the element plans.

Table 4. Provided parameters for modeling of elements

Provided parameters	**Value acc. arrangement plan**	**Value acc. element plans**
Model		
Axes dimensions [m]	6x6	-
Height of building [m]	10,5	-
Element types	column, girder, beam	Column, girder, slab
Elements		
Column S001 bxh [m]	0.5x0.5	0.4x0.4
Column S002 bxh [m]	0.5x0.5	-
Column S003 bxh [m]	0.5x0.5	-
Column S004 bxh [m]	0.5x0.5	-
Column S005 bxh [m]	0.5x0.5	-
Column S006 bxh [m]	0.6x0.6	-
Beam B001-01 bxhxL [m]	0.19x0.50x5.85	0.6x1.7
Beam B002-01 bxhxL [m]	0.20x0.51x5.85	-
Beam B002-02 bxhxL [m]	0.20x0.51x5.83	-
Beam B002-03 bxhxL [m]	0.20x0.51x5.75	-
Beam B003 bxhxL [m]	0.5x0.825x5.83	-
Beam B004 bxhxL [m]	0.5x0.825x5.905	-
Girder UZ001 bxhxL [m]	0.3x1.5x17.775	-
Girder UZ002 bxhxL [m]	0.3x1.5x17.36	-

Pattern drawings for precast concrete products. Musterzeichnungen für Betonfertigteile

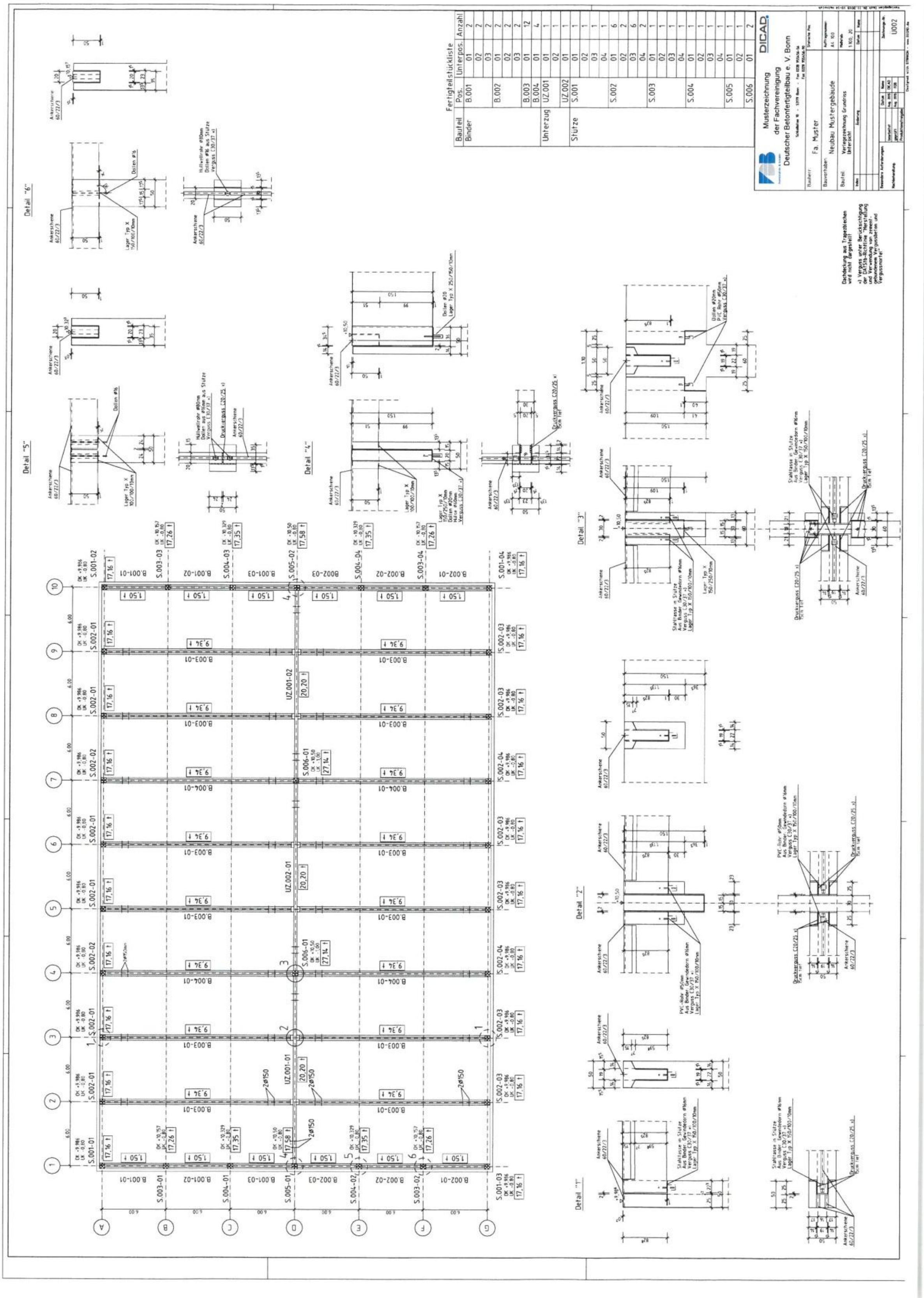


Figure 16. Arrangement plan

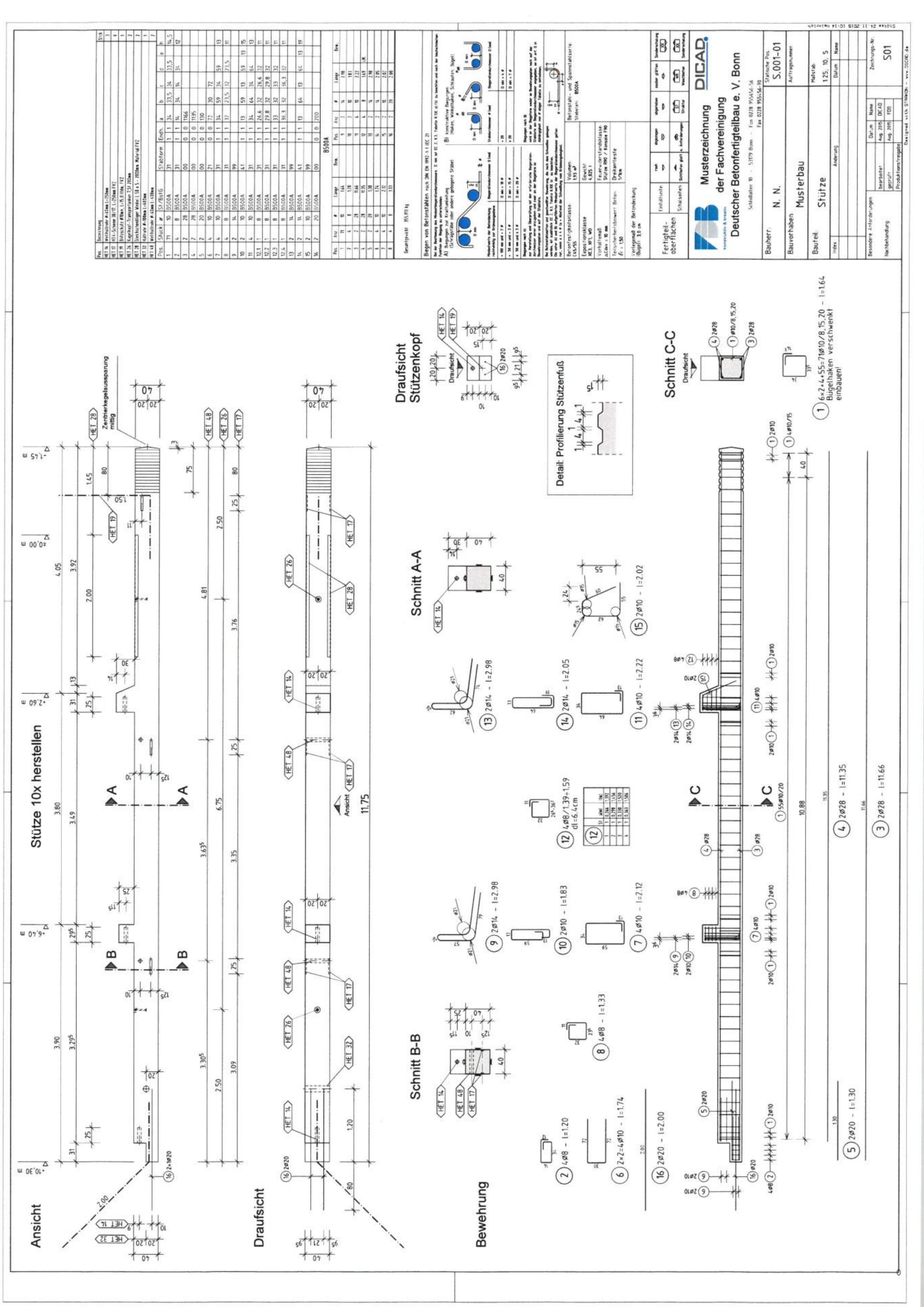


Figure 17. Element plan: precast column

4.2 Parametric assumptions for precast plans

Since all parameter sets for building elements can’t be clearly defined from precast plans some of them are assumed. Final parameter assumptions are based on both general and element plans.

Table 5. Assumed parameters for modeling of elements

Assumed parameters	**Value**
Model	
Axes dimensions [m]	6x6
Height of building [m]	10,5
Element types	column, girder, beam
Elements	
Column S001 bxh [m]	0.4x0.4
Column S002 bxh [m]	0.4x0.4
Column S003 bxh [m]	0.4x0.4
Column S004 bxh [m]	0.4x0.4
Column S005 bxh [m]	0.4x0.4
Column S006 bxh [m]	0.6x0.6
Beam B001-01 bxhxL [m]	0.16x0.5x5.85
Beam B002-01 bxhxL [m]	0.16x0.5x5.85
Beam B002-02 bxhxL [m]	0.16x0.5x5.85
Beam B002-03 bxhxL [m]	0.16x0.5x5.75
Beam B003 bxhxL [m]	0.4x0.825x5.83
Beam B004 bxhxL [m]	0.4x0.825x5.905
Girder UZ001 bxhxL [m]	0.3x1.5x17.775
Girder UZ002 bxhxL [m]	0.3x1.5x17.36

4.3 Representation of workflow in usage scenarios

Usage scenarios are represented further in the following manner

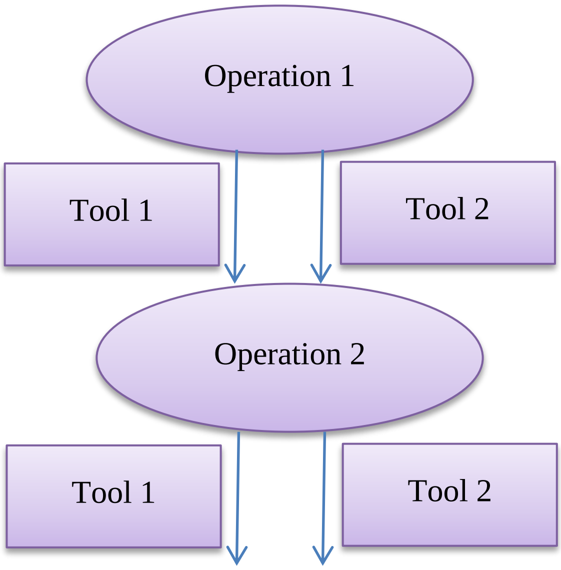


Figure 18. Representation form for usage scenarios

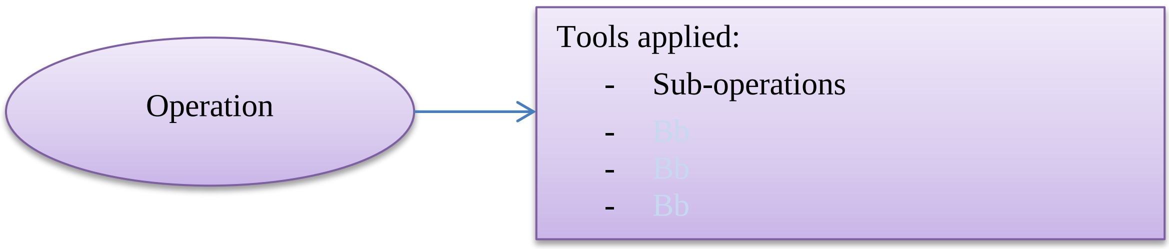


Figure 18. Operation description: Representation form

The usage scenario which is based only on application of basic tools is defined as scenario 1.

Many elements, especially in big projects, have similar geometry. The geometry data which is the same for many objects is defined as basic geometry and the elements with the same basic geometry are to be defined as basic elements. On the other hand, the geometry data which is different from element to element is to be defined as advanced geometry.

4.4 Usage scenario 1: basic design modules

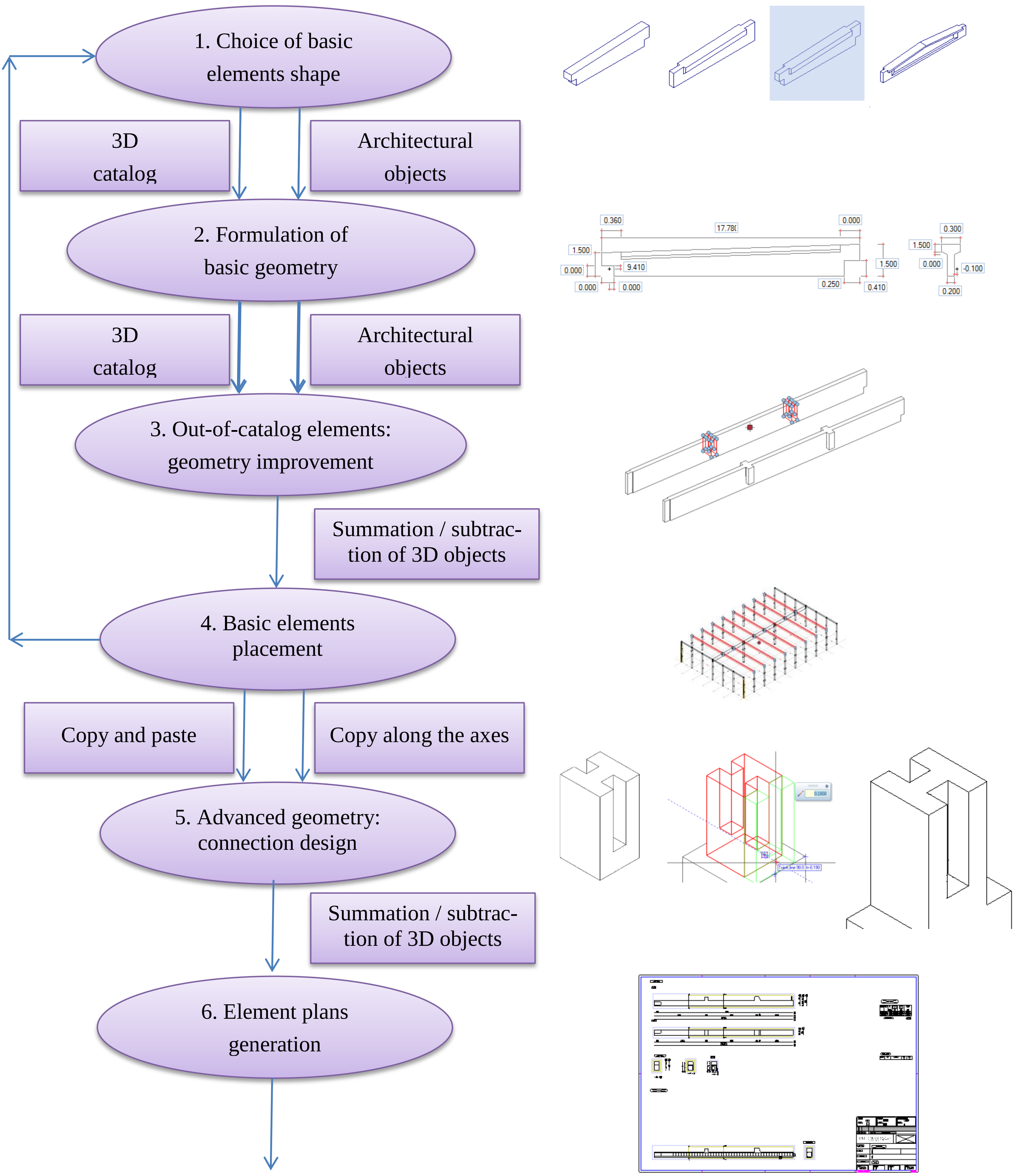


Figure 19. Scenario 1. Part 1: implementation of model with basic tools

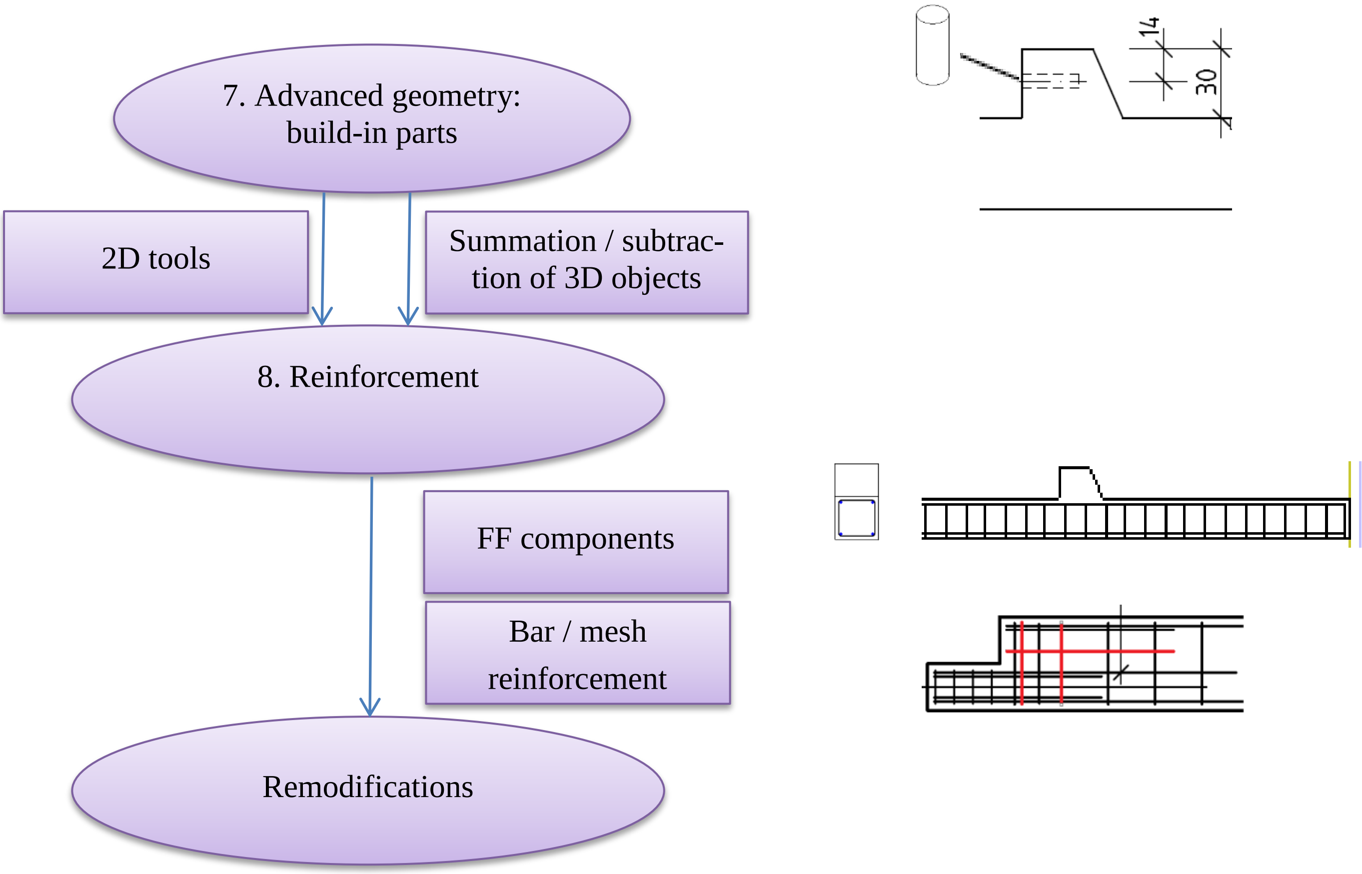


Figure 20. Scenario 1. Part 2: implementation of model with basic tools

4.5 Automatable operations in usage scenario 1.

Usage scenario 1 contains operations which can be partially or completely automatized using modules SmartParts and iParts. Those operations are further described in more detail.

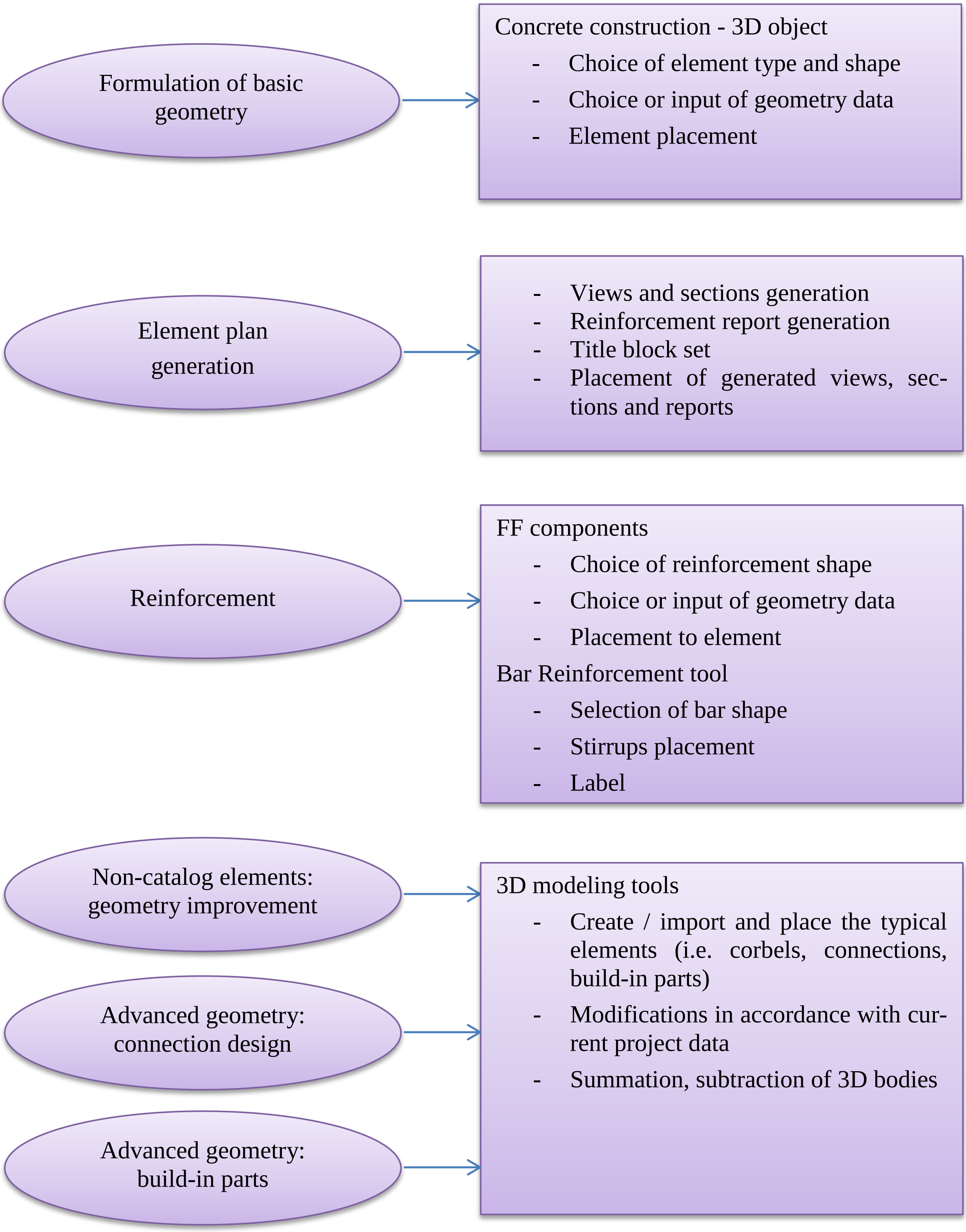


Figure 21. Scenario 1: automatable operations

5 IDENTIFICATION AND VALIDATION OF AUTOMATABLE DESIGN CAPABILITIES

5.1 Capabilities of iParts

5.1.1 Generation of element plans

One database with more advanced settings (layout pattern) allows avoiding of usage of 5 databases and related operations

Table 6. Element plan: work flow using various modules

Basic tools	**SmartParts**	**iParts**
View generation	Engineering – Concrete construction – Element plan tool	Precast elements – Structural precast element – Element plan tool
- Choice of elements	- Choice of elements	- Choice of elements
- Direction choice	- Choice of layout pattern	- Choice of layout pattern, scale options
- Settings		
- Placement		
- Label settings		
- Label placement		
Section generation		
- Choice of elements		
- Direction choice		
- Section cut placement		
- Settings		
- Placement		
- Label settings		
- Label placement		
Reinforcement reports tool		
- Table settings		
- Reinforcement choice		
Title block setting		
Placement of views, sections and reports		
- Choice		
- Settings		
- Placement		

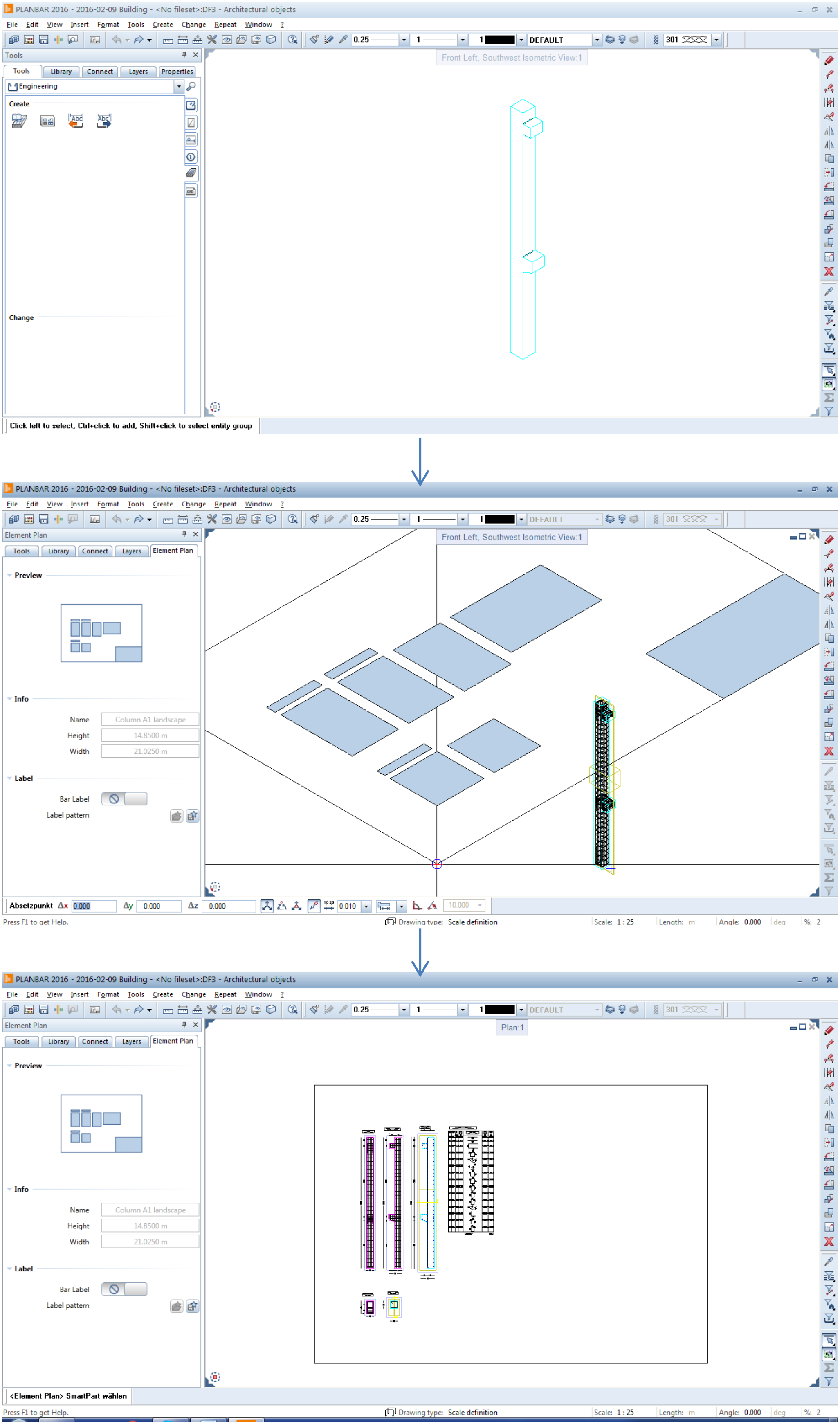


Figure 22. Generation of element plan using SmartParts

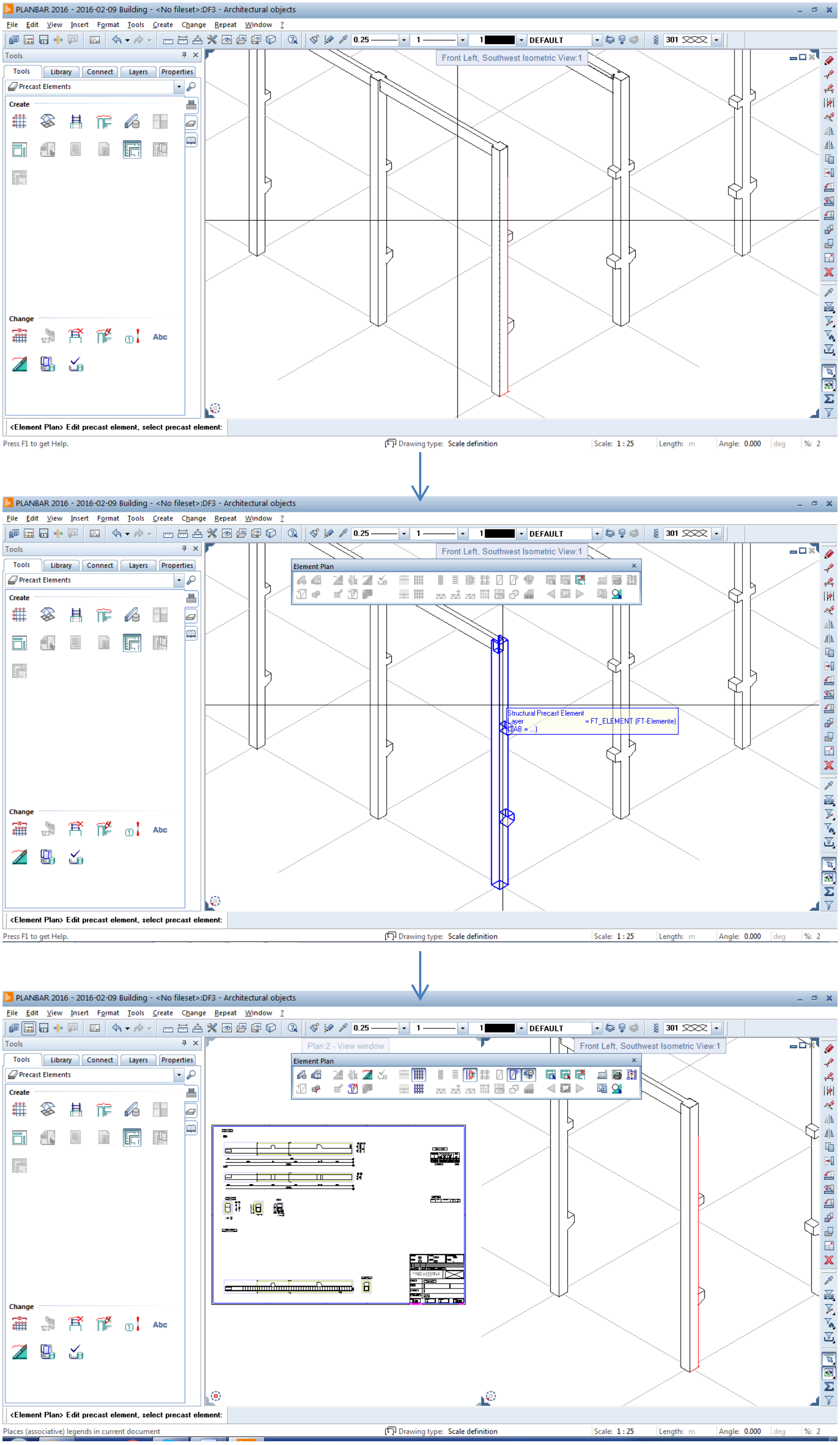


Figure 23. Generation of element plan using IParts

5.1.2 Connection design

Connection design with iParts allows avoiding steps in modeling of element geometry (either for column or for beam which are to be connected) and cut out of the body part using 3D Boolean operations. It's not necessary to be aware of exact element geometry near the connections. Also iParts connections are easier to remodify.

Table 7: Connection design: work flow using various modules

Basic tools	**iParts**
3D modeling tools	Join structural precast elements
- Create / import the elements	- Selection of connection type
- Placement of elements	- Choice or input of geometry data
- Adjustment of connection to element geometry	- Selection of connecting elements
- Summation of 3D objects	

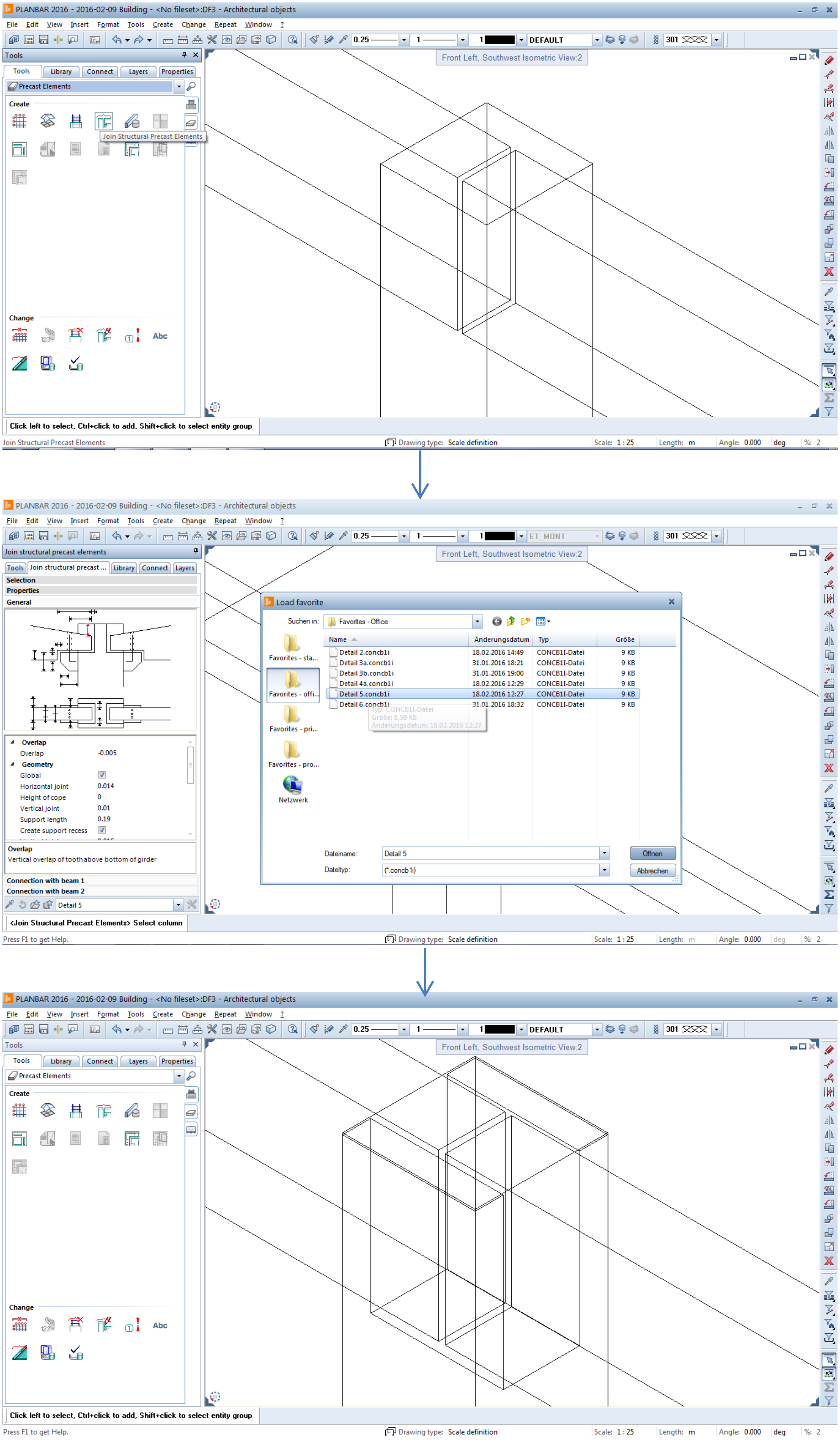


Figure 24. Connection design with iParts

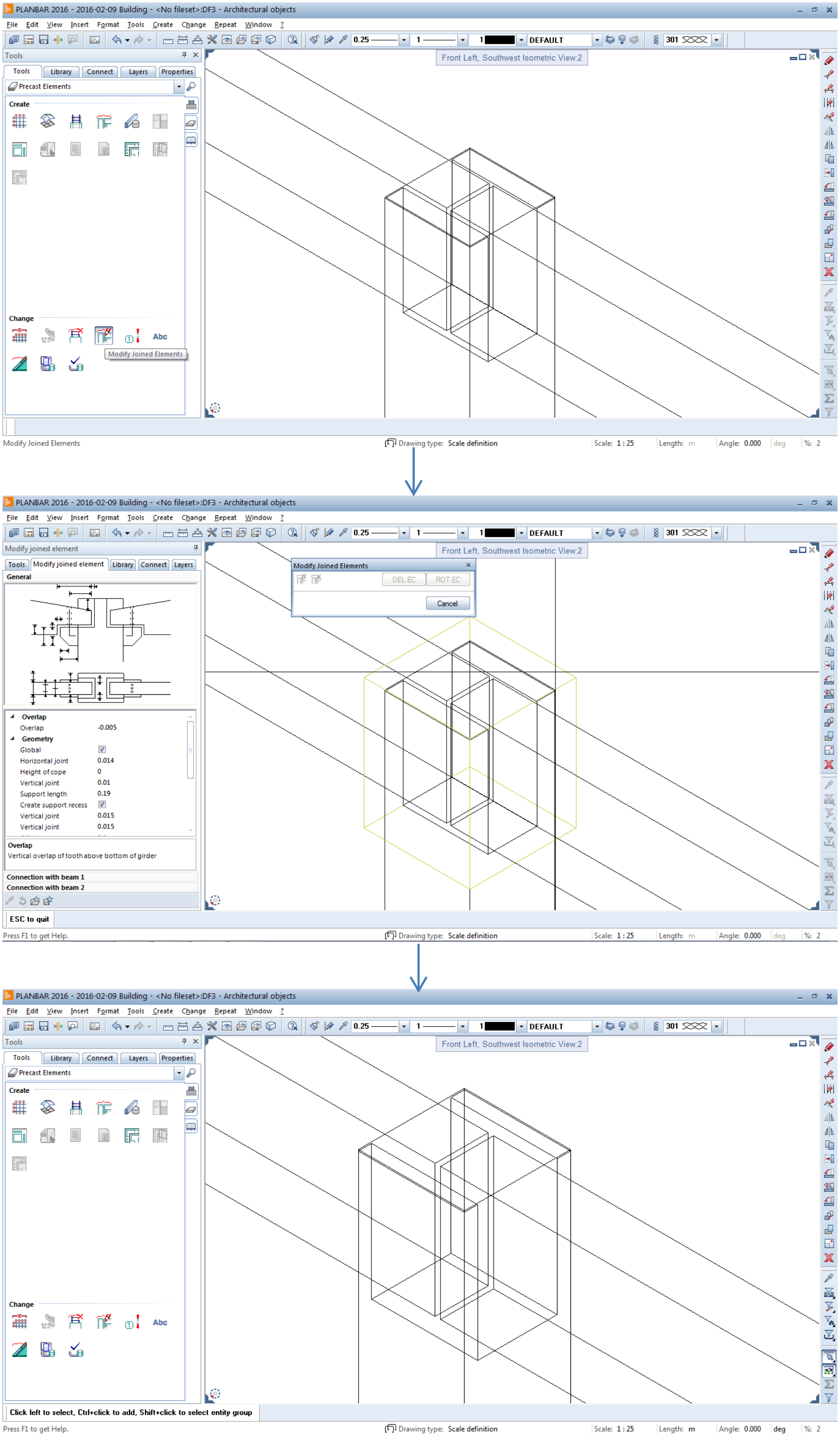


Figure 25. Modifications using parameter set

5.1.3 Locating of insufficiently parameterized elements

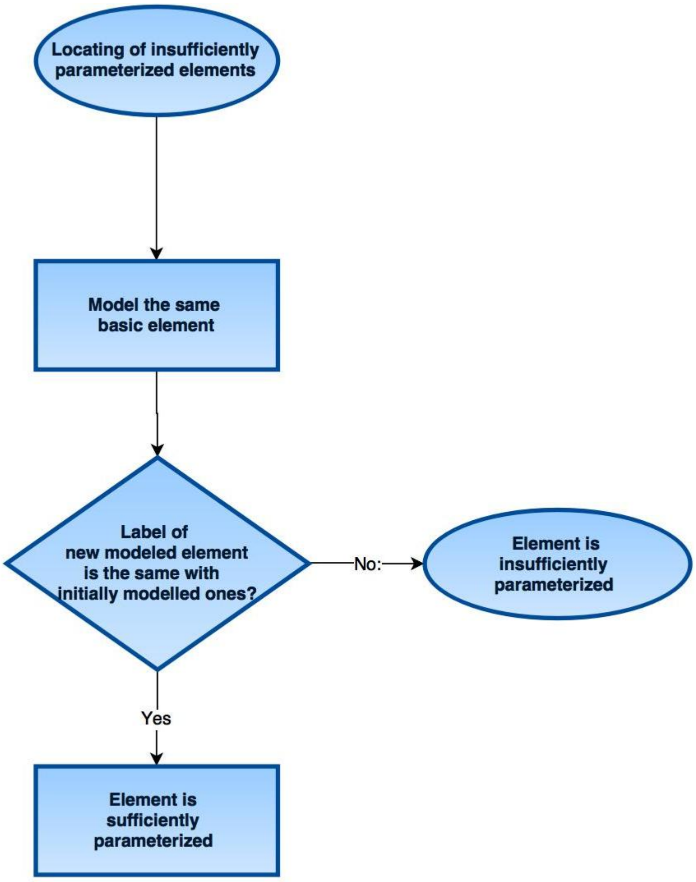
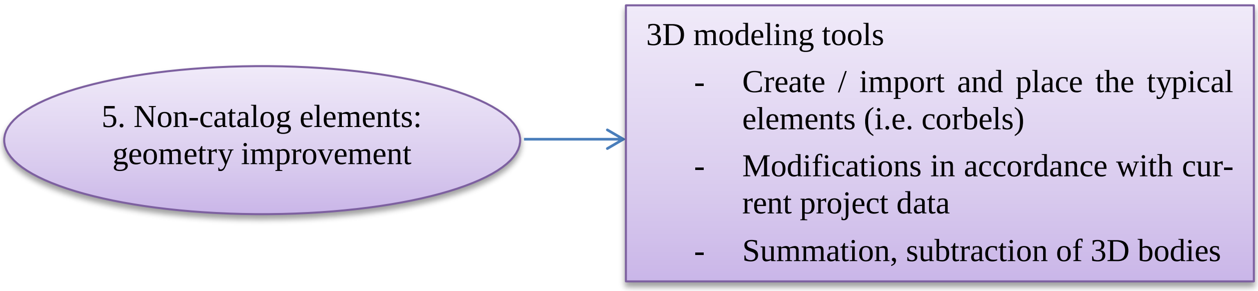


Figure 26. iParts: Locating of insufficiently parameterized elements

5.2 Capabilities of SmartParts

5.2.1 Refill of elements catalog



Out-of-catalog elements can be programmed with SmartParts script, saved and further used as element from SmartParts catalog. Operation in this case can be replaced by set of additional parameters while formulating the initial elements geometry.

In case of modifications the reset of parameters is more efficient than division of element back into the parts, modification of the parts and further summation.

5.2.2 Refill of reinforcement catalog

Due to possible lack of elements in basic catalog (FF components tool) and possible lack of options in parameter sets the programming of more detailed reinforcement allows to avoid some operations corresponding to detailing of reinforcement.

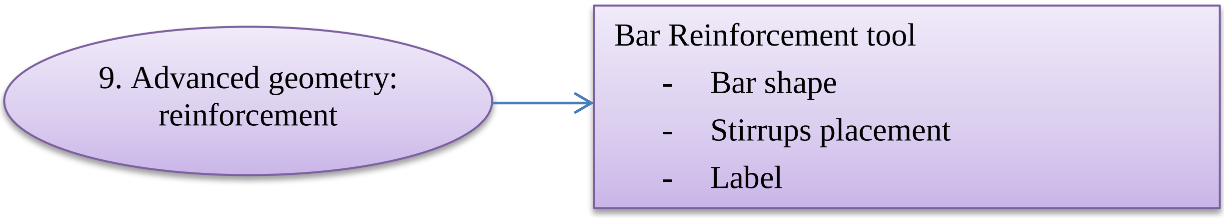


Table 8. Reinforcement design: work flow using various modules

Basic tools	**SmartParts**
FF components tool (basic geometry)	Insert SmartParts
- Reinforcement shape	- choice of file with element
- Geometry	- choice or input of geometry data for element and reinforcement
- Placement	- element placement
Bar Reinforcement tool (advanced geometry)	
- Bar shape	
- Stirrups placement	
- Label	
Reinforcement report	
At least 7 operations	

5.2.3 Design of 3-D build-in parts: transport anchors

Essential parameters for design of transport anchors:

- Type of anchor
- Amount for one element
- Placement on the element surface

Considerations in design

- Type and number of anchors according to weight requirements
- Limitations for anchor placement according to requirements for a particular anchor type

Algorithm aim

- To avoid error-prone operations corresponding to analysis of acceptable types of anchors in accordance with weight of element and analysis of corresponding limitations for anchor placement.

Assumptions

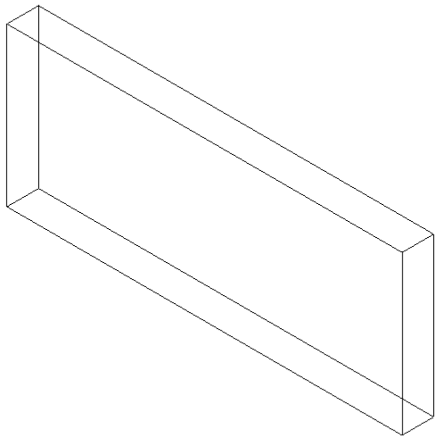


Figure 27. SmartPart: Element to be anchored

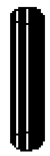


Figure 28. SmartPart: anchor

Precast element is assumed as box-shaped SmartParts element. Anchor is cylinder-shaped SmartParts element. Two anchors are assumed to be placed symmetrically on top surface.

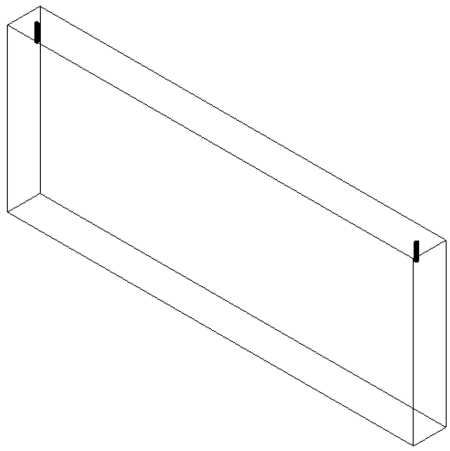


Figure 29. SmartPart: anchored element

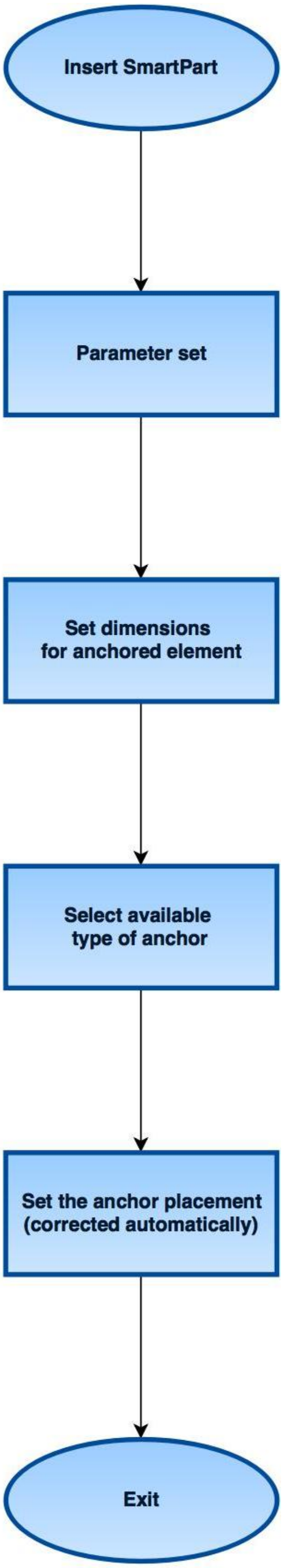


Figure 30. SmartParts: Design of transport anchors

User can modify dimensions of element, location of anchors 1 and 2 along the top surface of the element. Also user can select the anchor type. Acceptable types of anchor are defined automatically. Placement of anchors is corrected automatically.

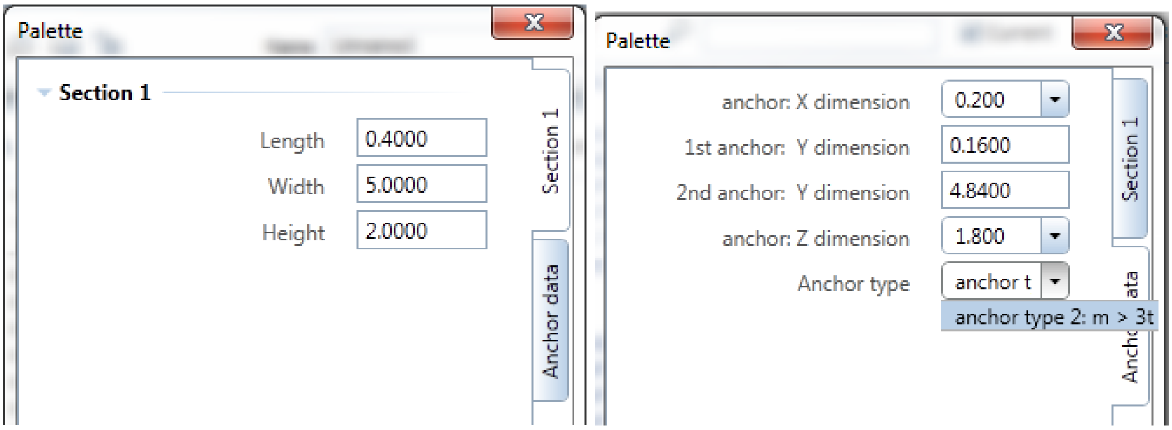


Figure 31. SmartPart with predefined transport anchors: Dialog box

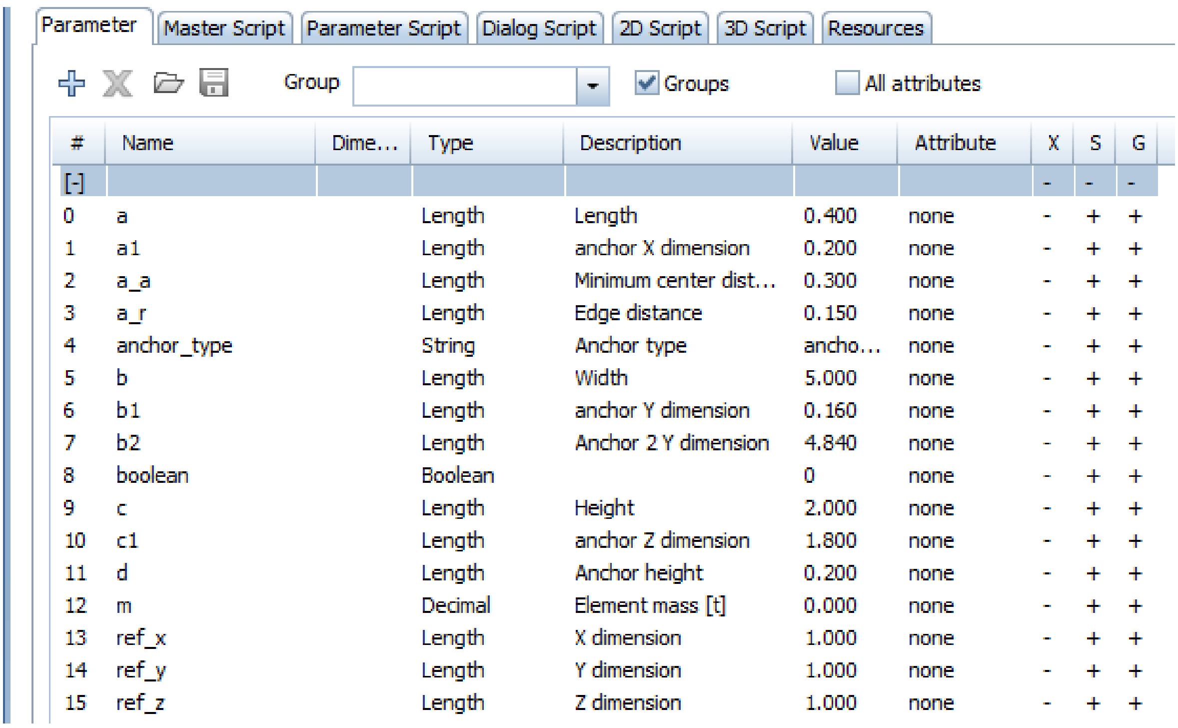
Implementation

Parameter lists

Parameters a, b, c define dimensions of box-shaped element. Parameters a1, b1, b2, c1 define location of anchors. Parameters b1 and b2 define location of anchor 1 and anchor 2 along the top surface of element.

m – mass of the element [t]

Requirements for anchor design are defined as parameters anchor_type, a_a (minimum center distance between anchors), a_r (minimum edge distance) [9]



#	Name	Dime…	Type	Description	Value	Attribute	X	S	G
[-]							-	-	-
0	a		Length	Length	0.400	none	-	+	+
1	a1		Length	anchor X dimension	0.200	none	-	+	+
2	a_a		Length	Minimum center dist…	0.300	none	-	+	+
3	a_r		Length	Edge distance	0.150	none	-	+	+
4	anchor_type		String	Anchor type	ancho…	none	-	+	+
5	b		Length	Width	5.000	none	-	+	+
6	b1		Length	anchor Y dimension	0.160	none	-	+	+
7	b2		Length	Anchor 2 Y dimension	4.840	none	-	+	+
8	boolean		Boolean		0	none	-	+	+
9	c		Length	Height	2.000	none	-	+	+
10	c1		Length	anchor Z dimension	1.800	none	-	+	+
11	d		Length	Anchor height	0.200	none	-	+	+
12	m		Decimal	Element mass [t]	0.000	none	-	+	+
13	ref_x		Length	X dimension	1.000	none	-	+	+
14	ref_y		Length	Y dimension	1.000	none	-	+	+
15	ref_z		Length	Z dimension	1.000	none	-	+	+

Figure 32. SmartPart: anchored element. Parameter list

Parameter script

Mass of precast concrete element m is defined by multiplication of concrete density which is assumed 2.5 t/m^3 and element volume. Volume of box-shaped element is easily defined by its dimensions. Anchor type 1 is available for selection by user if element mass is less / equal 3t while anchor type 2 is acceptable if element mass is more than 3t.

Depending on anchor type the limitations are to be applied:

Anchors are to be placed at least at 0.31m (anchor type 1) / 0.30m (anchor type 2) from the edge of precast element while distance between anchors is at least 0.16m (anchor type 1) / 0.15m (anchor type 2)

```
VALUES a RANGE [ 0 , 1000 ]
VALUES b RANGE [ 0 , 1000 ]
VALUES c RANGE [ 0 , 1000 ]


m = 2.5 * a * b * c
IF m <= 3 THEN VALUES anchor_type "anchor type 1: m <= 3t"
IF m > 3 THEN VALUES anchor_type "anchor type 2: m > 3t"

IF anchor_type = "anchor type 1: m <= 3t" THEN
   VALUES a_a 0.31
   VALUES a_r 0.16
ENDIF
IF anchor_type = "anchor type 2: m > 3t" THEN
   VALUES a_a 0.30
   VALUES a_r 0.15
ENDIF


VALUES a1 a / 2
VALUES b1 RANGE [ a_r , b - a_r - a_a ]
VALUES b2 RANGE [ b1 + a_a , b - a_r ]
VALUES c1 c - d
```

Figure 33. SmartPart: anchored element. Parameter script

Following script defines user options for parameter set.

Section 1 defines element dimensions. Section Anchor data defines location of transport anchors by parameters a1, b1, b2, c1 as well as anchor type

```
I_PALETTE "Palette"
i_section "Section 1"

I_PAGE "Section 1"
I_INFIELD "a", 0, 0, 0, 0
I_INFIELD "b", 0, 0, 0, 0
I_INFIELD "c", 0, 0, 0, 0
I_SECTION_END


I_PAGE "Anchor data"
I_INFIELD "a1", 0, 0, 0, 0
I_INFIELD "b1", 0, 0, 0, 0
I_INFIELD "b2", 0, 0, 0, 0
I_INFIELD "c1", 0, 0, 0, 0
I_INFIELD "anchor_type", 0, 0, 0, 0
I_SECTION_END
```

Figure 34. SmartPart: anchored element. Dialog script

3-D script

Element is defined as box with dimensions a, b, c. Anchor is defined as cylinder with h=0.2m and diameter 0.02m. SmartPart anchor is stored as “a.smt”. Using command CALL two anchors are placed with coordinates a1, b1, c1 and a1, b2, c1.

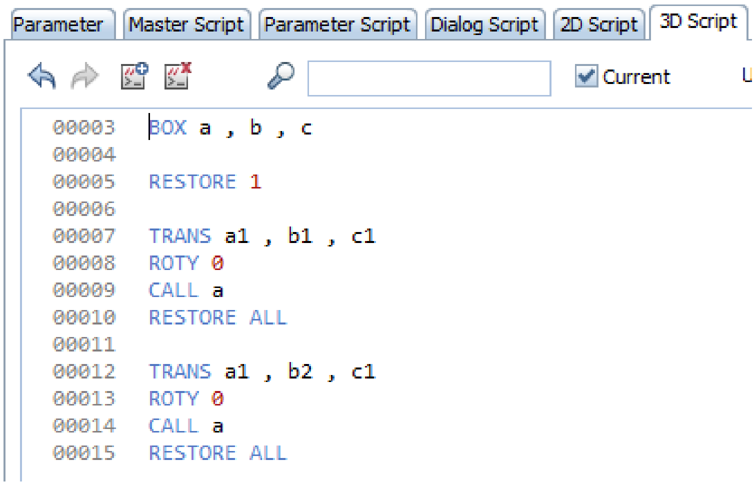


Figure 35. SmartPart: anchored element. 3D script

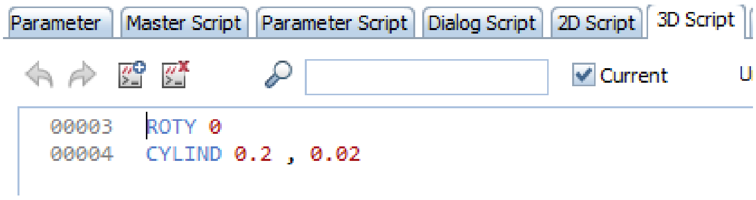


Figure 36. SmartPart: transport anchor. 3D script

Limitations in application

- Anchored element has to be parametrically defined as SmartPart using SmartParts script commands. Imported external objects which can be included into SmartPart are not taken into account. Script is designed to use parameters which define the dimensions of anchored element. If those parameters are missing they have to be defined by user.
- Volume of element has to be determined by user. Unfortunately element volume or weight can’t be provided by SmartParts script. Presented script after modification is able to provide for user the input option for the element volume / weight which can be easily determined by other Allplan’s tools.

Outlook

- A database of requirements for transport anchors (limitations for amount and placement) can be stored and used in the script in the way that user doesn’t have to consider this data him- / herself.
- In the same manner anchor placement can be implemented considering reinforcement of anchored element. If parameters for reinforcement are defined by script they can be used to place anchors at certain distance from reinforcing bars.Furthermore, element-dependent anchors can be replaced by reinforcement-dependent anchors. Since precast element has to be detailed and 3-D build-in parts have to be designed this element also has to be reinforced. While placement of anchor depends from reinforcement the placement of reinforcement depends on precast element.

5.2.4 Locating of insufficiently parameterized elements

If the element is missing some parameter data then the element is considered as insufficiently parameterized. Lack of data input can lead to appearance of mistakes in element plans.

Priorities for check of parameters

Design automation leads to higher requirements for control of results. Automatic generation of results can lead to mistakes. Those parameters which are specified for further transfer to next stages of BIM workflow – should be checked. In general, parameters which can't be checked in further department, parameters which are directly related to current stage detailed design and also to earlier stages - must be checked.

Example of SmartParts parameter set

Available by default SmartPart element Sleeve foundation.

- General
    - Standard
        - Concrete grade, Steel grade
    - Reinforcement
        - Layer options, Visualization
    - Foundation
        - Foundation type (rough, smooth), concrete cover for foundation and for sleeve
    - Object settings
        - Visualization, Layer options, Color settings etc.
- Geometry
    - Foundation geometry
    - Sleeve geometry, Geometry of sleeve wall
- Longitudinal reinforcement
    - in X direction, in Y direction
        - Bar diameter, offset, hook length
- X-sleeve wall
    - in X direction – Vertical stirrup, in Y direction – Horizontal stirrup
        - Bar diameter, bar spacing in center and from edge, hook length / overlap length, bending pin diameter
- Y-sleeve wall
    - in X direction – Vertical stirrup, in Y direction – Horizontal stirrup
        - Bar diameter, number of bars in corner, bar spacing in and from center and from edge, hook length / overlap length, bending pin diameter

Example of iParts parameter set

iPart element type Foundation

- Geometry
    - General
        - Input option (parameters, 3-D object)

        - Shape (Rectangle, Hexagon)
    - Dimensions
        - Widths, thickness, socket (without, attached, cut out), piles settings
    - Socket position
        - If not centered – reference point, eccentricities
    - Socket geometry
        - Widths, enlarge at top, wall thicknesses, wall height, depth, profiling
- Attributes
    - Factory
    - Label
        - Label, mark number, additional text, component name, text anchor point
    - Classification
        - Concrete grade, Exposure class, Fire resistance classification
- Dimensions
    - Alignment
    - Loading dimension
    - Production dimensions
- Invoicing
    - Invoice item
    - Invoice for customer
    - Invoice for factory

SmartPart script locates insufficient parameter sets

Based on demonstrated earlier example of transport anchor design the algorithm is able to locate insufficient parameter sets for any other SmartPart element.

Algorithm aim

- To check whether parameters fulfill the requirements and to inform user about wrong or missing parameters

Parameters to be checked

- Type of acnhor

Design requirements

- Type of anchors according to weight requirements

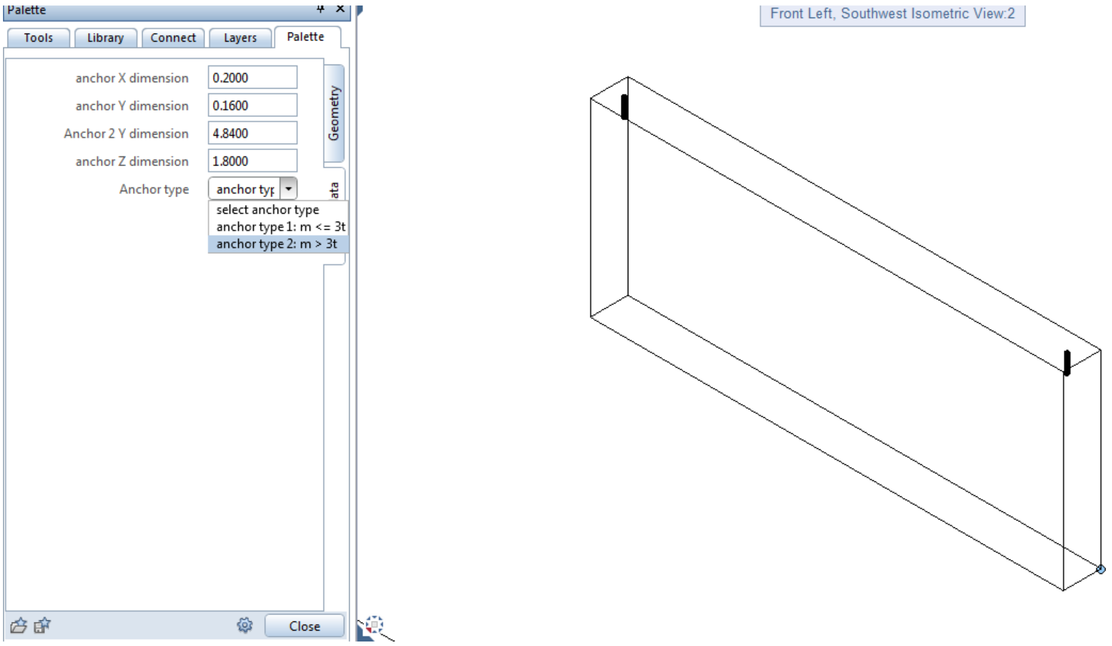


Figure 37. Smart Part. Parameter set: Selection of anchor type

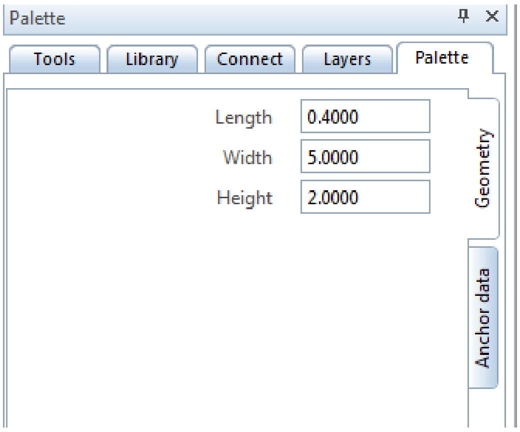


Figure 38. SmartParts. Correctly defined parameter set..

The algorithm informes user about wrong/missing parameters by appearance of additional parameter section „Wrong parameters“ or „Missing parameters“. Section contains these parameters structutred in the list. Section provides possibility to reset missing/wrong parameters.

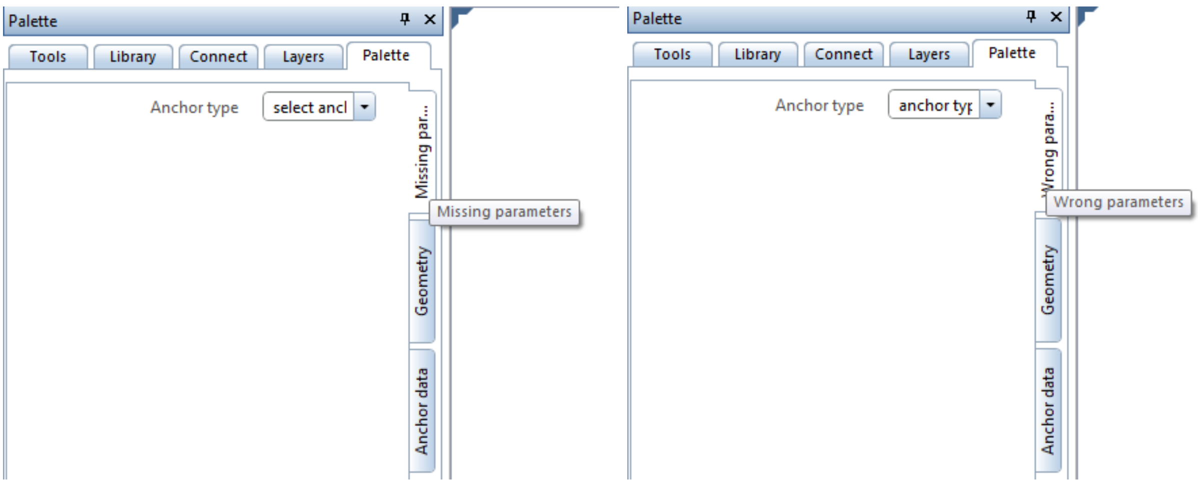


Figure 39. Smart Part. Parameter set: Section Missing / Wrong parameters

Algorithm also informes user by appearance of error window „SmartPart cannot be generated correctly“ with possibility to agree with a mistake (further reset of parameters is possible) or cancel the parameter set

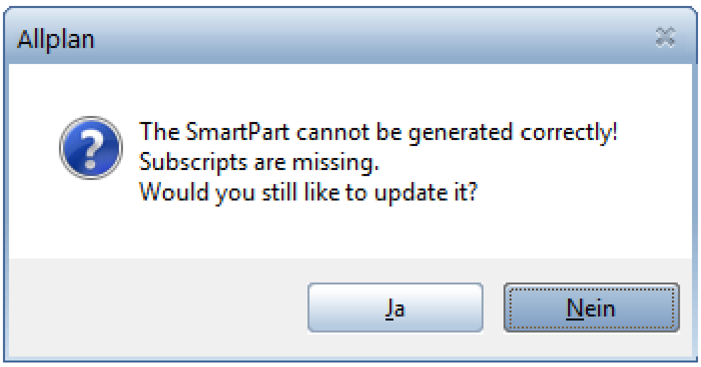


Figure 40. SmartPart. Error window

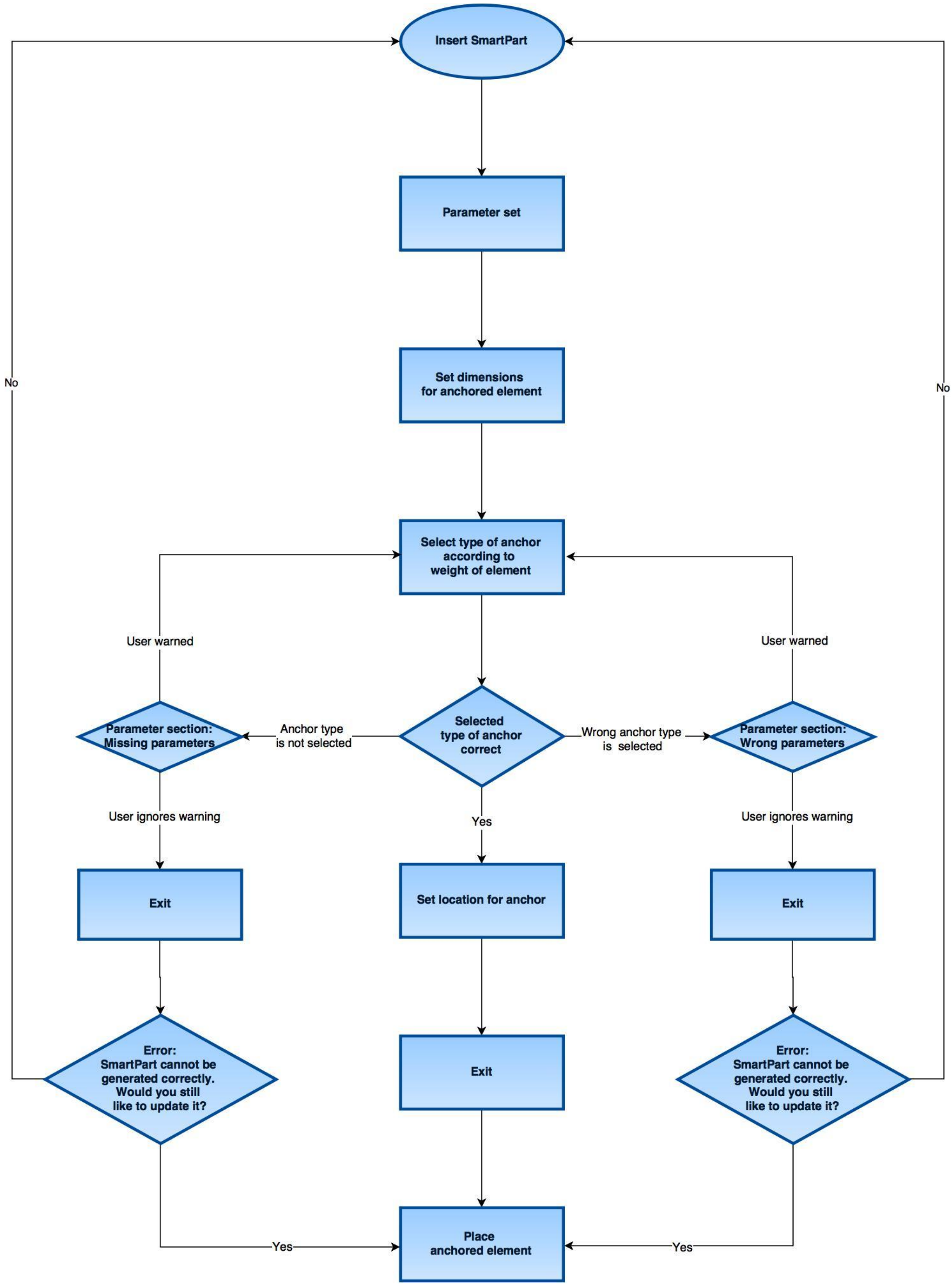


Figure 41. SmartParts: locating of insufficient parameter sets

Implementation

#	Name	Dime...	Type	Description	Value	Attribute	X	S	G
0	a		Length	Length	0.400	none	-	+	+
1	a1		Length	anchor X dimension	0.200	none	-	+	+
2	a_a		Length	Minimum center dist...	0.300	none	-	+	+
3	a_r		Length	Edge distance	0.150	none	-	+	+
4	anchor_type		String	Anchor type	ancho...	none	-	+	+
5	b		Length	Width	5.000	none	-	+	+
6	b1		Length	anchor Y dimension	0.160	none	-	+	+
7	b2		Length	Anchor 2 Y dimension	4.840	none	-	+	+
8	boolean		Boolean		0	none	-	+	+
9	c		Length	Height	2.000	none	-	+	+
10	c1		Length	anchor Z dimension	1.800	none	-	+	+
11	d		Length	Anchor height	0.200	none	-	+	+
12	m		Decimal	Element mass [t]	0.000	none	-	+	+
13	Parameter_is_missing		String		none	none	-	+	+
14	Parameter_is_wrong		String		none	none	-	+	+
15	ref_x		Length	X dimension	1.000	none	-	+	+
16	ref_y		Length	Y dimension	1.000	none	-	+	+
17	ref_z		Length	Z dimension	1.000	none	-	+	+

Figure 42. SmartPart: Parameters „Parameter is missing“ and „Parameter is wrong“

Parameter anchor_type has 3 options for user to be selected: Select anchor type, Anchor type 1, Anchor type 2. Parameter_is_missing and Parameter_is_wrong define whether corresponding parameter section appears in dialog box as warning for user. If selected anchor type doesn’t fit with correct value script assigns to parameter_is_wrong value „anchor type“. Also command CALL „unexisting file“ is used to cause the appearance of error window after modeling is finished.

```
m = 2.5 * a * b * c
VALUES anchor_type "select anchor type" , "anchor type 1:

IF anchor_type = "select anchor type" THEN
   VALUES parameter_is_missing "anchor type"
   CALL "unexisting file"
ELSE
   VALUES parameter_is_missing "none"
ENDIF

IF m <= 3 THEN
   IF anchor_type = "anchor type 2: m > 3t" THEN
      VALUES parameter_is_wrong "anchor type"
      CALL "unexisting file"
   ELSE
      VALUES parameter_is_wrong "none"
   ENDIF
ENDIF

IF m > 3 THEN
   IF anchor_type = "anchor type 1: m <= 3t" THEN
      VALUES parameter_is_wrong "anchor type"
      CALL "unexisting file"
   ELSE
      VALUES parameter_is_wrong "none"
   ENDIF
```

Figure 43. SmartParts: parameter script

Limitation in application

Algorithm doesn’t have limitations. It has potential to check any parameters of SmartParts element.

Identical elements tool for locating of insufficiently parameterized objects

Other person also sets the same parameters for the elements. After engineer finished a model those elements are to be imported. If objects are not identical with the imported ones then they miss some of parameters.

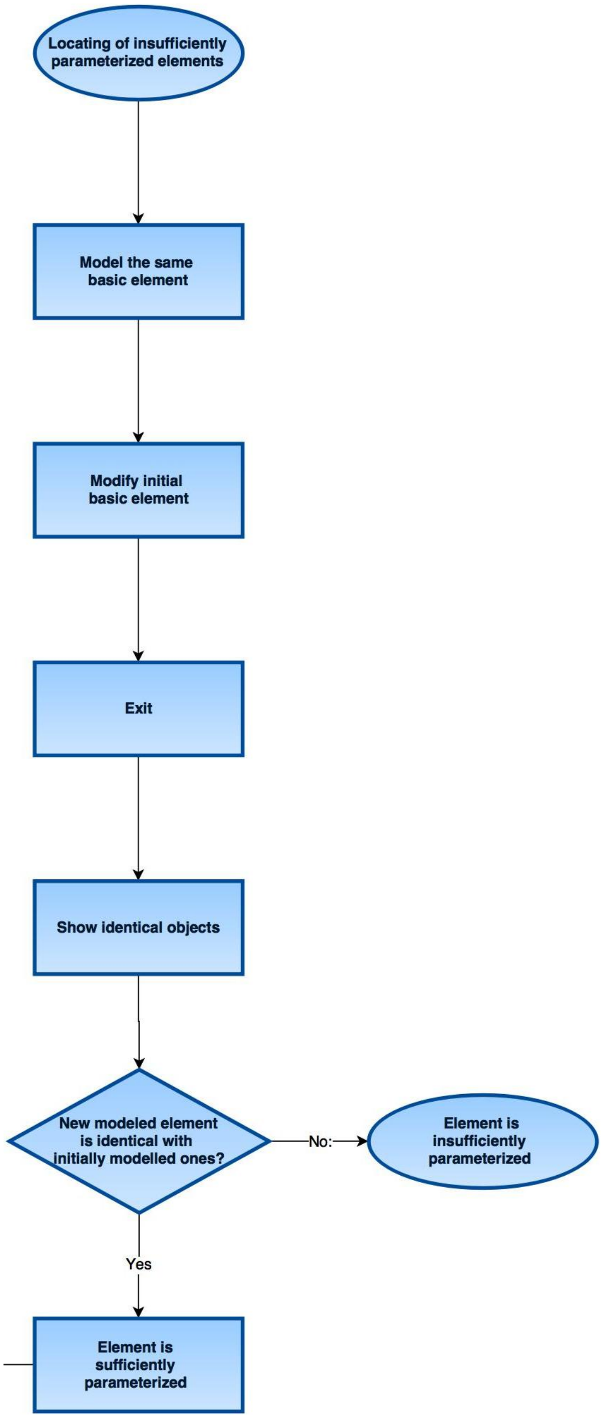


Figure 44. SmartParts: Locating of insufficiently parameterized elements

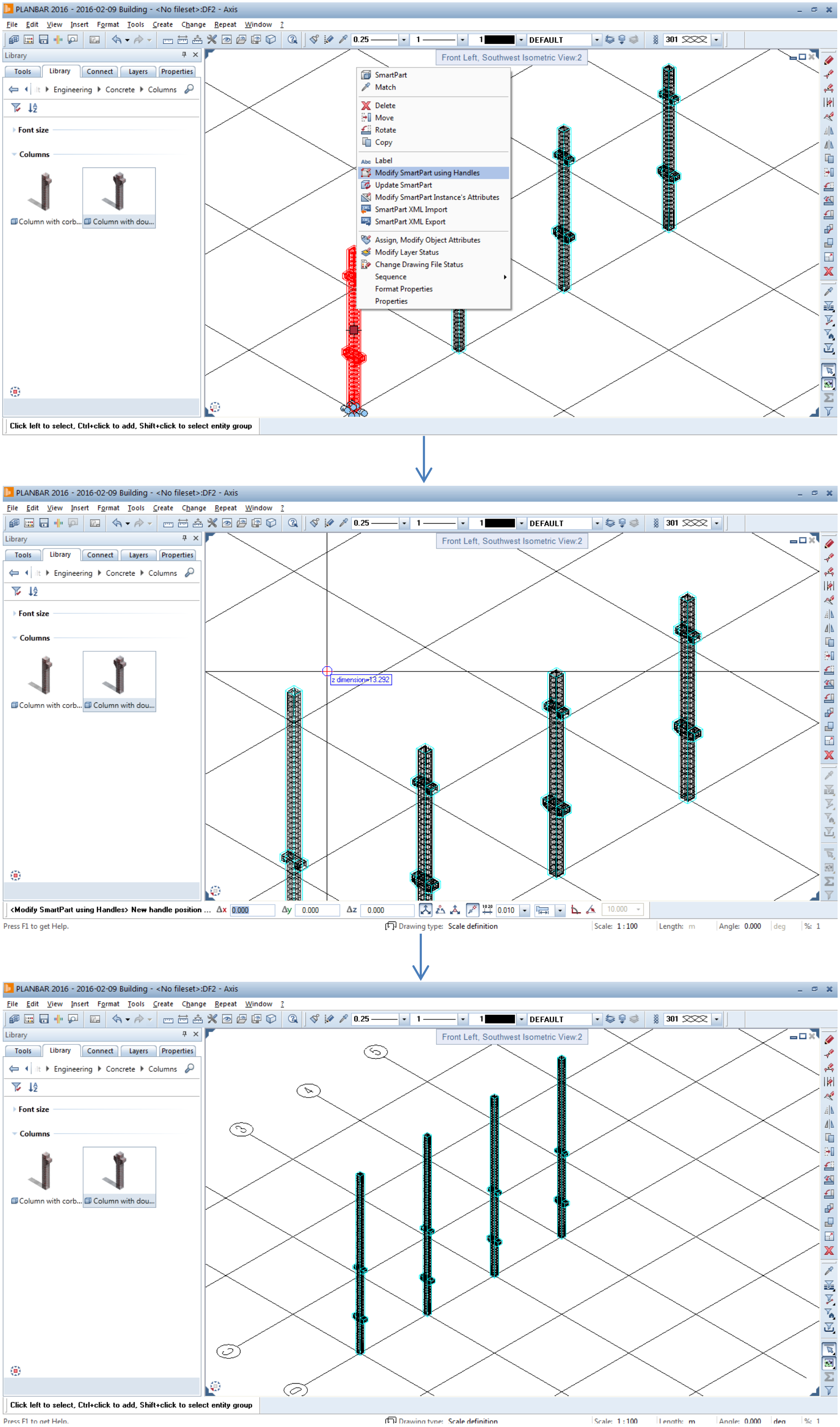


Figure 45. Remodification of SmartPart elements using identical elements tool

6 USAGE SCENARIOS FOR MODULES SMARTPARTS AND IPARTS

6.1 Usage scenario 2

Module SmartParts are able to provide more efficient detailed design of single elements including build-in parts and reinforcement.

Module iParts supports modeling of parametric relations between elements and automatic generation of user-predefined element plans. In addition, SmartParts can be indirectly transformed into iParts.

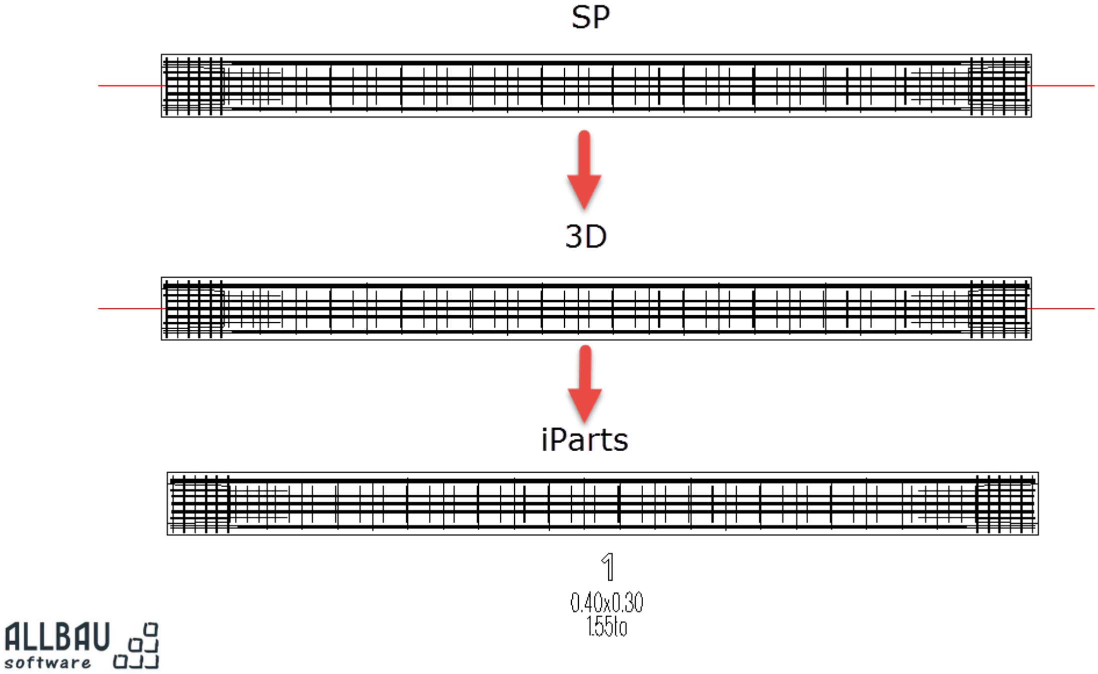


Figure 46. Concept of scenario 2 (SP – SmartParts) [10]

Combined usage of SmartParts and iParts has potential to simplify modeling by partial automation of detailing, connection design and plan generation.

6.2 Usage scenario 3

Considering that:

- parametric objects are never absolutely universal since there will be always new more complex objects to design

- only a few of detailed SmartPart elements are provided by default

- SmartParts programming takes more effort/costs

Usage scenario 2 isn't realistic. The full potential of SmartParts can be used only partially depending on individual cases. Hence, some of operations have to replaced by the ones from scenario 1. Realistic scenario for usage of modules is the combination of scenarios 1 and 2.

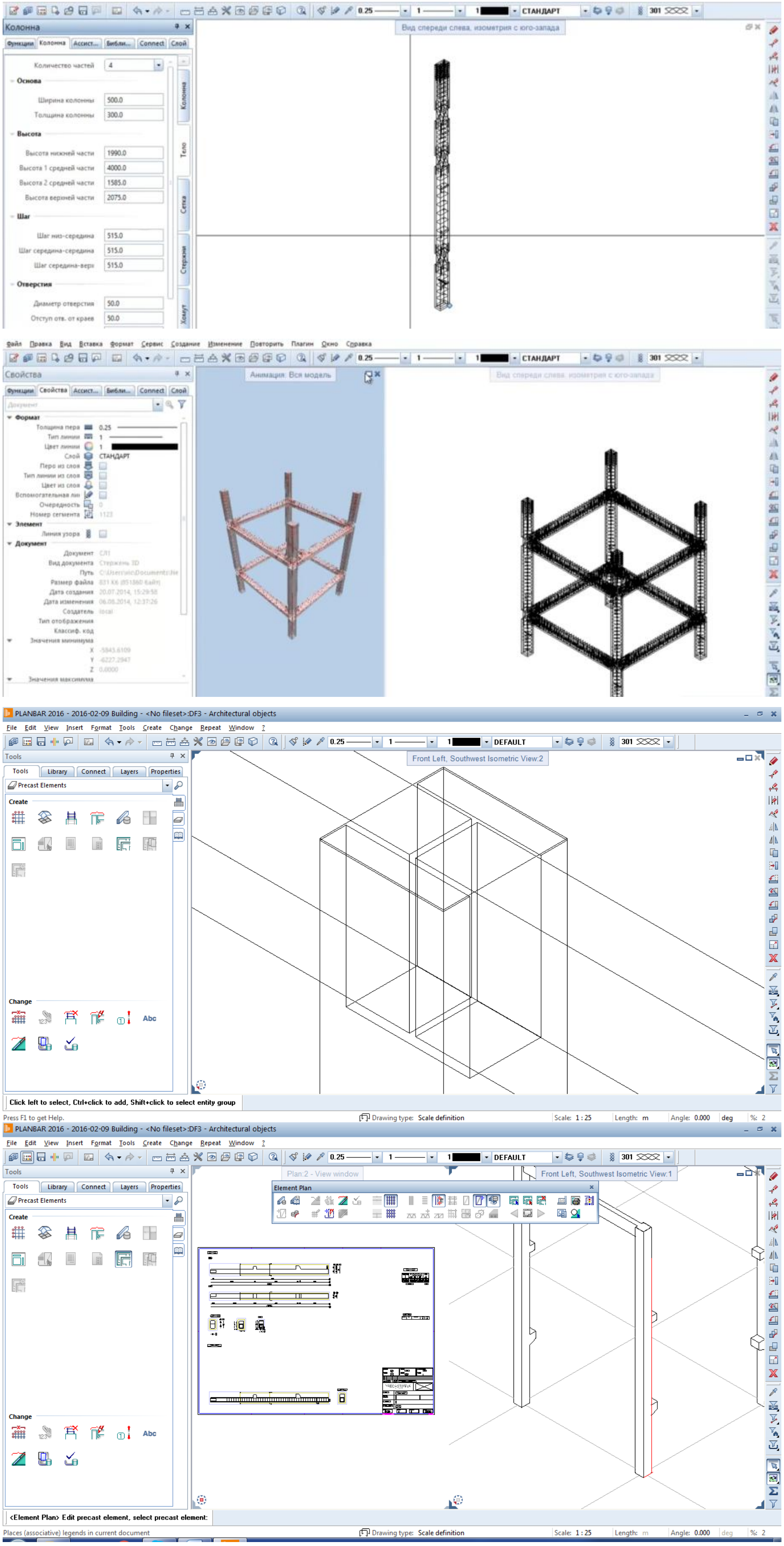


Figure 47. Scenario 2. Modelling of elements - with SmartParts; connection design and plan generation - with iPrats

6.3 Workflows

Automatable operations are marked red. Replacing operations are marked green.

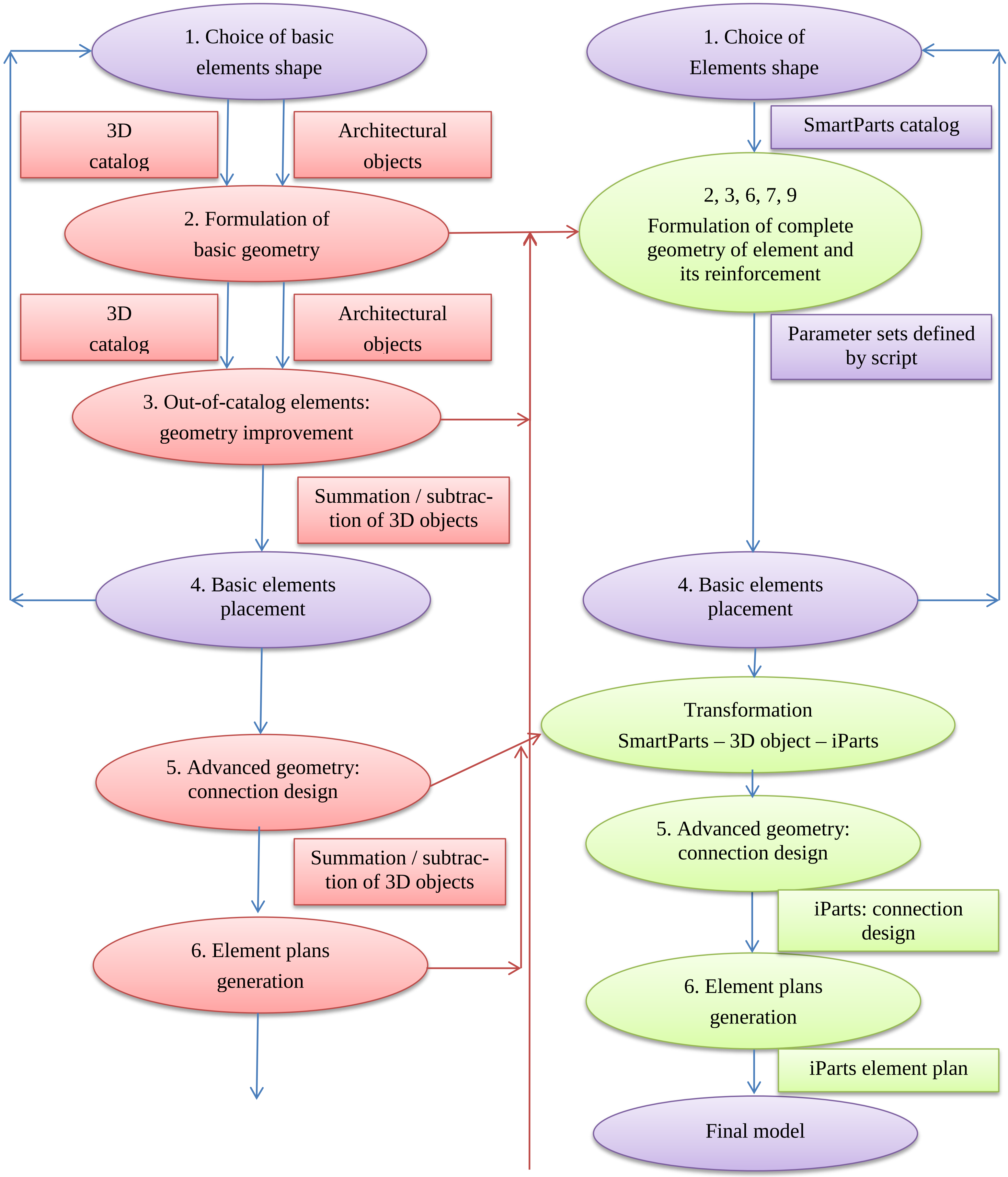


Figure 48. Scenario 2 (Part 1) Modelling of elements - with SmartParts; connection design and plan generation - with iPrats

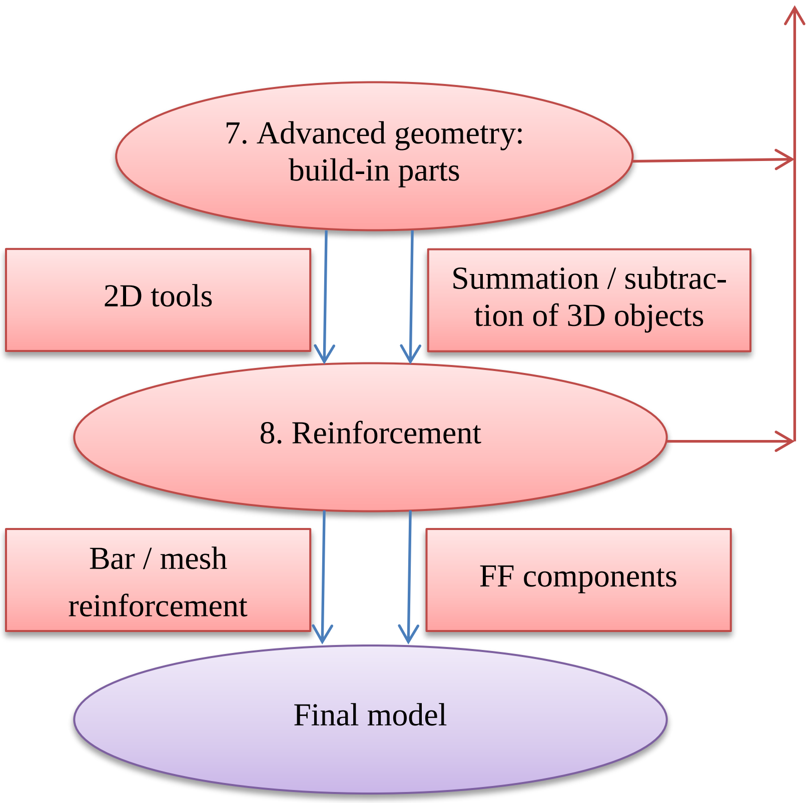


Figure 49. Scenario 2 (Part 2)
Modelling of elements - with SmartParts; connection design and plan generation - with iPrats

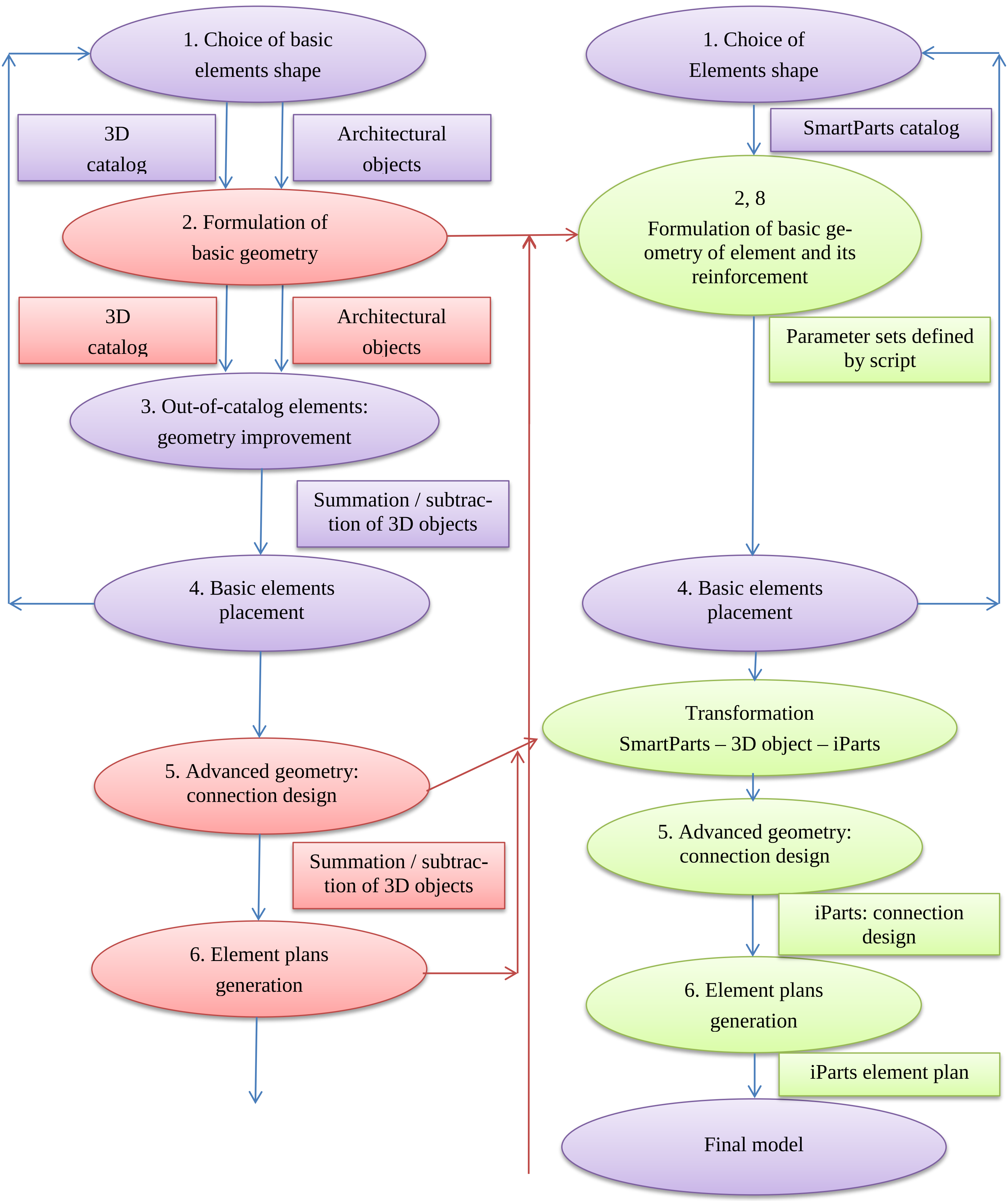


Figure 50. Scenario 2 (Part 1) Modelling of elements - with SmartParts; connection design and plan generation - with iPrats

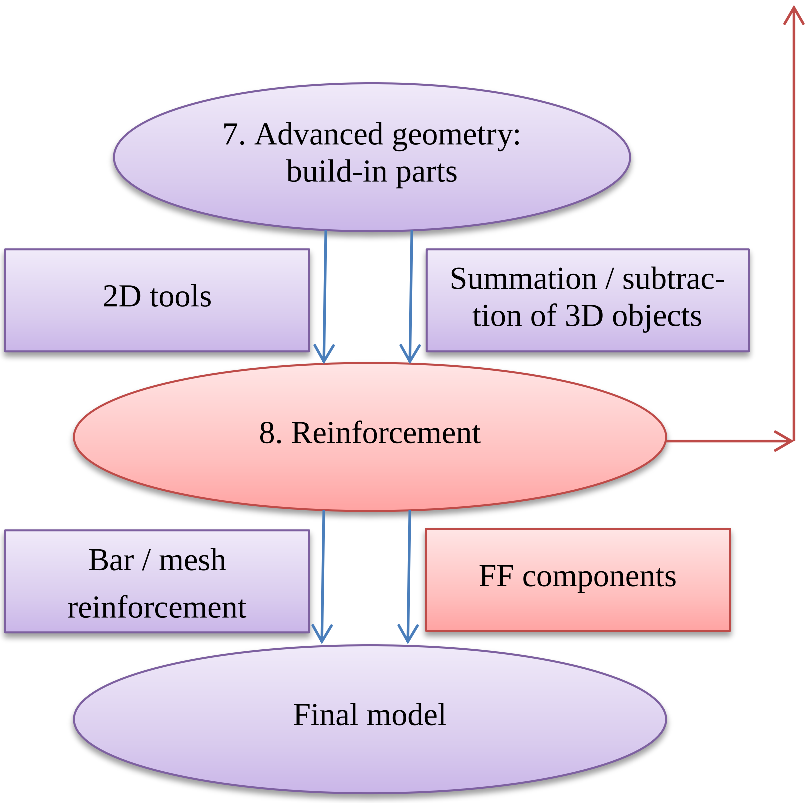


Figure 51. Scenario 2 (Part 2)
Modelling of elements - with SmartParts; connection design and plan generation - with iParts

6.4 Prototypical implementation of BIM model using modules SmartParts and iParts

The flowchart is aimed to describe the design of precast elements in phases 4 and 5 acc. HOAI[1]: approval planning and detailed planning (orig. Genehmigungsplanung; Ausführungsplanung). Flowchart is based on implemented models. Nevertheless it is universal and designed to be used for other projects. Flowchart suggests the variety of usage scenarios. Green arrows are used to describe the usage scenario 3.

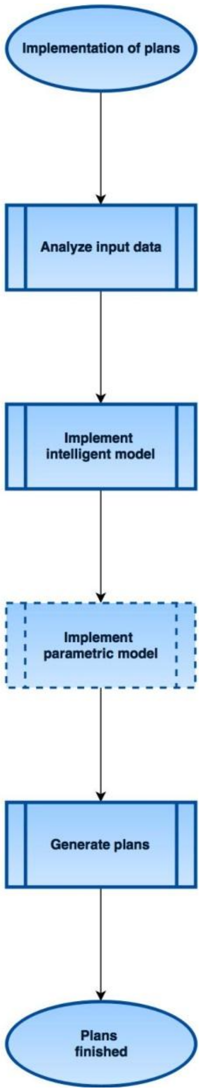


Figure 52. Implementation of element plans

[1] Overall performance of architects and engineers in Germany is divided into phases according to HOAI (Fees for architects and engineers, orig. Honorarordnung für Architekten und Ingenieurleistungen).

[2] A distinction is made between a parametric building model and an intelligent building model. In parametric building model, all elements can be mutually brought into dependencies (walls, ceilings, dimensions, annotations, objects, cutting lines, etc.), while the intelligent building model intelligence is limited to individual objects.

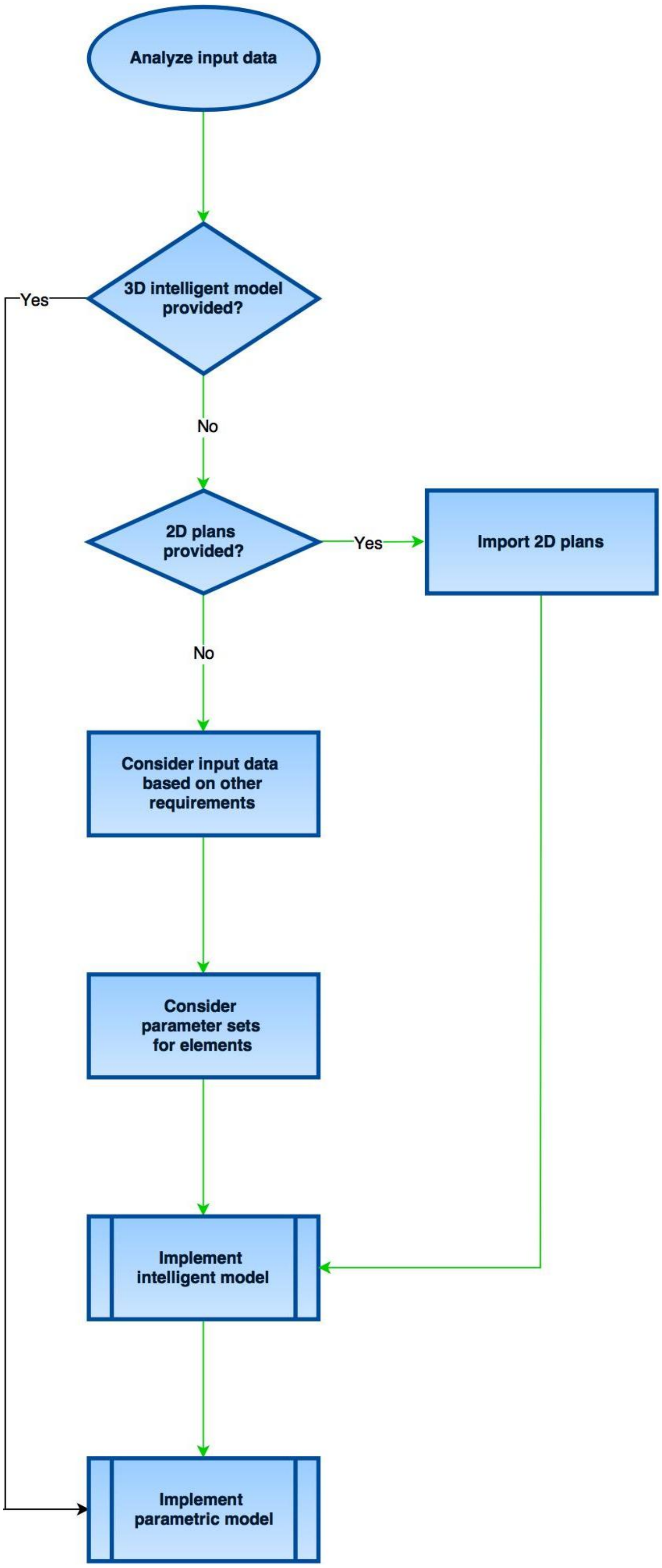


Figure 53. Analyze input data

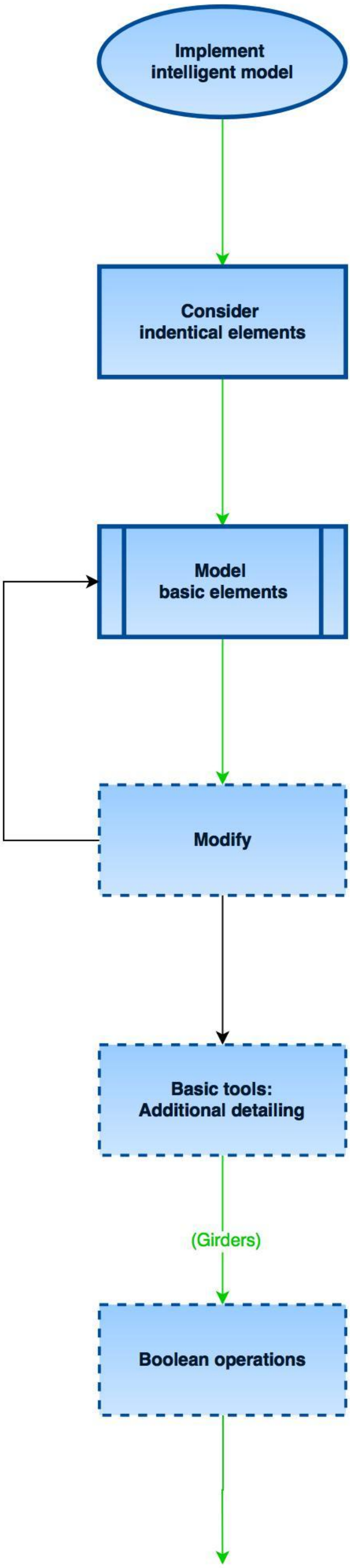


Figure 54. Implement intelligent model. Part 1

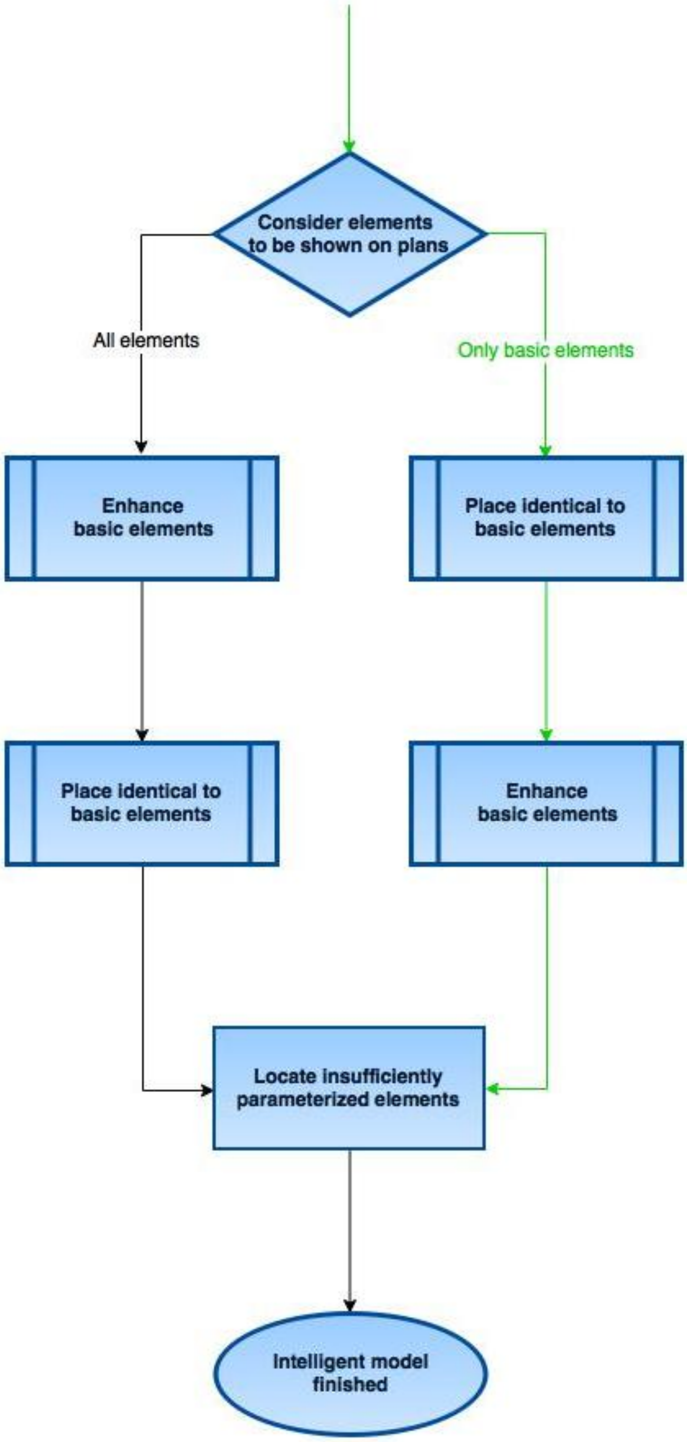


Figure 55. Implement intelligent model. Part 2

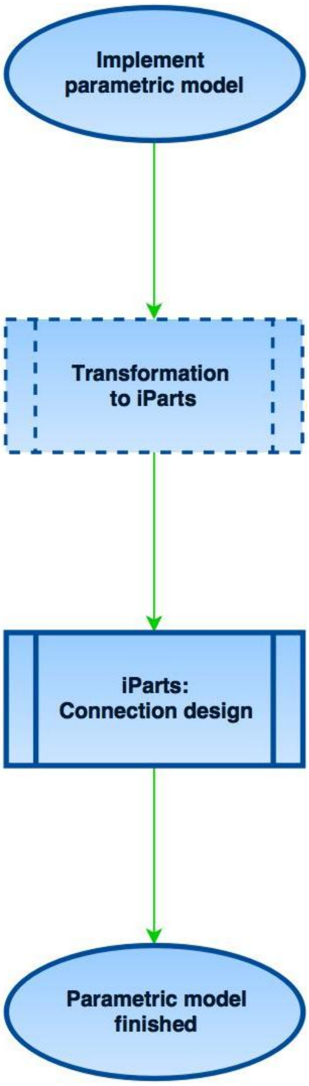


Figure 56. Implement parametric model

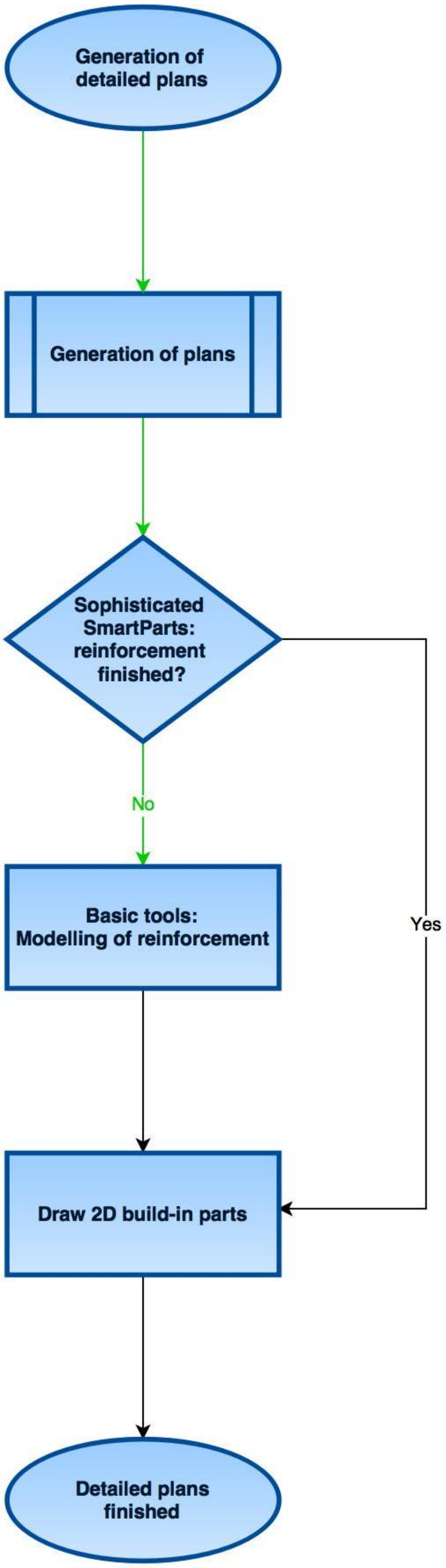


Figure 57. Generation of detailed plans

6.4.1 Analysis of input data

Import of 2D drawing

Provided 2D drawings have differences in arrangement and element plans. Based on provided data the combined solution is used to consider the parameters for modeling. Hence, the drawings can be used in modeling only partially. Nevertheless, arrangement plan is imported and also used as axis net.

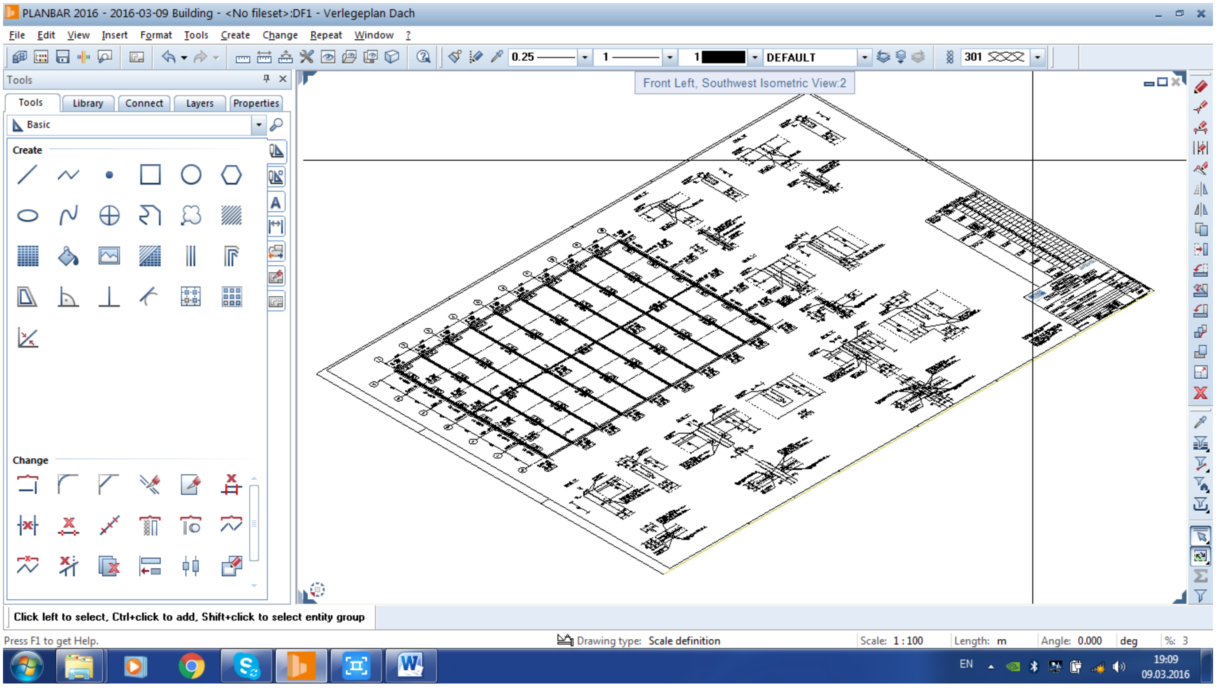


Figure 58. Import of 2D drawing

6.4.2 Intelligent modeling

Consider identical elements:

Elements are mainly divided into groups on basis of geometry of cross section, connections and additional details such as corbels. Since the connection design is out of concern at current stage the geometry of cross section and corbels remain decisive parameters for division of elements into groups.

Columns

Parameter sets:

Columns along axis A, G:	Section 0.4x0.4m, 2 corbels
Columns along axis B, C, E, F:	Section 0.4x0.4m, no corbels
Columns D1, D10:	Section 0.4x0.6m, 4 corbels
Columns D4, D7:	Section 0.6x0.6m, 4 corbels

Beams

Beams along axis 2-9:	Length 18m
Beams along axis 1, 10:	Length 6m

Girders

Girders D1-D4, D7-D10:	Length 18m, 4 corbels, end supported by console
Girder D4-D7:	Length 18m, 4 corbels, both ends supported by consoles

Consider software tools for modeling

Columns: SmartParts

Shape of columns is available in all element catalogs. SmartParts catalog in this case has advantage. SmartPart column has parameters for modeling of reinforcement. It allows partiall avoiding of this step at further stages.

Beams (axis 1, 10): iParts

Beams have simple quadratic shape. None of catalogs is able to provide a strong advantage. In this use of iParts allows avoiding the transformation of these elements to iParts since they are initially modeled as iParts.

Beams (axis 2-9) and girders: basic tools

Beams and girders have complicated shapes which are not provided by other catalogs. Moreover, complete shape of some elements can be provided only partially by tool Concrete Construction – 3D object.

The variety of available initial data is assumed, either this is model or plans at stage of preliminary or approval planning as well as lack of initial data for modeling.

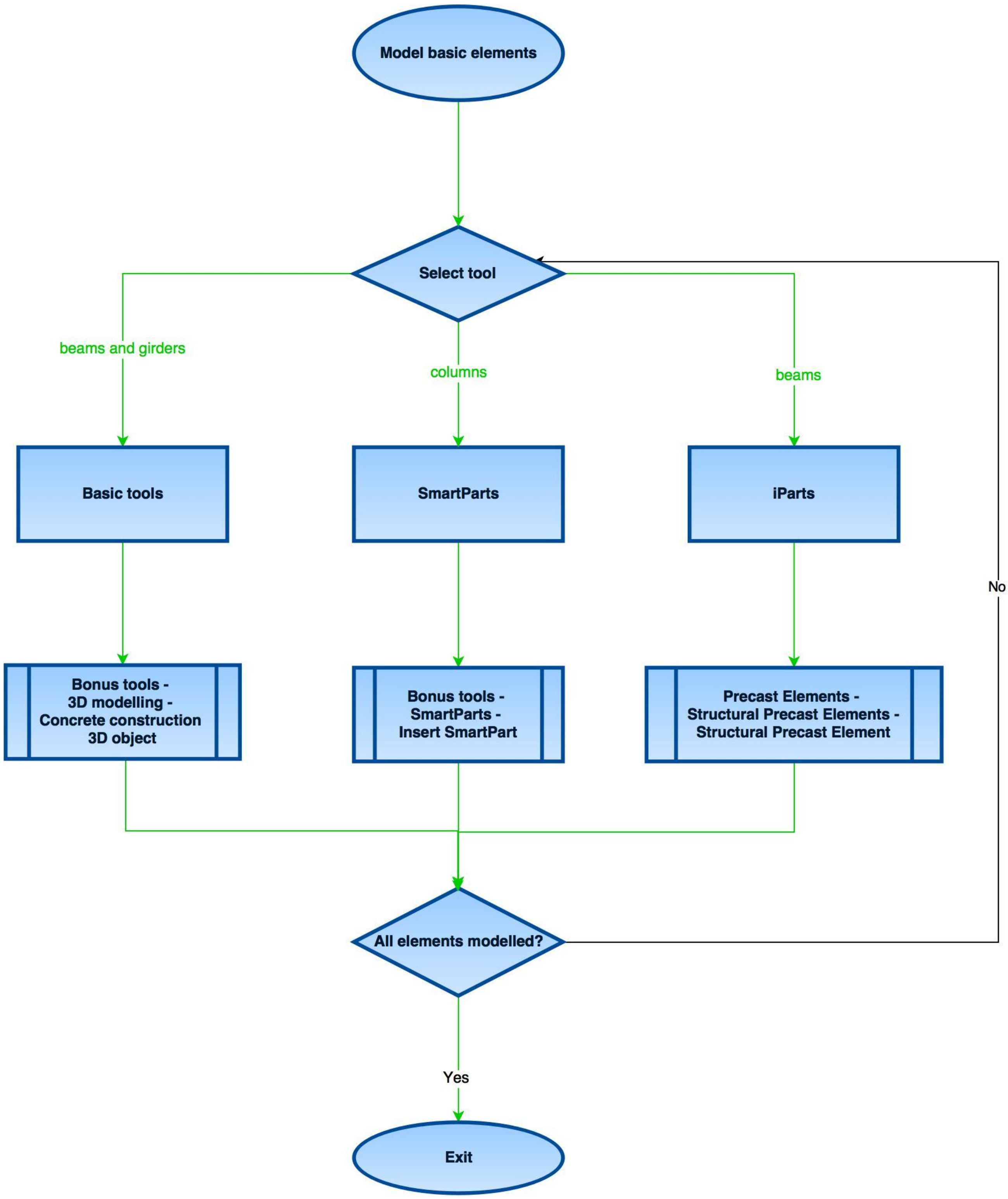


Figure 59. Model basic elements

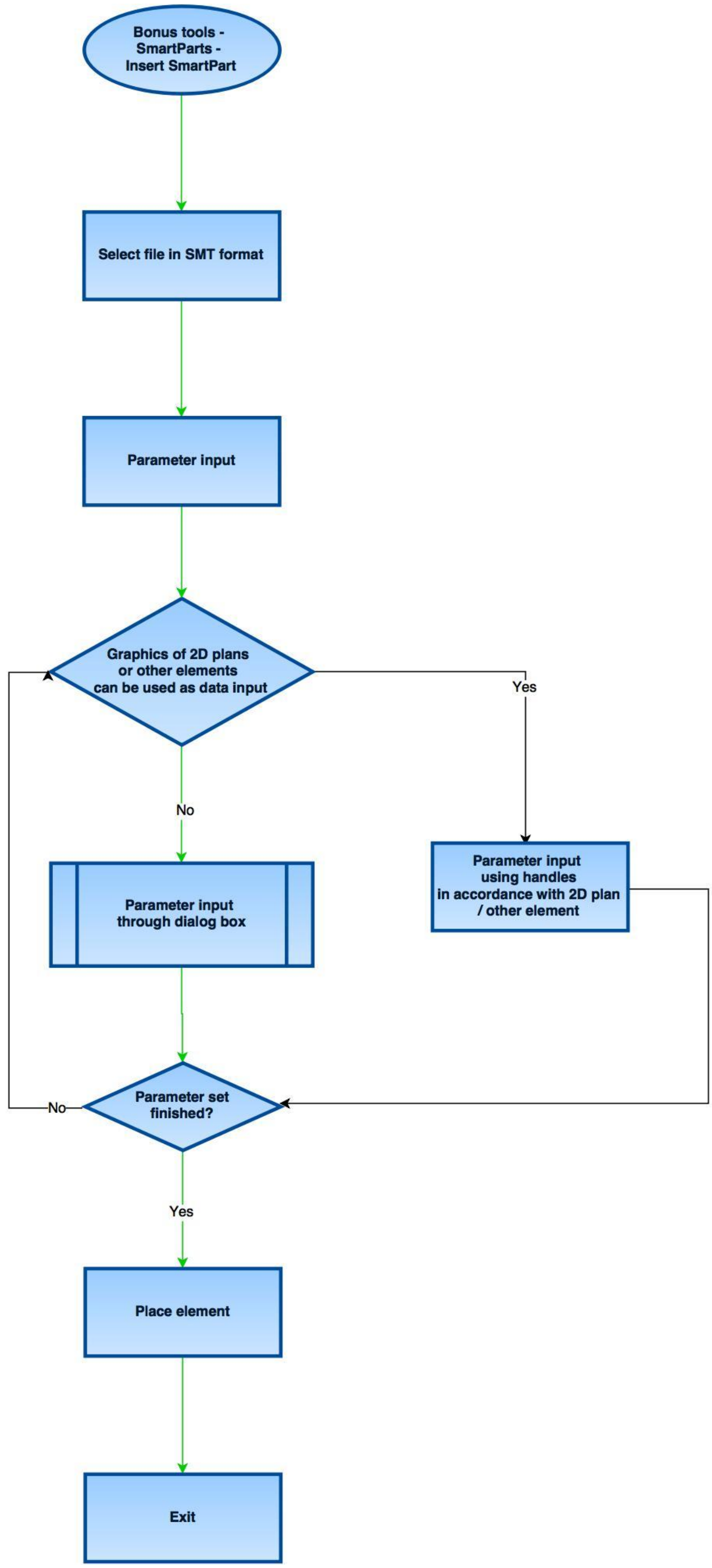


Figure 60. SmartParts: parameter set

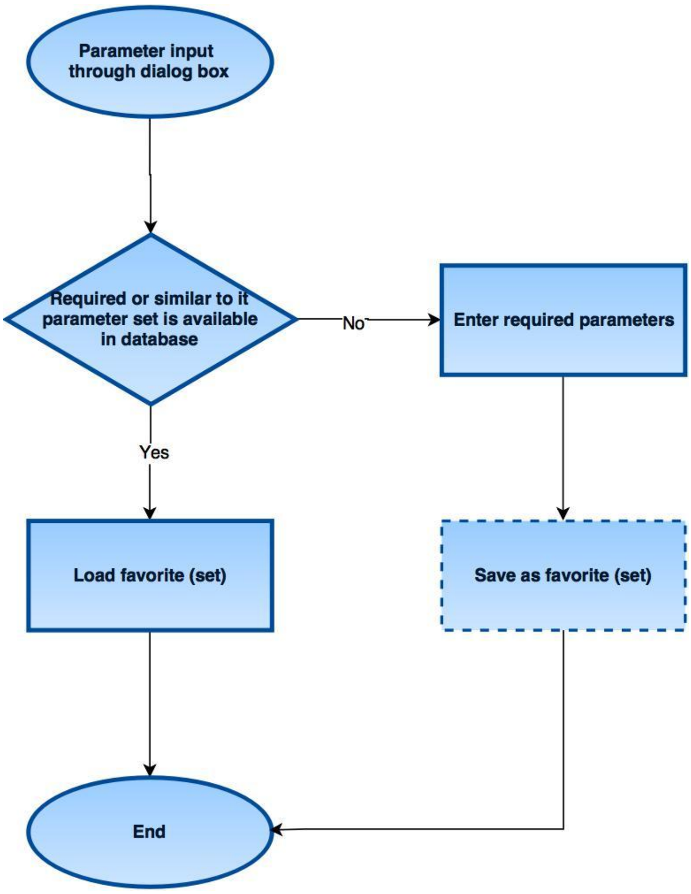


Figure 61. Parameter input through dialog box

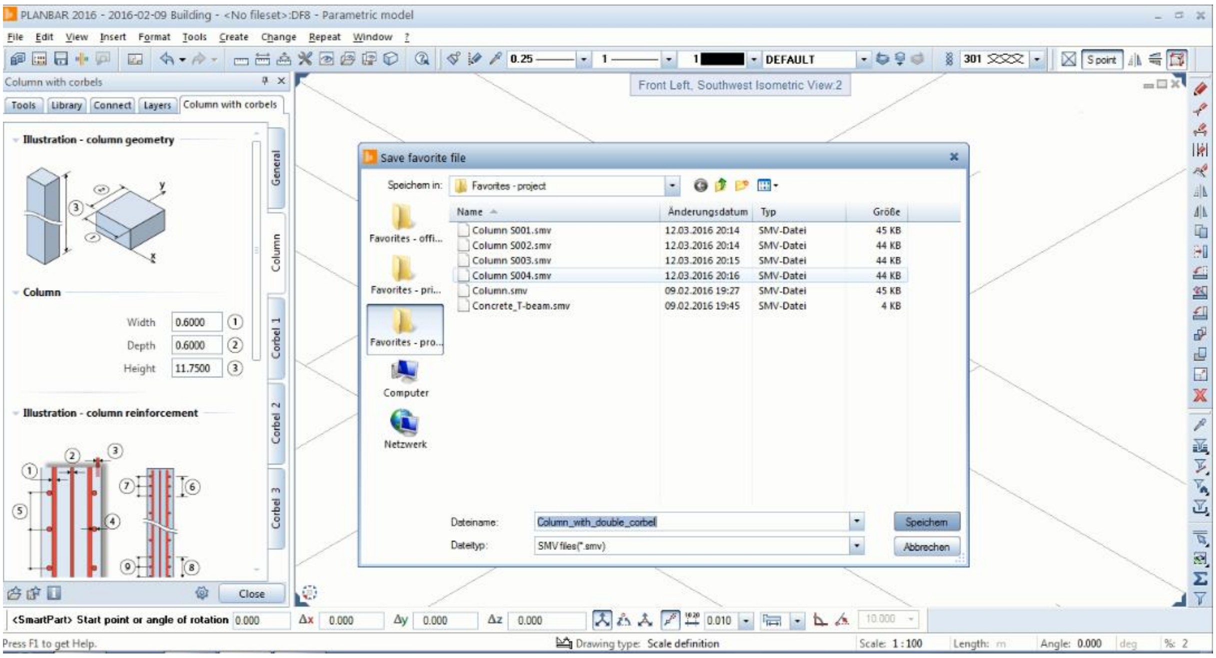


Figure 62. Implemented model: SmartPart column. Parameter set

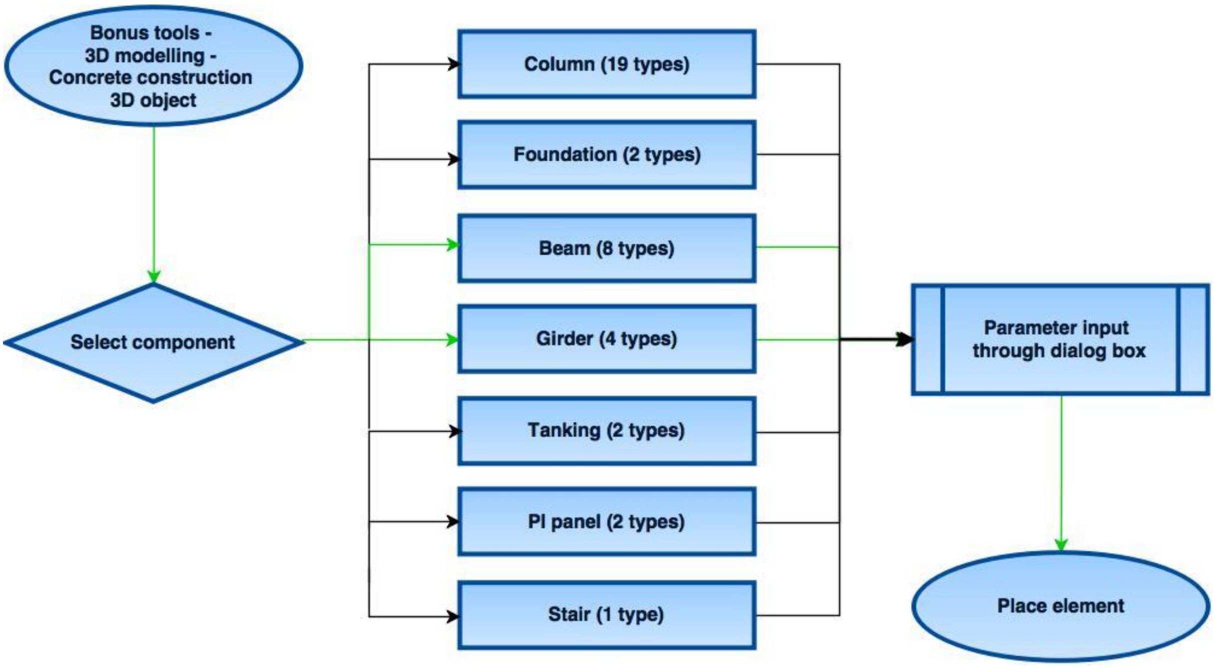


Figure 63. Basic tools – Concrete construction 3D object: parameter set

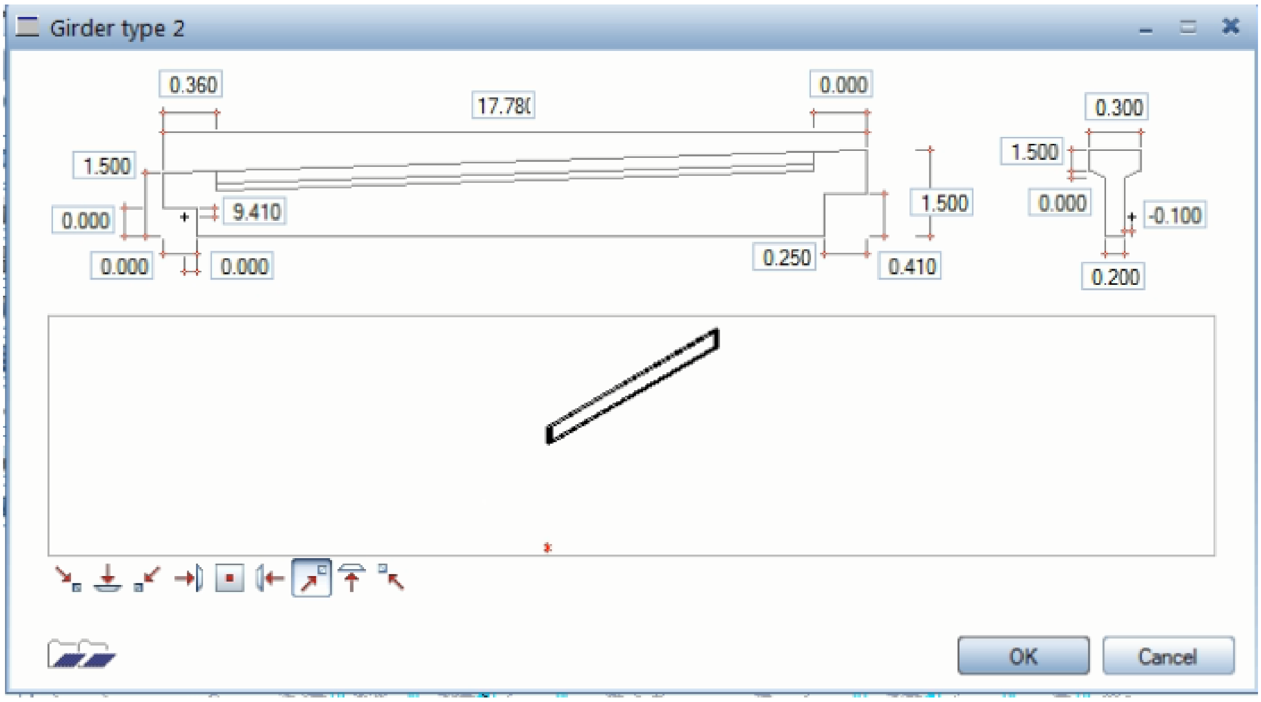


Figure 64 Implemented model: Modeling of girder with basic tools

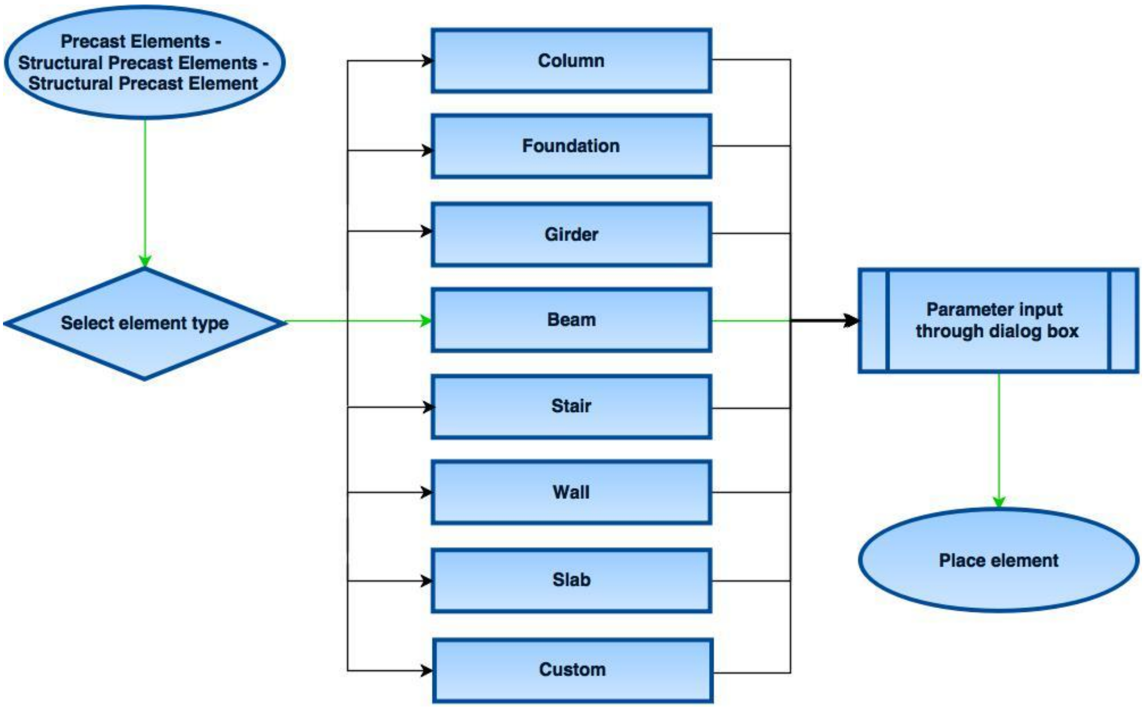


Figure 65. iParts: parameter set

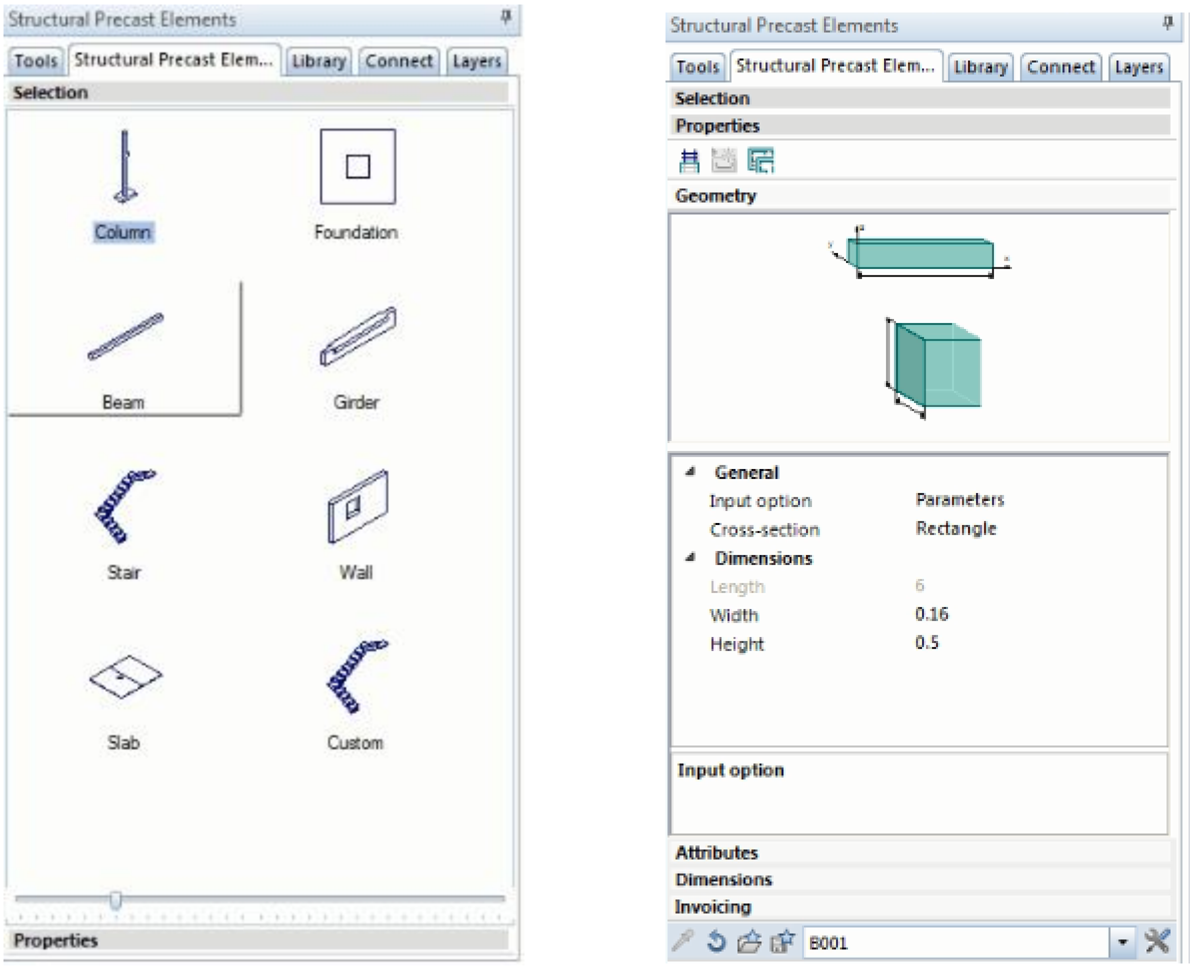


Figure 66. Implemented model: iPart beam. Parameter set

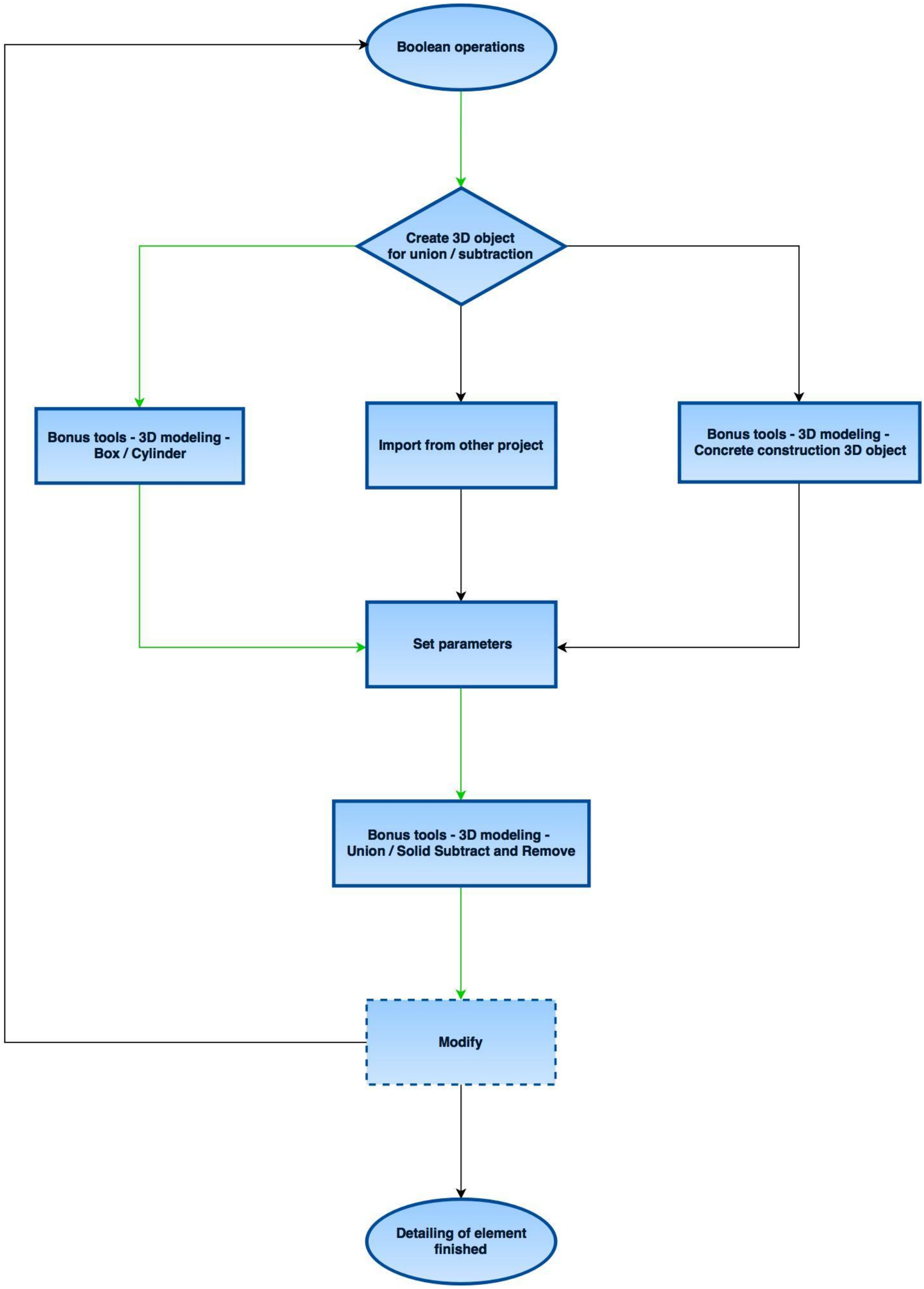


Figure 67. Basic tools: Boolean operations.

Applied for detailing of elements (additional consoles, connections, build-in parts)

Additional detailing of elements

Girders

None of catalog contains finished parametric object. In this case, boolean operations (union, subtraction of solids) are considered to be taken. Similar parametric object is found in catalog using one of basic tools (Concrete construction 3D object). Additional objects are defined. All objects are unified.

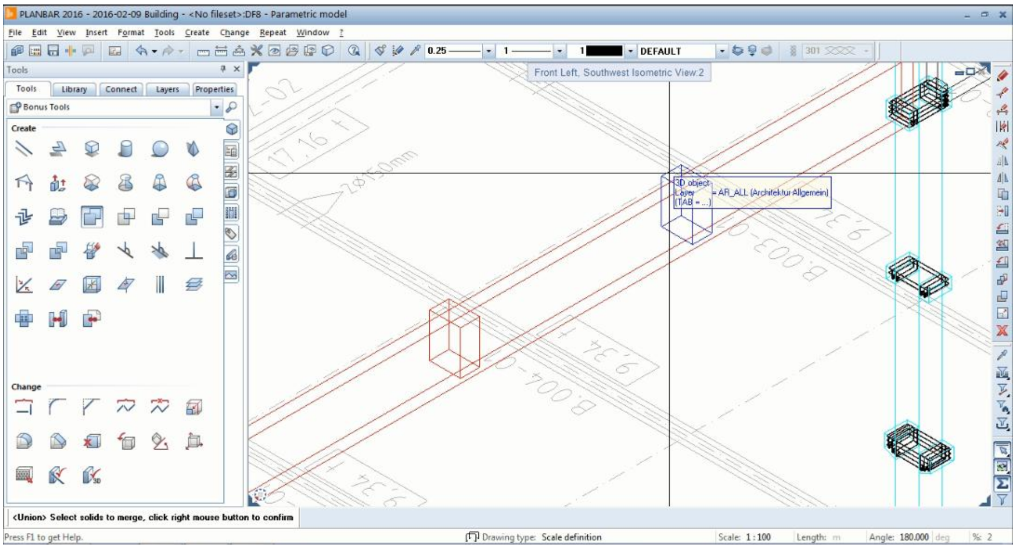


Figure 68. Implemented model. Detailing of girder. Union of 3D objects

Consider elements to be shown on plans

Only basic elements are considered to be shown on plans. Those elements are to be copied. Generally model for approval design is finished at this stage.

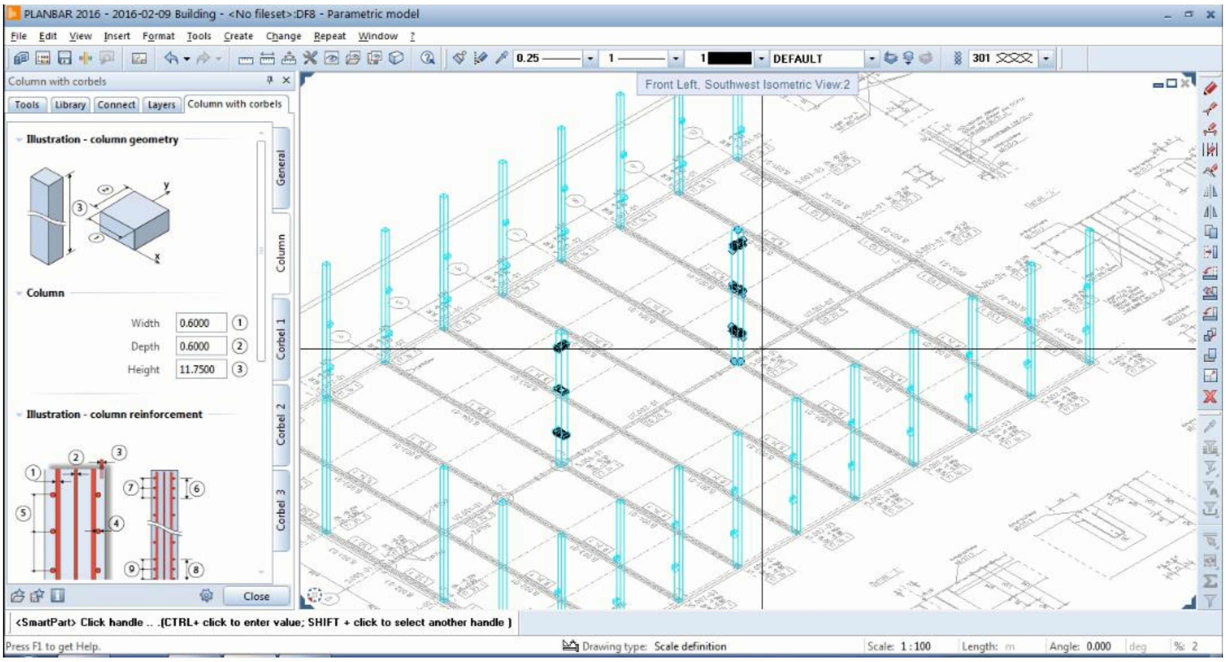


Figure 69. Implemented model: SmartPart column. Placement of elements

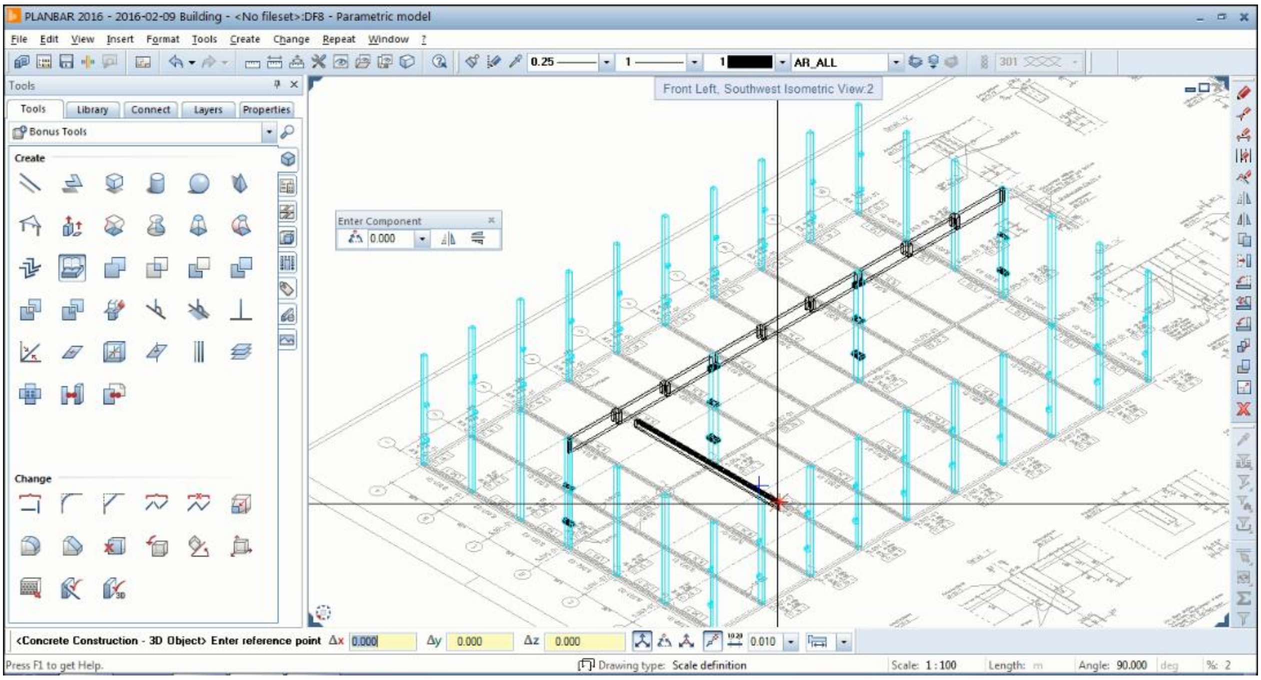


Figure 70 Implemented model: modeling of beam with basic tools. Placement of elements

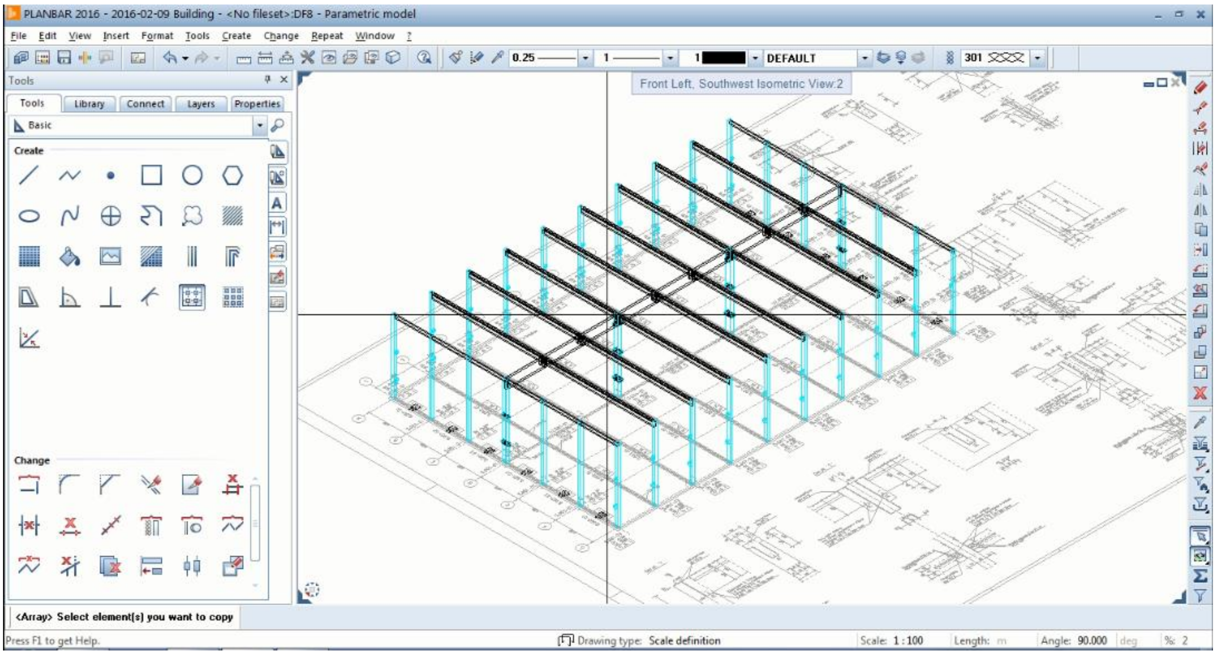


Figure 71 Implemented model: intelligent modeling finished

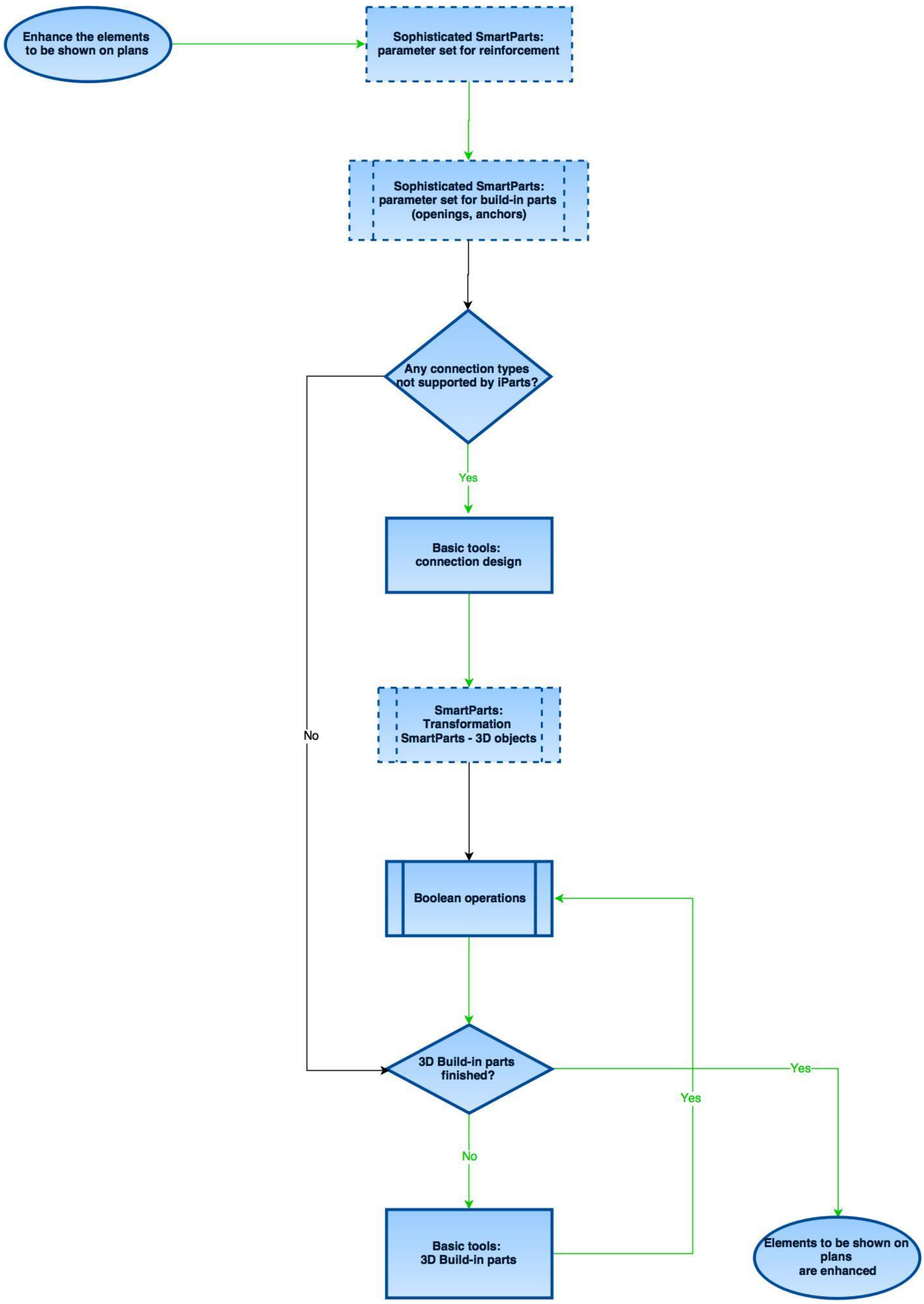


Figure 72. Enhance the elements to be shown on plan

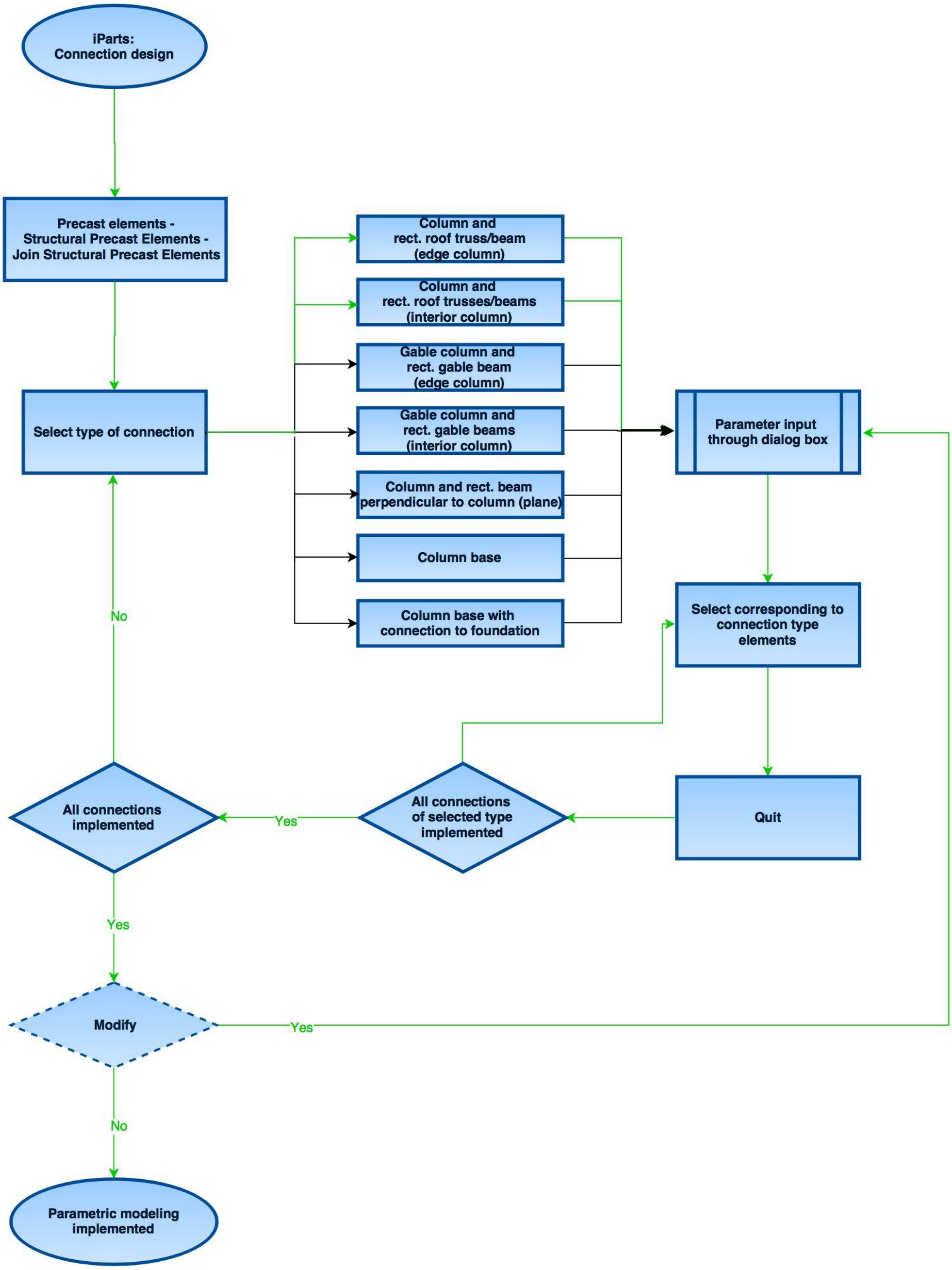


Figure 73. iParts: connection design

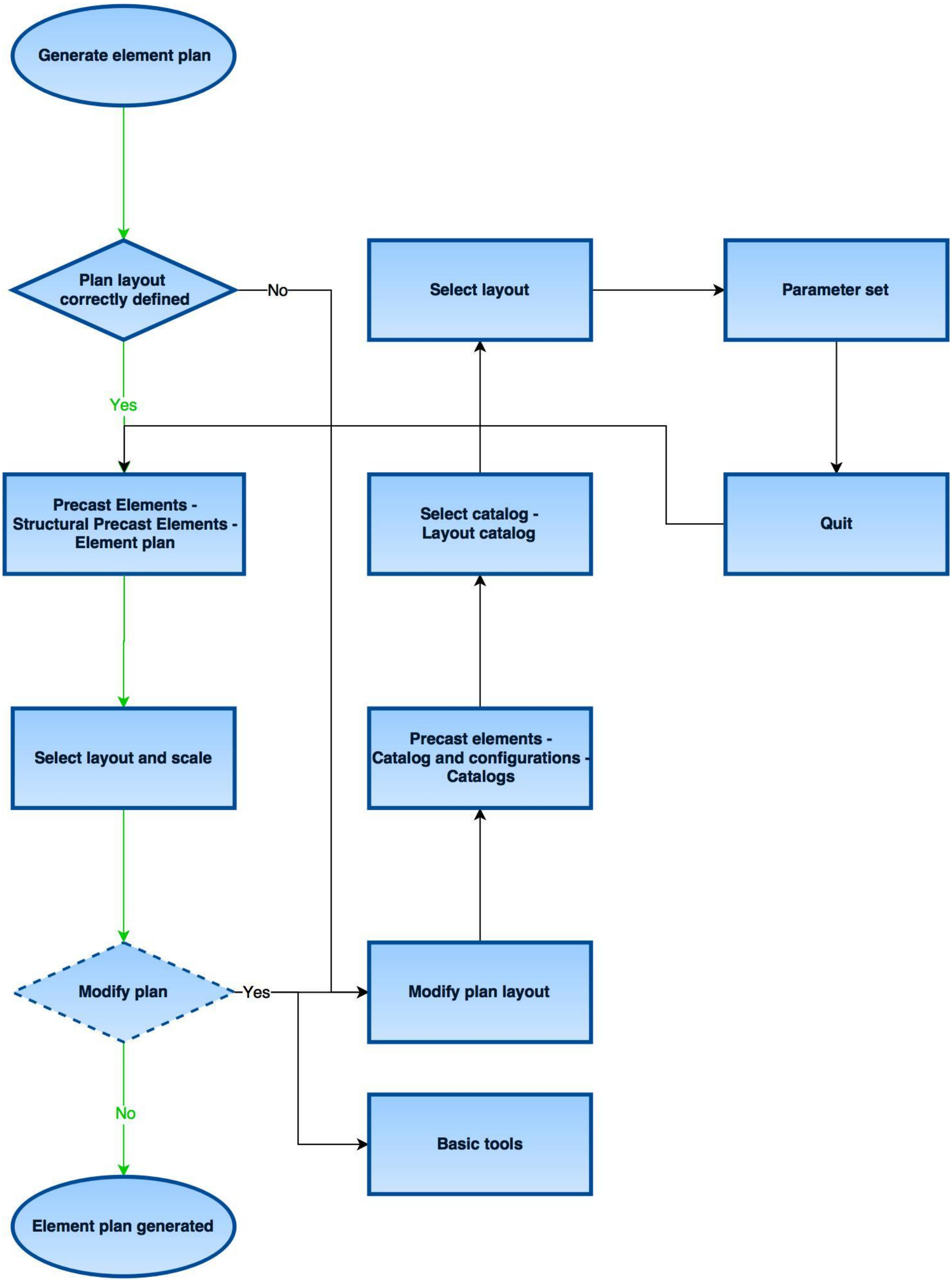


Figure 74. Generation of element plan

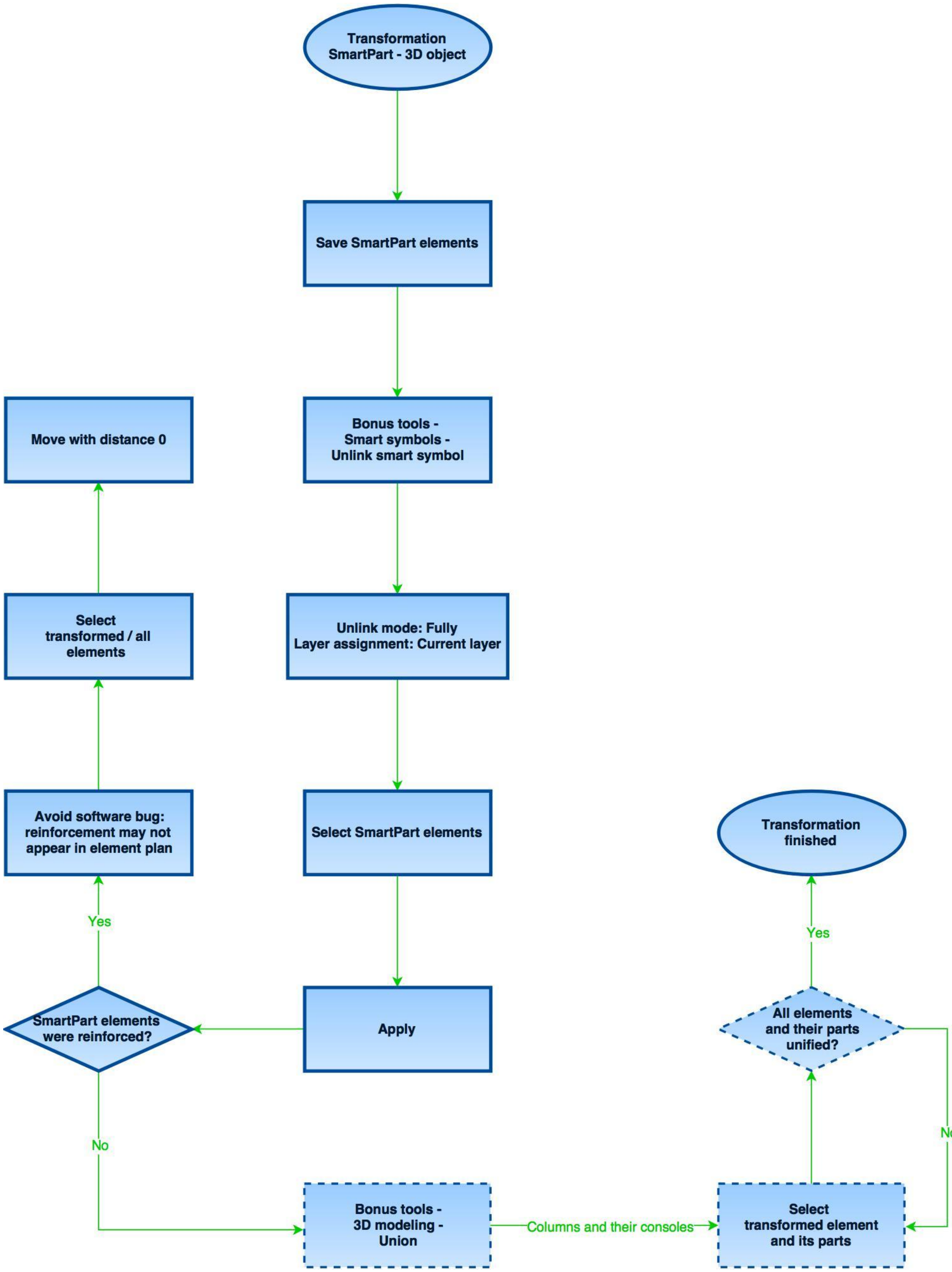


Figure 75. Transformation SmartParts – 3D objects

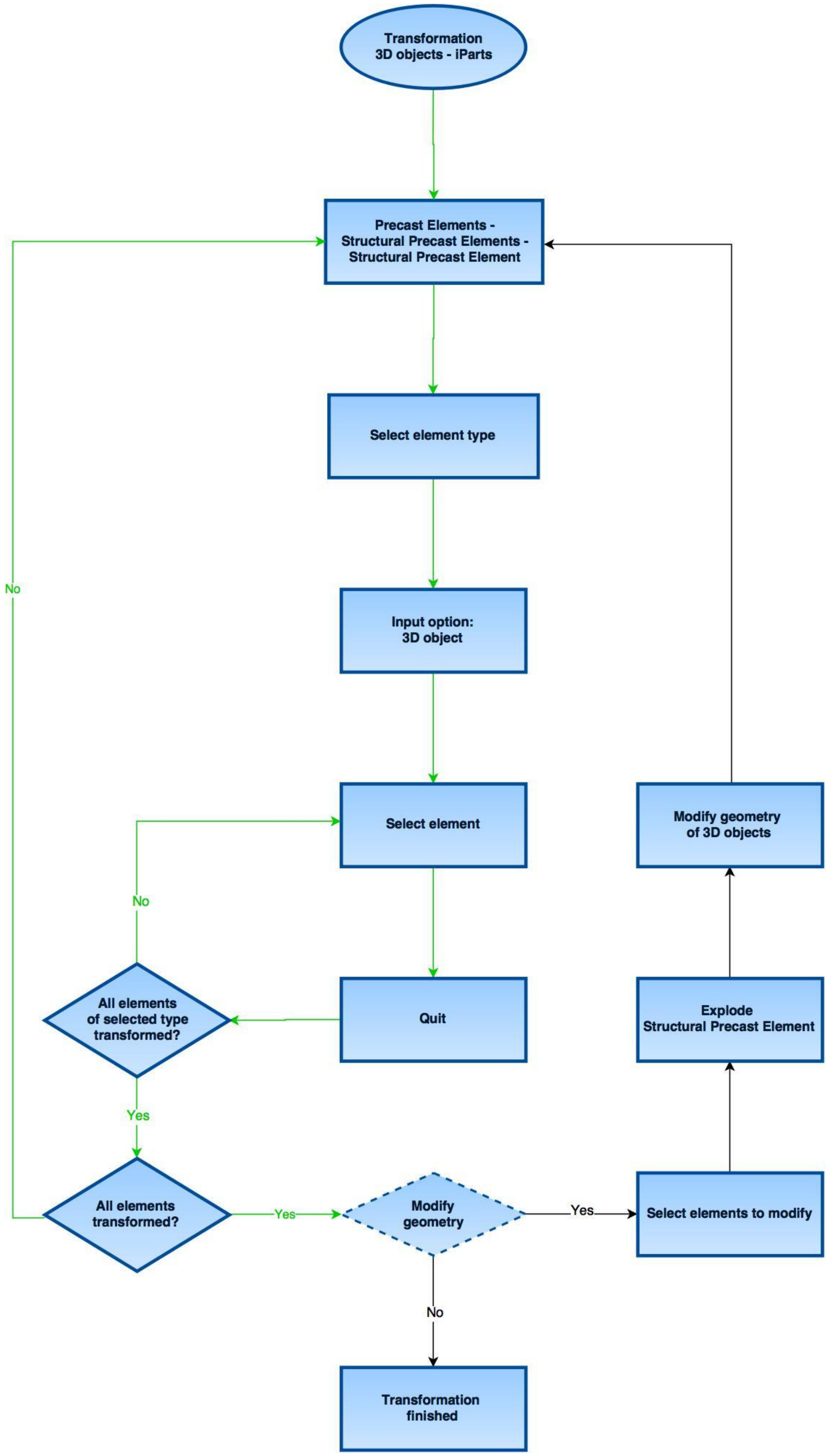


Figure 76. Transformation 3D objects - iParts

7 QUALITATIVE ASSESSMENT OF MODULES SMARTPARTS AND IPARTS. RECOMMENDATIONS FOR USE IN PRACTICE

A parametric 3D model has to fulfill two basic requirements. Once model has been implemented it or its elements can be used in further models (model reusability). Alteration of implemented model has to be performed in an easy way (model flexibility). Goal of modules SmartParts and iParts is to minimize the number of steps in general and number of the error-prone steps while creation, alteration and reuse of a model. Those are the qualities to assess.

Implementation of element plans.

Both modules SmartParts and iParts can replace stepwise section-by-section modeling of plans by generation of complete plans in a few steps. Both SmartParts and iParts provide flexibility of generated element plans. Plans can be always altered using basic design modules. In terms of reusability iParts Element plan tool has strong advantage. Predefined layouts provides highly qualified design automation.

iParts Element plan tool strongly recommended for use in practice. From all available and considered in this work tools only 'Element plan' supports automatic generation of element plans according to predefined layout.

iParts layout catalog has relatively complicated sets of parameters with high amount of them. Use of help files, contact to support or taking a training for software usage is recommended to get along with procedure of parameter set.

Connection design.

Reuse of 3D connections is complicated while they have to be imported and unified. Those are time-consuming operations. Also 3D connection has to be adjusted to each element graphically. This is error-prone. If many elements have to be moved then 3D connections have to be again adjusted to new conditions. iParts connections have to be parametrically set once, and then only elements of corresponding elements (such as column and beam) have to be selected. It replaces graphic set and "copy and paste" operations of connection design in 3D format. Parametric relations between connected elements provide high flexibility. Connection remains if elements are moved (conditional: depends on changes to be applied).

Use of iParts connections is strongly recommended. This is the only option which supports parametric relations between elements which is one of the main ideas of parametric 3D modeling. Alteration of elements in big projects is to be simplified by using iParts connections.

On the other hand, iParts has limited number of supported types of connections. iParts don't support boolean operations. 3D connection design is to be done with 3D objects. Not supported types of connections are to be modeled in advance in 3D format before transformation of elements into iParts format. Otherwise, iParts are to be transformed into 3D, connected and then transformed back into iParts.

Transformations.

iParts is strongly recommended for connection design and implementation of plans. Nevertheless, module implies that elements have to be in iParts format. This means that either elements have to be initially modeled as iParts or they have to be transformed into iParts. Due to poor options of iParts element catalog the second option is more suitable for detailed design. Transformed into iParts elements have lower flexibility and reusability due to lack of options for alteration. If iParts are modeled based on 3-D objects these iParts cannot be altered. Consequently they have to be transformed back into 3D objects, altered and transformed again into iParts which brings additional steps into design process. Furthermore, use of detailed

SmartParts elements implies their one-way transformation into 3D objects with breaking some of SmartParts parametric relations. Also SmartPart element may be splitted into several parts during transformation. Furthermore, software bug may appear after transformation - transformed reinforcement elements may not be shown on plans. Then 3D objects are to be transformed into iParts. These indirect transformations lead to lower flexility and reusability.

It is recommended to choose correct strategy for parametric modeling (Figure Strategies for modeling). The best is to avoid transformations. It is possible if detailing of elements is not required. Also transformation Architectural objects - iParts doesn't remove model of architectural objects which is still flexible and reusable. On the other hand, iParts element catalog and architectural objects are efficient not for detailed design but for approval planning (de Genehmigungsplanung).

On one hand, SmartParts can provide partial automation of element detailing. On the other hand, SmartParts have to be saved before transformation and used in case of alteration. Then they are to be transformed, probably unified, move with distance 0 (in order to avoid software bug). In some cases it may be not possible to determine whether 3D modeling and detailing of elements is faster then modeling of detailed SmartParts elements with further transformation. If SmartParts doesn't bring high benefits in the workflow - if they are not able to simplify significantly the detailed design and they have to be further detailed - then it is recommended to use basic design modules for modeling and detailing of elements.

Modeling of detailed elements

Time-consuming steps in modeling of elements can be avoided if complete element is found as parametric object in element catalog. The biggest catalog of elements is provided by basic design modules. iParts catalog has similar catalog but limited comparing with basic design modules. SmartParts catalog is the worst in terms of availability of elements and the best in terms of design freedom due to use of numerical modeling of elements. SmartParts have potential to model detailed elements directly including reinforcement, build-in parts, if needed connections. On the other hand, SmartParts have no user-friendly options for boolean operations. Reusability of modeled parametric objects doesn't vary much from module to module. Parameter sets always can be stored, loaded, altered or reused. SmartPart elements are the most flexible. Element has variety options for alteration: graphic, alphanumeric, by script. iPart elements can be changed using dialog box. And element of convenient format (3-D objects) have only graphic options for alteration which is error-prone. Strategy for tools selection (see Figure) is recommended for use in practice.

Modeling of reinforcement and build-in parts share the same principle except that iParts don't provide detailing of elements. Moreover, boolean operations which are necessary part of detailing (i.e. additional consoles, openings) are possible either by basic design modules or SmartParts script. Second option involves intensive programming considering that complicated elements can contain more than 50 parameters.

It is recommended that not modeler but programmer implements detailed SmartPart elements by script. SmartParts programming is complicated for user due to commercial interests of vendor for selling finished detailed SmartPart elements and due to consequent lack of tutorials.

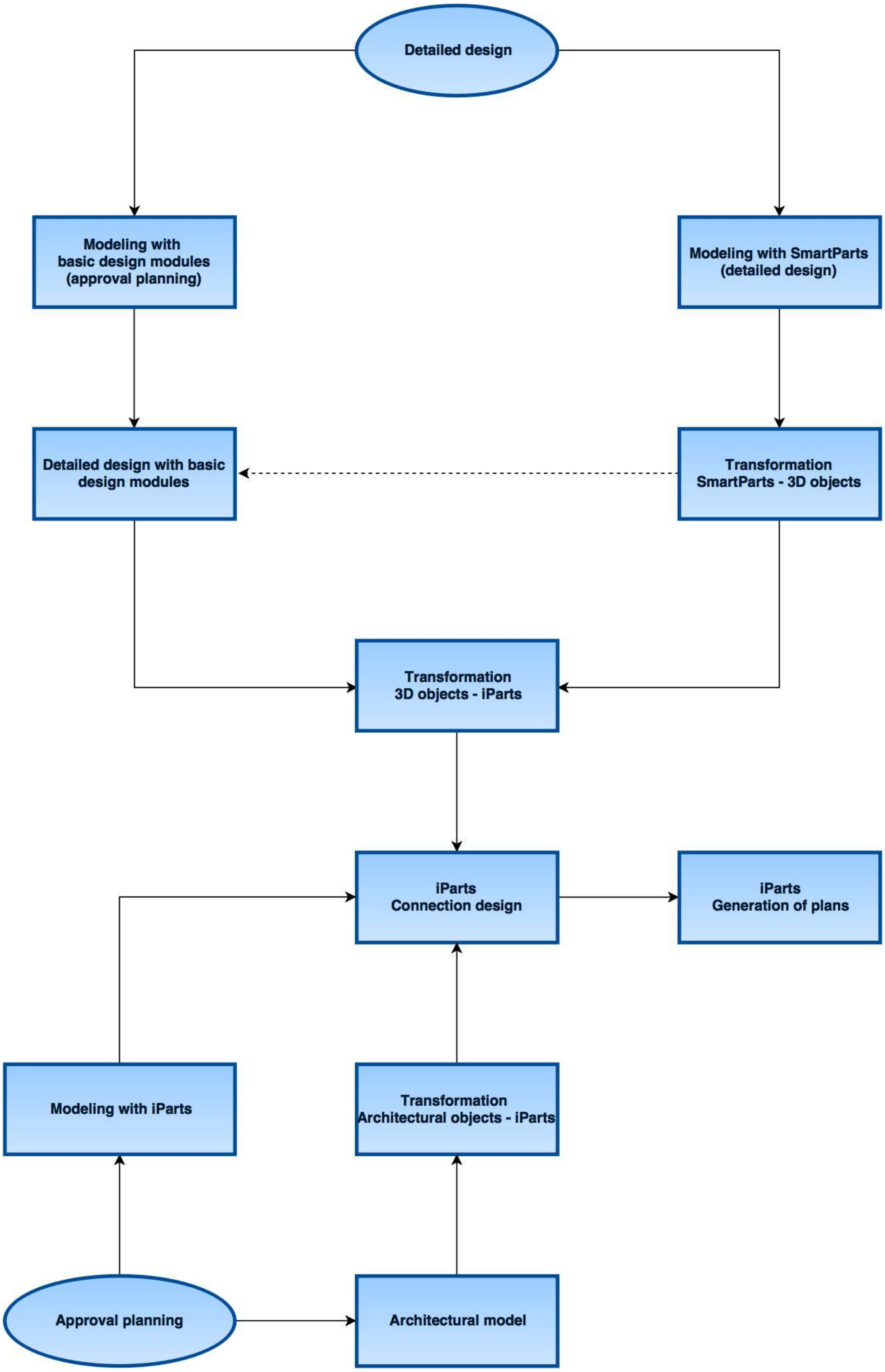


Figure 77. Strategies for usage of SmartParts, iParts and basic design modules.

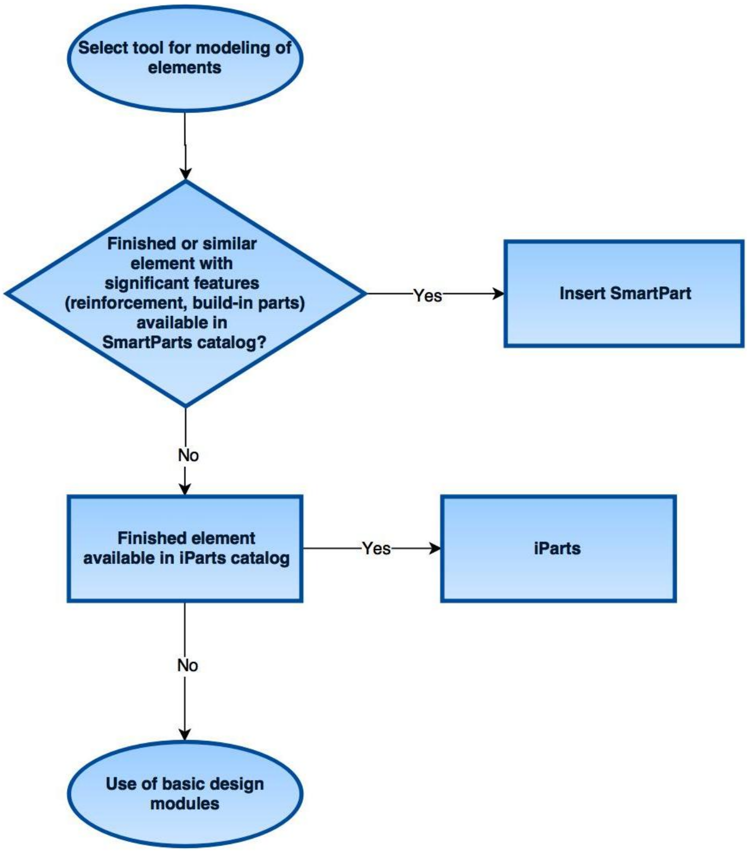


Figure 78 Tools selection for modeling of elements

8 SUMMARY AND OUTLOOK

"Parametric 3-D modeling a building, rather than intelligent modeling, holds the potential to simplify the process of detailed design and to reduce the occurrence of errors and the need for rework in construction projects. Employing parametric 3-D CAD is necessary, but not sufficient: the building model must be developed in an integrated parametric fashion, must be comprehensive, must cover as much of the project scope as possible, and must drive the production of all drawings and reports, if the benefits are to be fully realized. "[11]

Study on two implemented BIM models demonstrates that use of software modules SmartParts and iParts, as expected, is able to increase the overall efficiency in planning of precast concrete elements. Time-consuming and error-prone steps can be replaced through partial automation. Semi-automation of design process results in precast concrete production saving time and money.

Nevertheless, fulfillment of general requirements of the parametric 3D modeling - flexibility and reusability of a model - is conditional. Conditions vary depending on the particular tool of a module to use, stage of modeling to perform and types of elements to be modeled.

SmartParts don't support intelligent modeling. iParts have poor options for parametric design comparing with SmartParts and 3D objects. Transformed into iParts elements have poor modification options. Development of new tools which combine advantages of both modules would increase efficiency:

- modification options and freedom of design by SmartParts (programming, alphanumeric, graphic modification)

- Modelling options (openings, union and subtraction) of 3D objects

- intelligent modelling and automatic generation of plans using iParts

The Allplan Precast currently developes the module which has potential to advance the module SmartParts. Currently applied SmartPart script language is to be replaced by more common Python language.

9 REFERENCES

[1] Messe Bau 2013, Building Information Modeling: Disziplinübergreifende Forschung und Lehre an der TU München, Prof. Dr. Ing. Borrmann, Prof. Dr. Ing. Petzold http://www.allplan.com/en/software/bim/bim-for-engineers.html

[2] Messe Bau 2013, BIM in Tragwerksplanung, Christof Lorenz, Robert Boenisch http://www.allplan.com/en/software/bim/bim-for-engineers.html

[3] ITcon Vol. 13 (2008), Kaner & al. T. Olofsson, G. Lee, C. Eastman, CASE STUDIES OF BIM ADOPTION FOR PRECAST CONCRETE DESIGN BY MID-SIZED STRUCTURAL ENGINEERING FIRMS, 2008, p. 320.

[4] Shah JJ. Assessment of features technology. Computer-Aided Design 1991; 23 (5): 331-43

[5] Camba J.D., Contero M., Company P. Parametric CAD modeling: An analysis of strategies for design reusability. Computer-Aided Design 74 (2016): 18

[6] Automation in Construction 13 (2004), Parametric 3D modeling in building construction with examples from precast concrete, Rafael Sacks, Charles M. Eastman, Ghang Lee, College of Architecture, Georgia Institute of Technology, Atlanta, GA 30332-0155, USA, 12 May 2003, p. 291– 312

[7] http://help.allplan.com/PLANBAR/2016-1/1033/Allplan/index.htm

[8] http://www.fdb-fertigteilbau.de/planungshilfen/musterzeichnungen-fuer-betonfertigteile/

[9] Philipp. Installation and usage instruction.Threaded transport anchors - straight version, 2007: p.3

[10] Presentation Allbau Software: Second International User seminar Allplan Precast, Moscow, 23.09.2015

[11] R. Sacks, C.E. Eastman, G.Lee. PCI journal. Process improvements in precast concrete construction using top-down parametric 3-D modeling, 2003: p. 10

APPENDIX 1.
PATTERN DRAWINGS FOR PRECAST CONCRETE PRODUCTS.
MUSTERZEICHNUNGEN FÜR BETONFERTIGTEILE.

Drawing are provided by organization German Precast Concrete Construction (Fachvereinigung Deutscher Betonfertigteilbau e.V.)

http://www.fdb-fertigteilbau.de/planungshilfen/musterzeichnungen-fuer-betonfertigteile/