# Processing Geometric Models of Assemblies for Functional Structure Extraction

**Ahmad Shahwan - Gilles Foucault - Jean-Claude Léon**

G-SCOP Laboratory

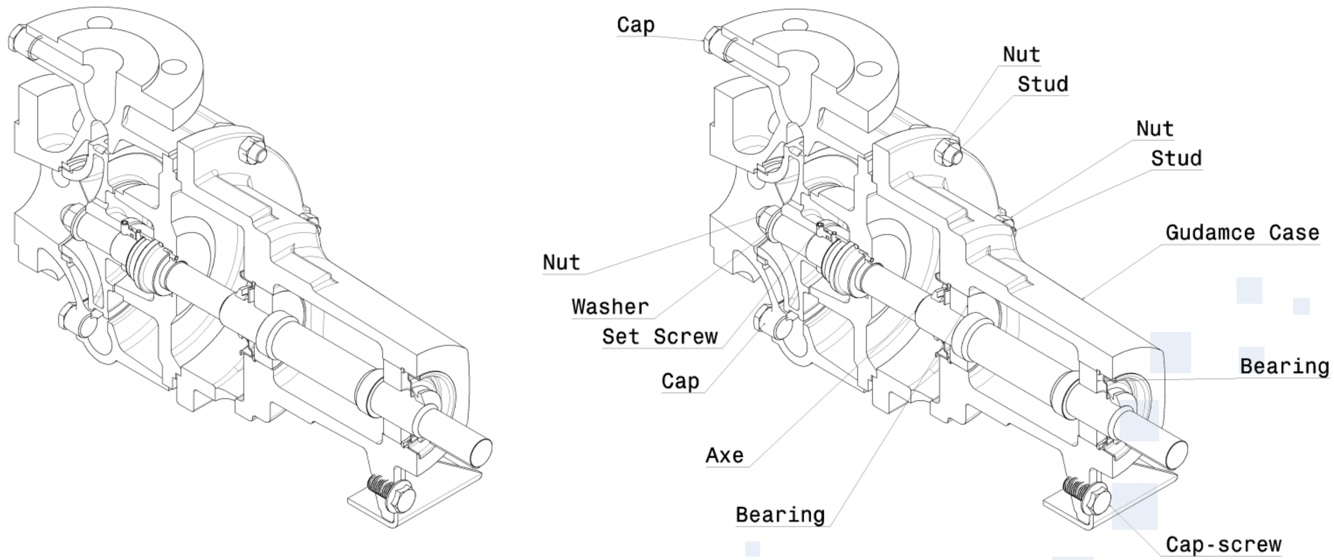
Université de Grenoble, Grenoble-INP

**May 2011**

Laboratoire G-SCOP
46, av Félix Viallet
38031 Grenoble Cedex
www.g-scop.inpg.fr

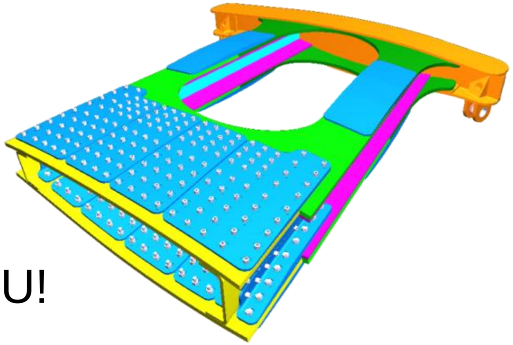
# Introduction

- A PhD thesis in participation to the ANR-funded ROMMA Project (Task 1).
- The ultimate goal is to identify the funtional designations of components within a product  given its geometric model.
- Input: the solid model of a product (its DMU).
- Output: the same model, now annotated with components functional designations.

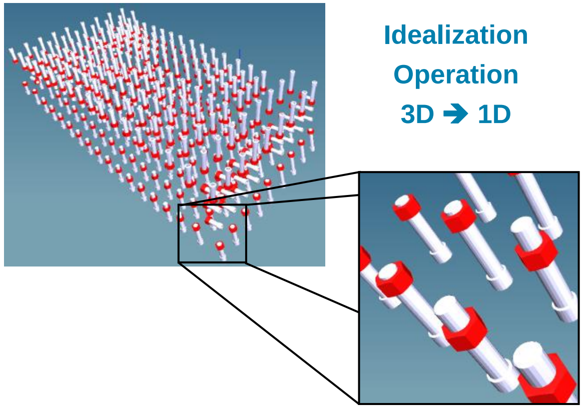


# Motivation

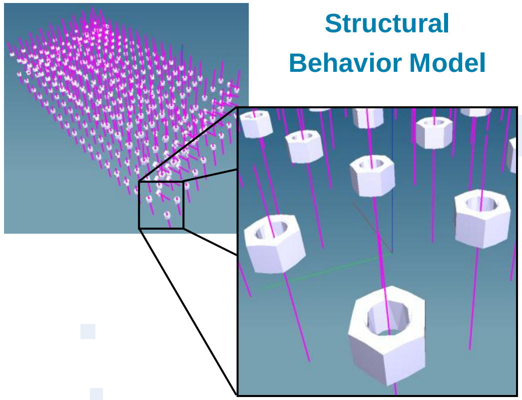
- Importance of DMUs in the product lifecycle.
- Different level of details are needed for different engineering needs. The simplification process.
- Little or no semantic is present in a DMU!
- The knowledge about components designations permits the automation of the simplification process.



**Original Geometric Model**

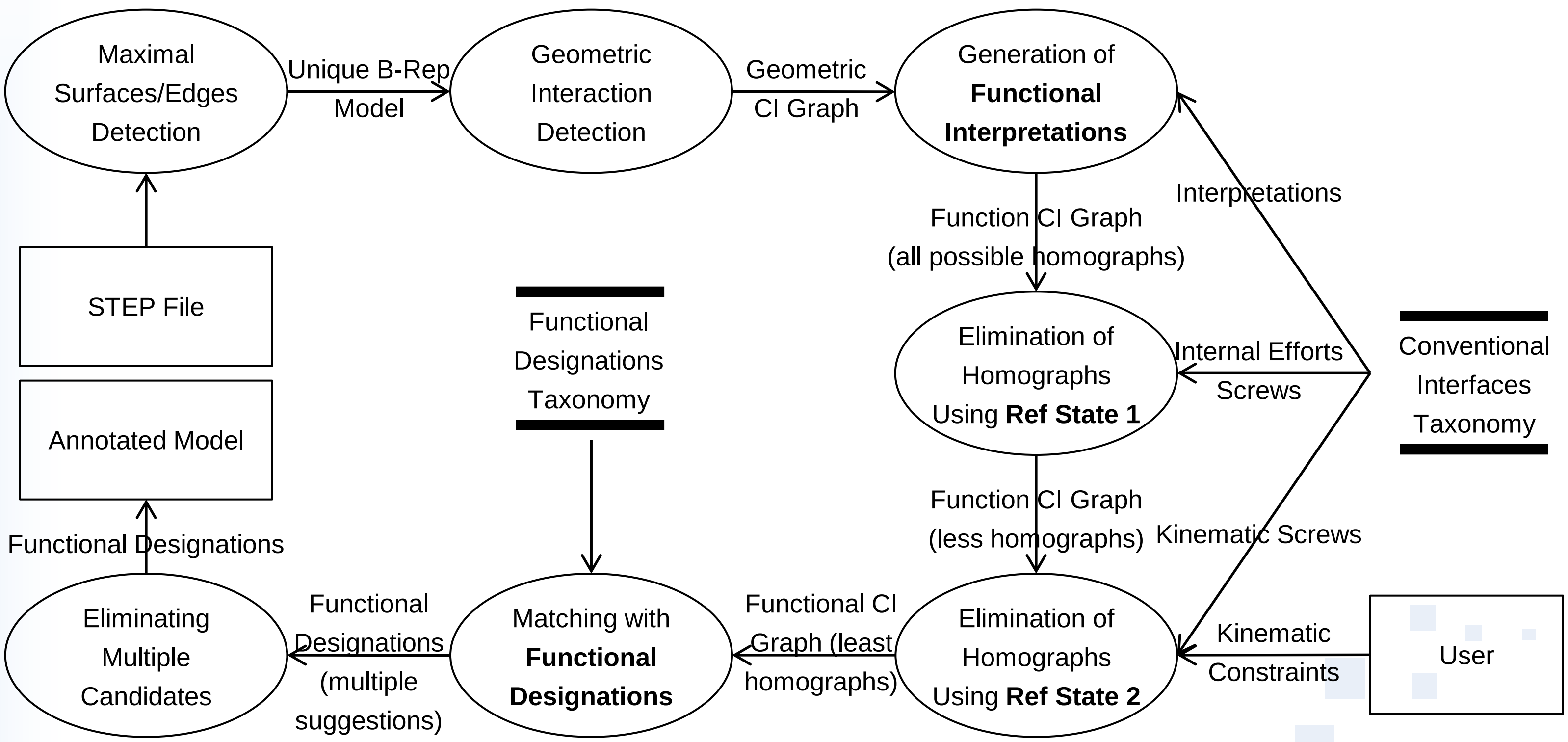


**Idealization Operation 3D → 1D**

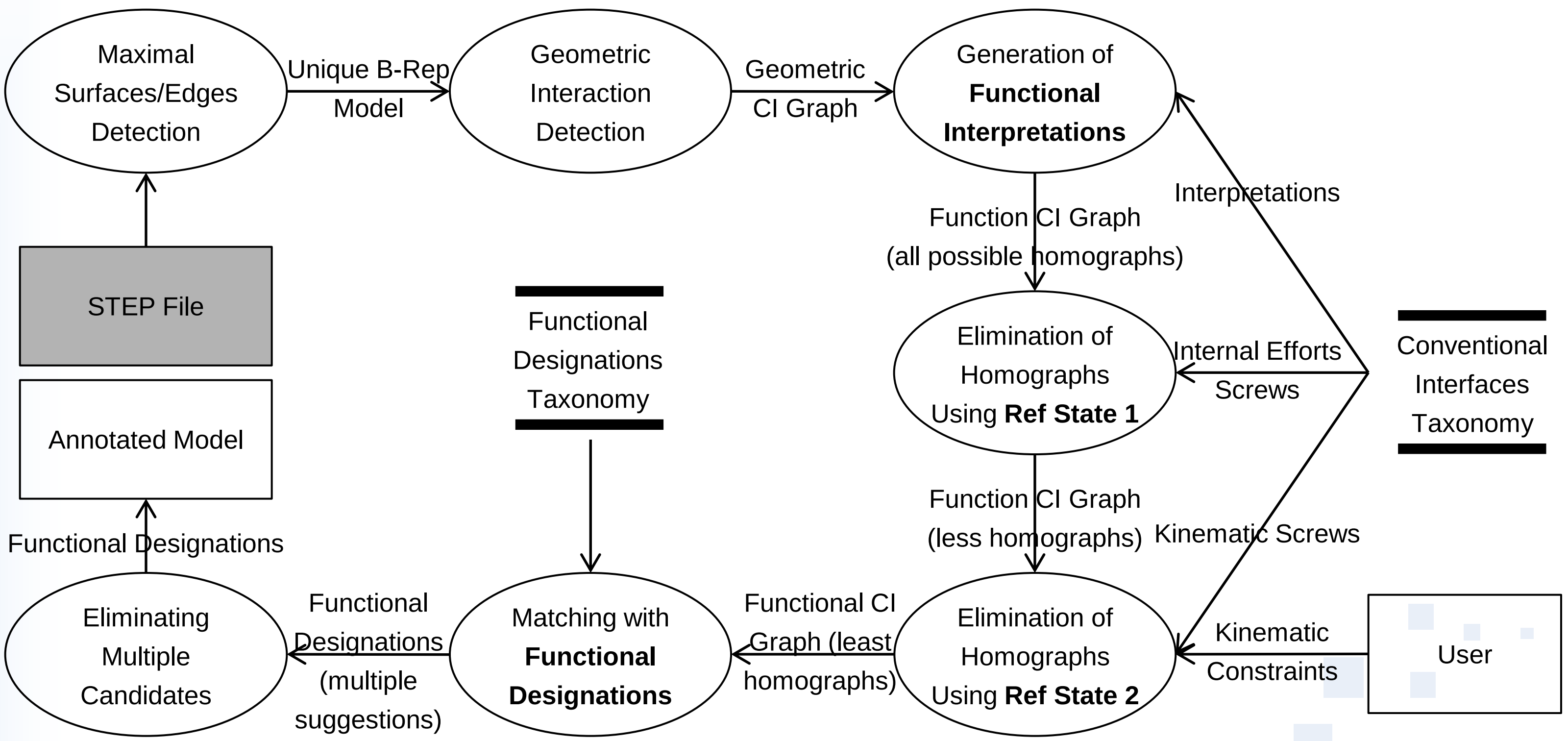


**Structural Behavior Model**

# Data Flow Diagram



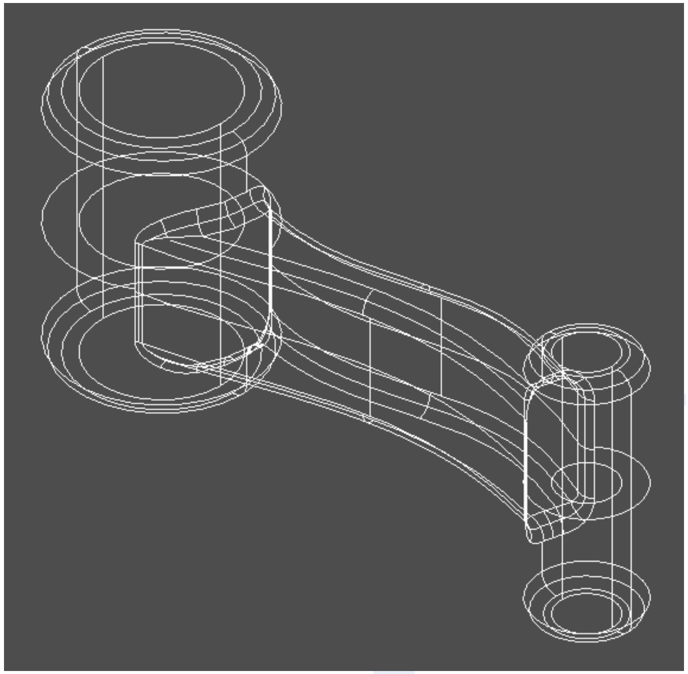
# Data Flow Diagram



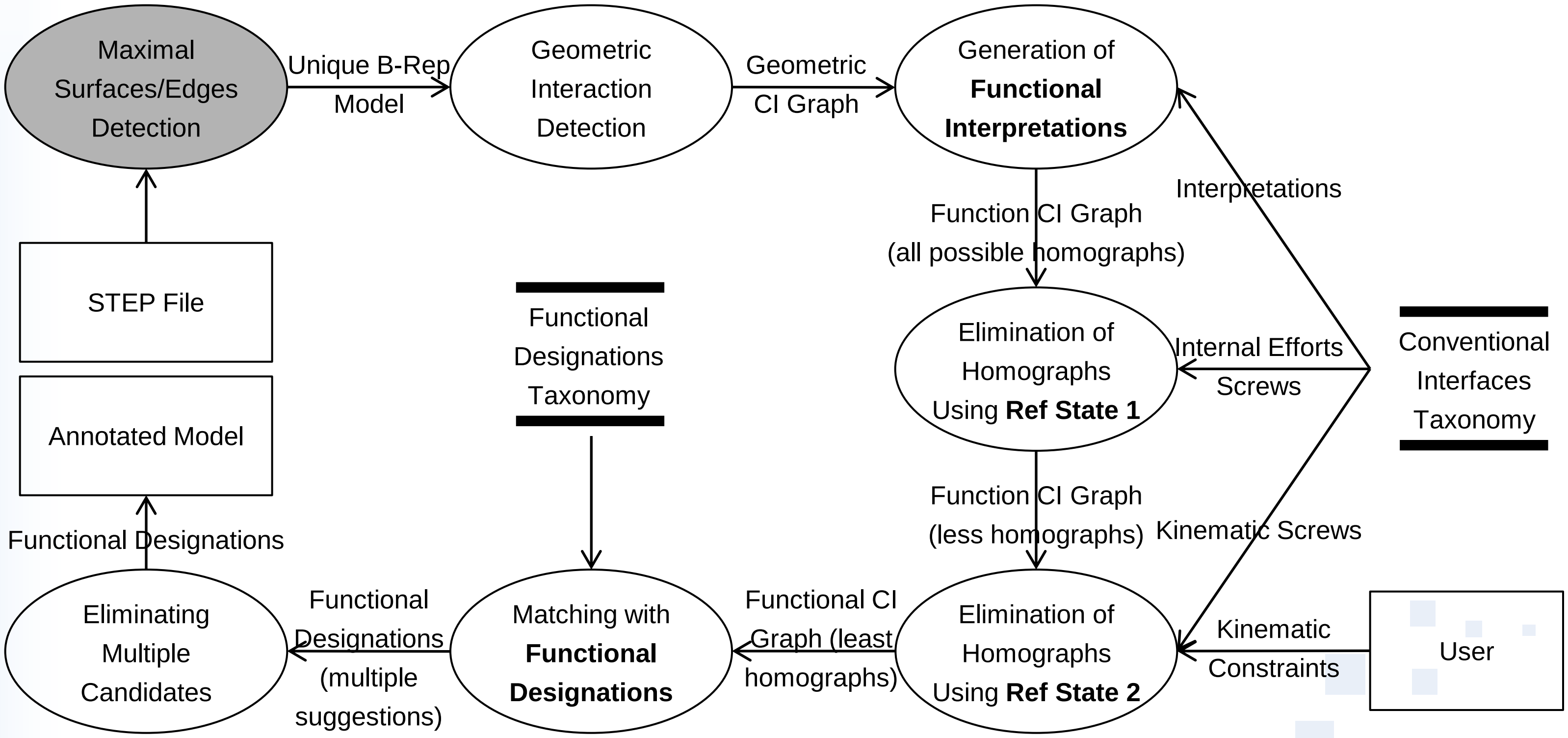
# STEP File

- ISO 10303 Standard.
- AP203 allows the representation of the geometry of the product in a standard textual format using Boundary Representation (B-Rep).

```
ISO-10303-21;
HEADER;
 FILE_DESCRIPTION(('a Product shape'),'1');
 FILE_NAME('Euclid  Shape Model','1998-09-10T11:31:03',('Author Name'),(
    'MATRA-DATAVISION'),'OL-2.0B','EUCLID','Authorisation status');
 FILE_SCHEMA(('AUTOMOTIVE_DESIGN_CC1 { 1 2 10303 214 -1 1 3  2}'));
ENDSEC;
DATA;
#1 = PRODUCT_RELATED_PRODUCT_CATEGORY('Undefined Category','Undefined De
scription',(#2));
#2 = PRODUCT('the product name','the product name','void',(#3));
#3 = MECHANICAL_CONTEXT('Mechanical',#4,'Assembly');
#4 = APPLICATION_CONTEXT('EUCLID');
#5 = APPLICATION_PROTOCOL_DEFINITION('CommitteeDraft','automotive_design
',1997,#4);
#6 = SHAPE_DEFINITION_REPRESENTATION(#7,#11);
#7 = PRODUCT_DEFINITION_SHAPE('void','void',#8);
#8 = PRODUCT_DEFINITION('void','void',#9,#10);
#9 = PRODUCT_DEFINITION_FORMATION('ID','void',#2);
#10 = PRODUCT_DEFINITION_CONTEXT('as proposed',#4,'First_Design');
#11 = ADVANCED_BREP_SHAPE_REPRESENTATION('',(#12),#18620);
#12 = MANIFOLD_SOLID_BREP('',#13);
#13 = CLOSED_SHELL('',(#14,#291,#3567,#3629,#3762,#3869,#4146,#7477,
    #7539,#7672,#7779,#7807,#7835,#7998,#10155,#12312,#12461,#12610,
    #14726,#16765,#16844,#17057,#17221,#17365,#17502,#17591,#17752,
    #17847,#18042,#18071,#18165,#18278,#18327,#18393,#18540,#18606,
    #18613));
#14 = ADVANCED_FACE('',(#15),#30,.T.);
#15 = FACE_BOUND('',#16,.T.);
```



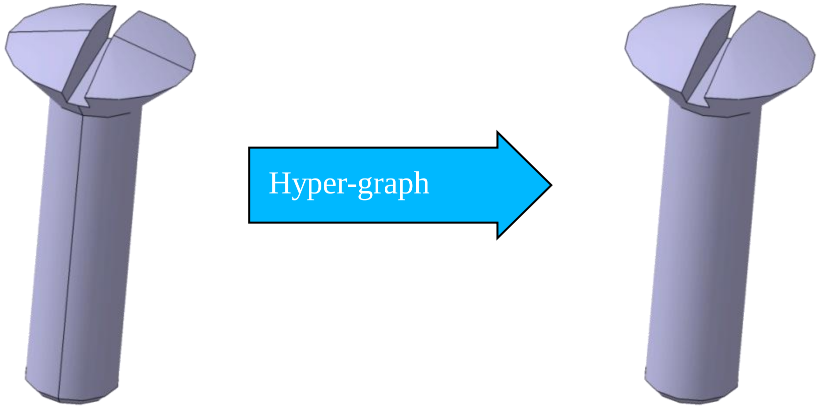
# Data Flow Diagram



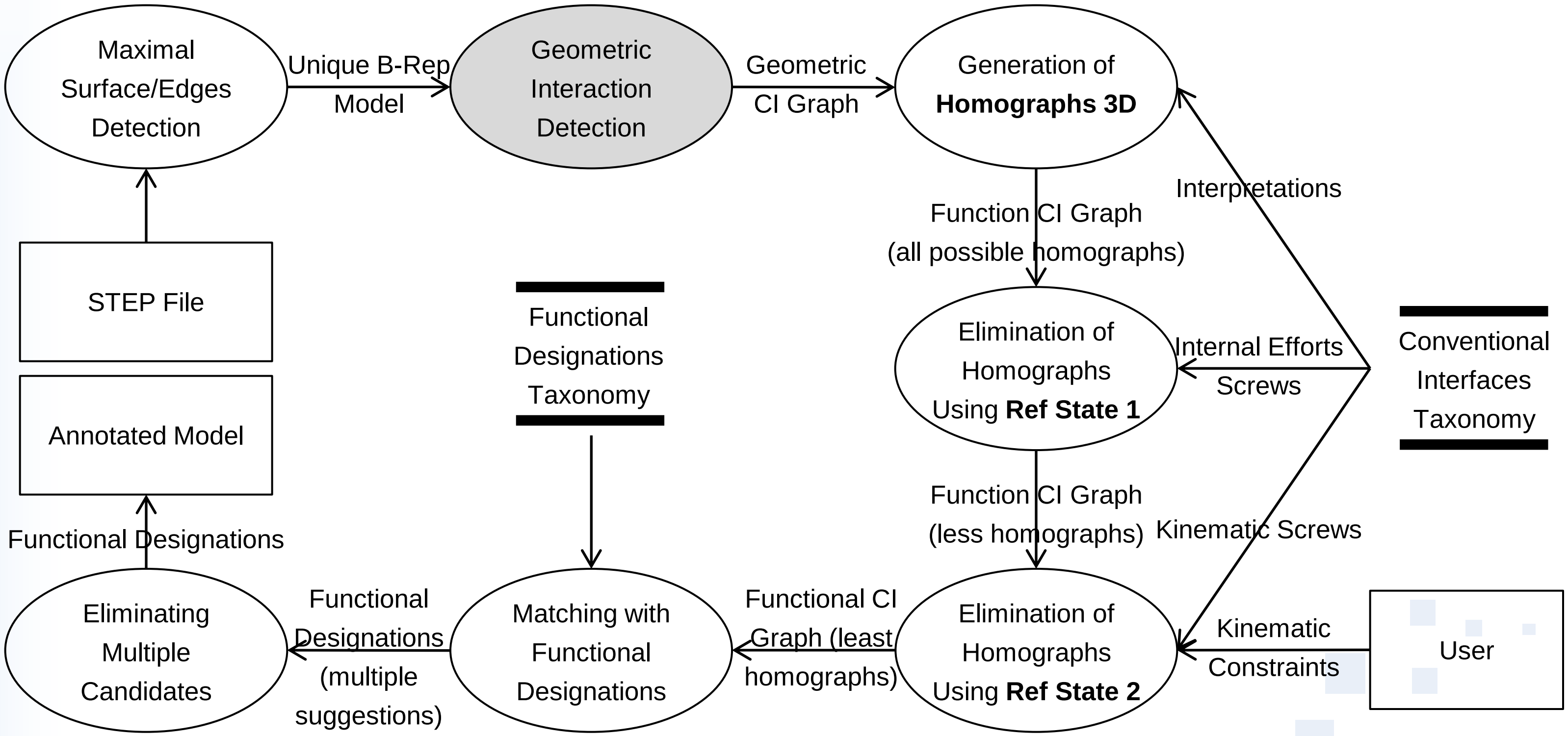
# Maximal Surfaces/Edges Detection

• Using B-Rep the same object might be represented differently, as the same face might be divided into more than one without changing the geometry. Same for the edges.

• To guarantee a unique representation and the independence of the modeling process, we first convert the B-Rep model into maximal surfaces and edges, which represent the product intrinsically.

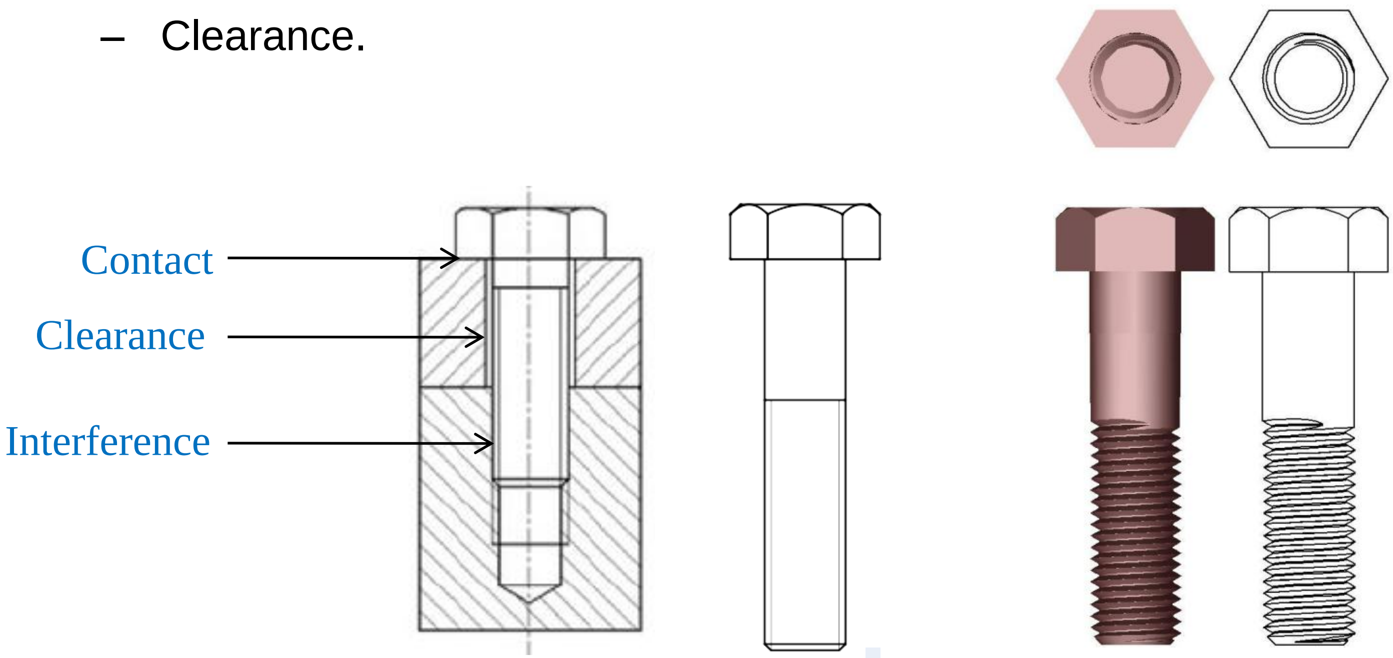


# Data Flow Diagram



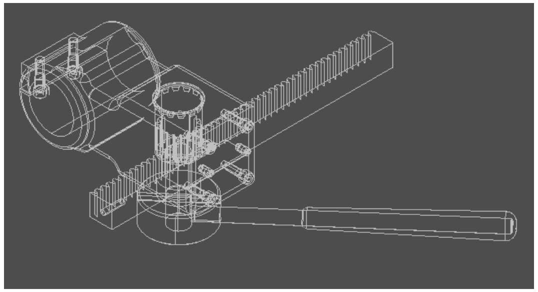
# Conventional Interfaces

- The result of the geometric interaction between neibouring components.
- Can be one of the following:
  - Interference;
  - Contact; and
  - Clearance.

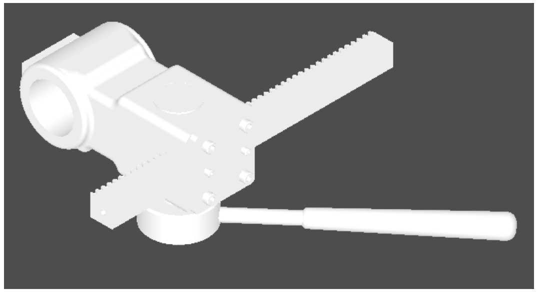


# Geometric Analysis

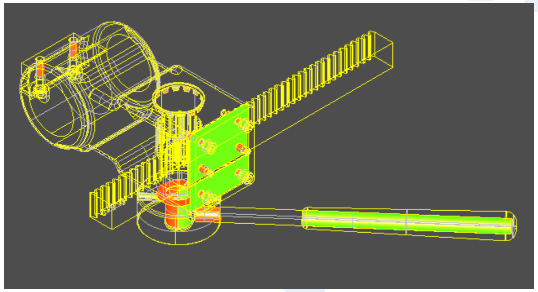
1. Input: the product’s DMU as a STEP file.



2. The product’s DMU as maximal surfaces/edges B-REP.

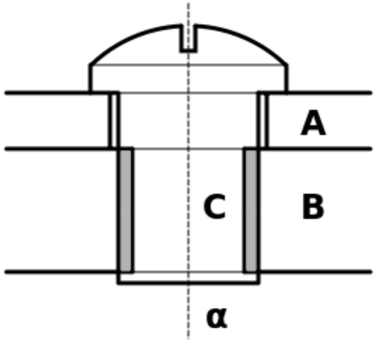


3. Identify CI: detect interference, contact, or clearance zones.

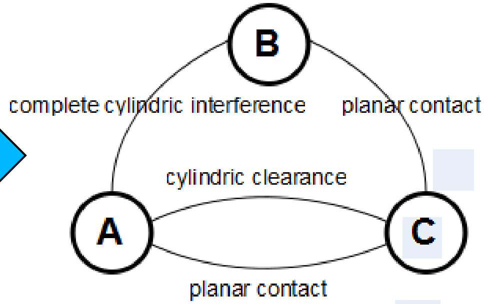


# Geometric Analysis

- The model is then transformed into a Geometric Conventional Interface Graph, where:
  - Nodes are components;
  - Edges are geometric interaction.

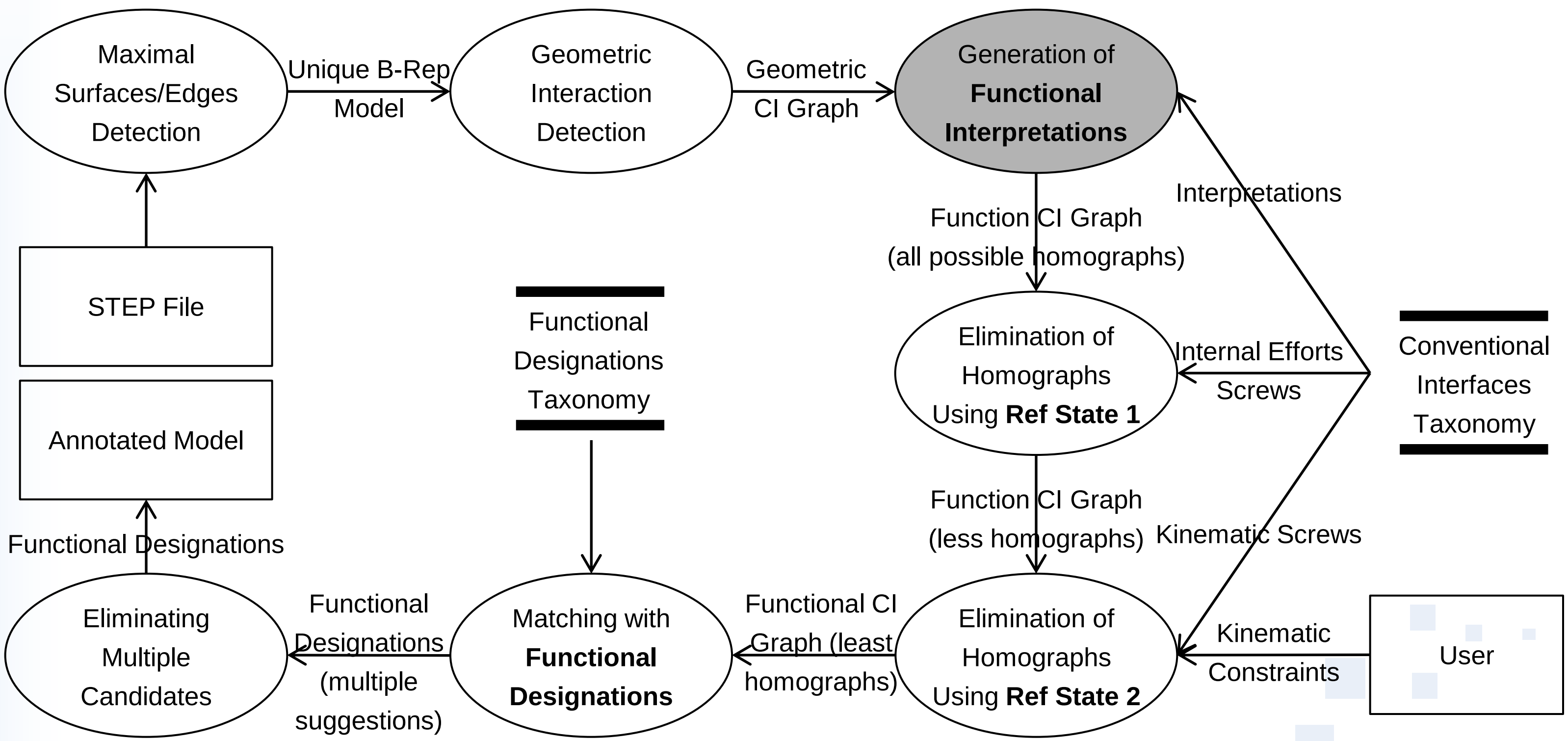


Solid Model



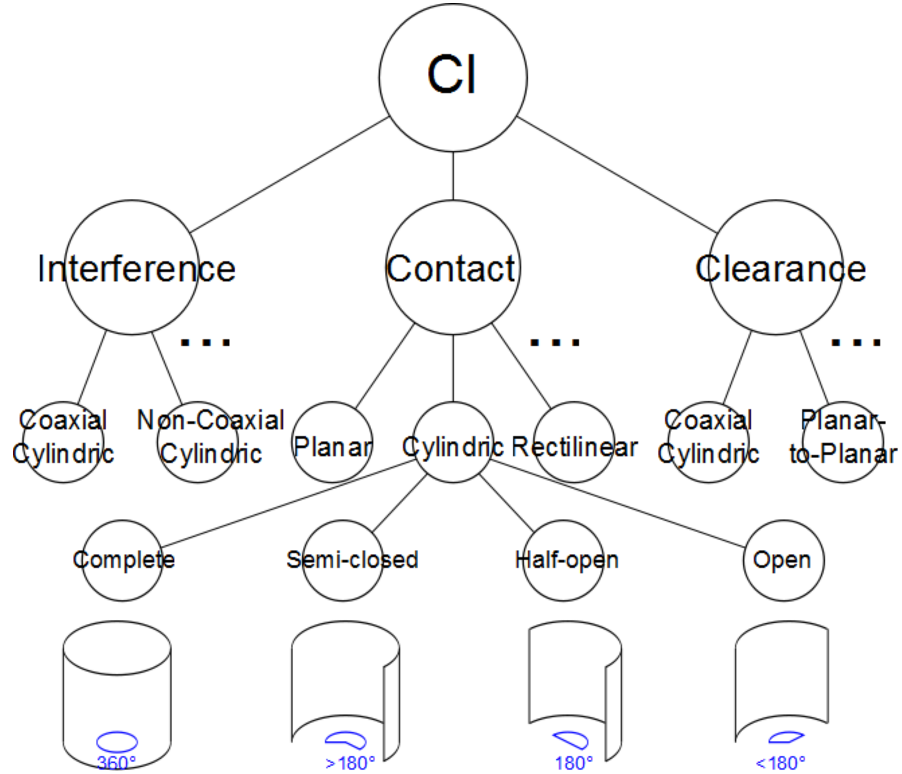
Geometric CI Graph

# Data Flow Diagram



# The CI Taxonomy

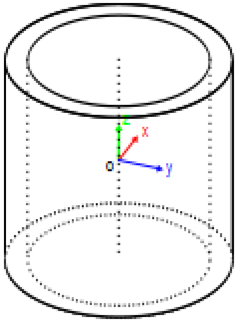
- A hierarchy of all possible geometric configurations.
  - Each leaf in the hierarchy defines unambiguously a particular geometrical configuration called morpheme.



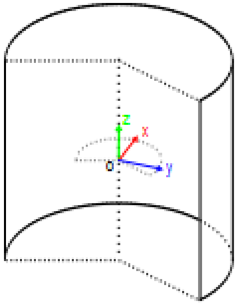
# The IC Taxonomy

- **Morpheme**: is the smallest meaningful geometric configuration. Those are the leaves of the hierarchy. Morphemes are associated with local coordinate systems.
  - Examples:

Coaxial Cylindric Interference




Semi-closed Cylindric Contact



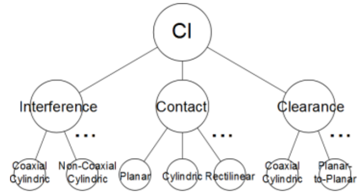

- **Lexeme**: is a morpheme possibly in a special context.

# The IC Taxonomy

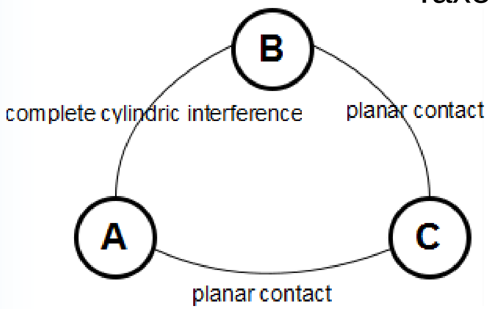
- The Thesaurus links lexemes (Geometric Configurations) with their interpretations (Functional Interfaces).
- Examples:
  - Threaded Link
    - Complete Cylindric Contact;
    - Coaxial Cylindric Interference linking a connected component.
  - Cap Link
    - Coaxial Cylindric Interference linking an (otherwise) isolated component.
- Each Functional Interface defines a mechanical and a kinematic screw.

# Generation of Functional Interfaces

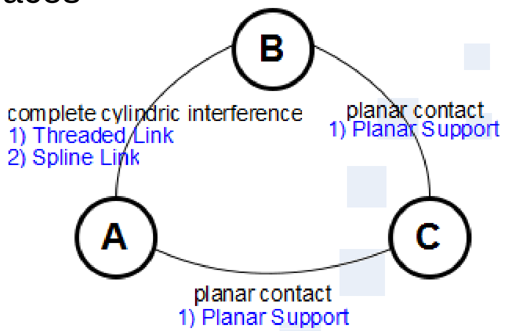
- In the Geometric Conventional Interface Graph we map each edge to its lexeme in the taxonomy and replace it with all related homographs, generating the **Functional Interface Graph**.




Taxonomy of Conventional Interfaces

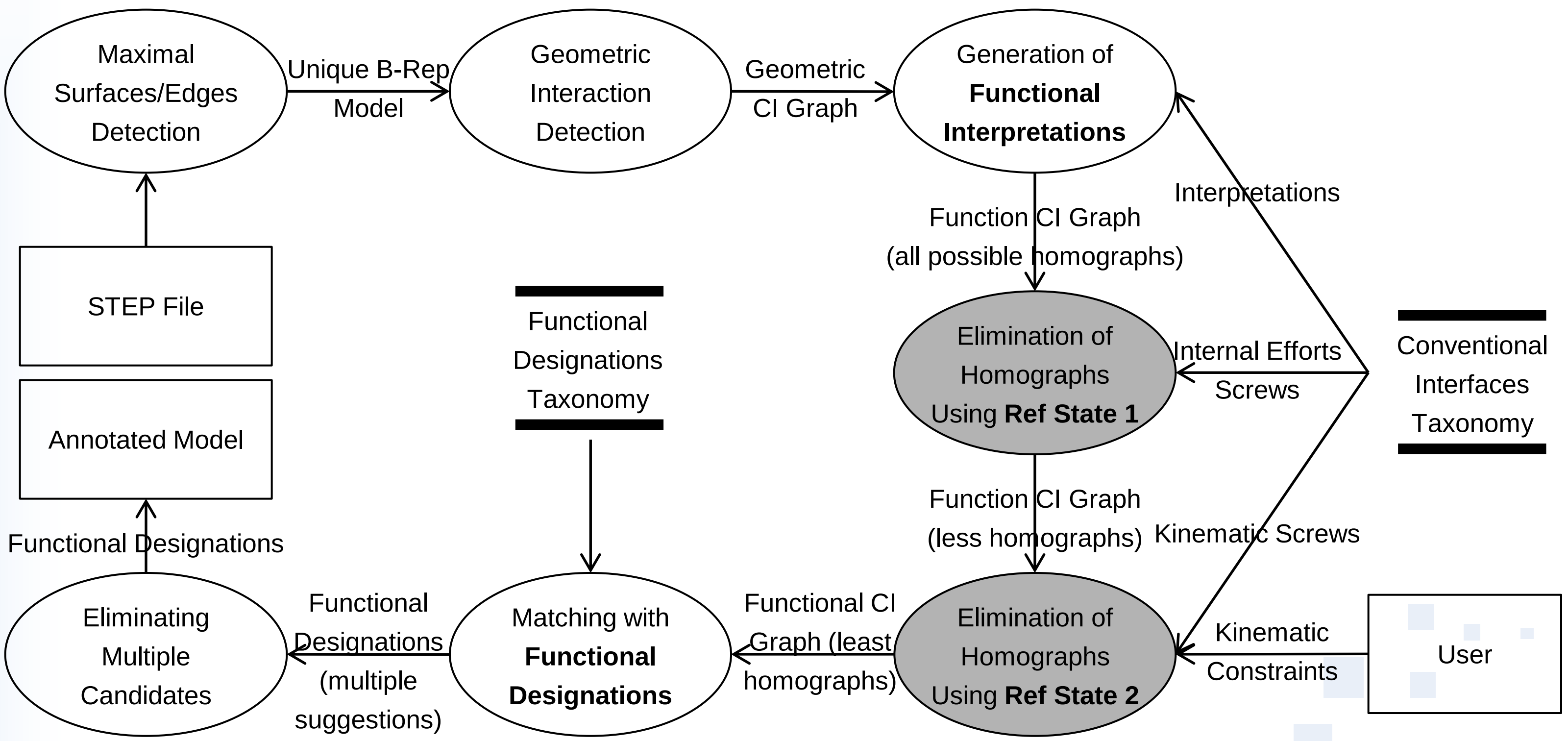



Geometric Conventional Interface Graph




Functional Conventional Interface Graph

# Data Flow Diagram



# Reference States and Dualities

- Two reference states:
  1. The product is mechanically isolated; no external forces. The interactions between components is characterized by internal forces.
  2. The product is kinematically operational; user's input of  a kinematic constraints.
- Relate to two dualities, respectively:
  1. Geometry/Force duality.
  2. Geometry/Mobility duality.

# Mechanical & Functional Analysis

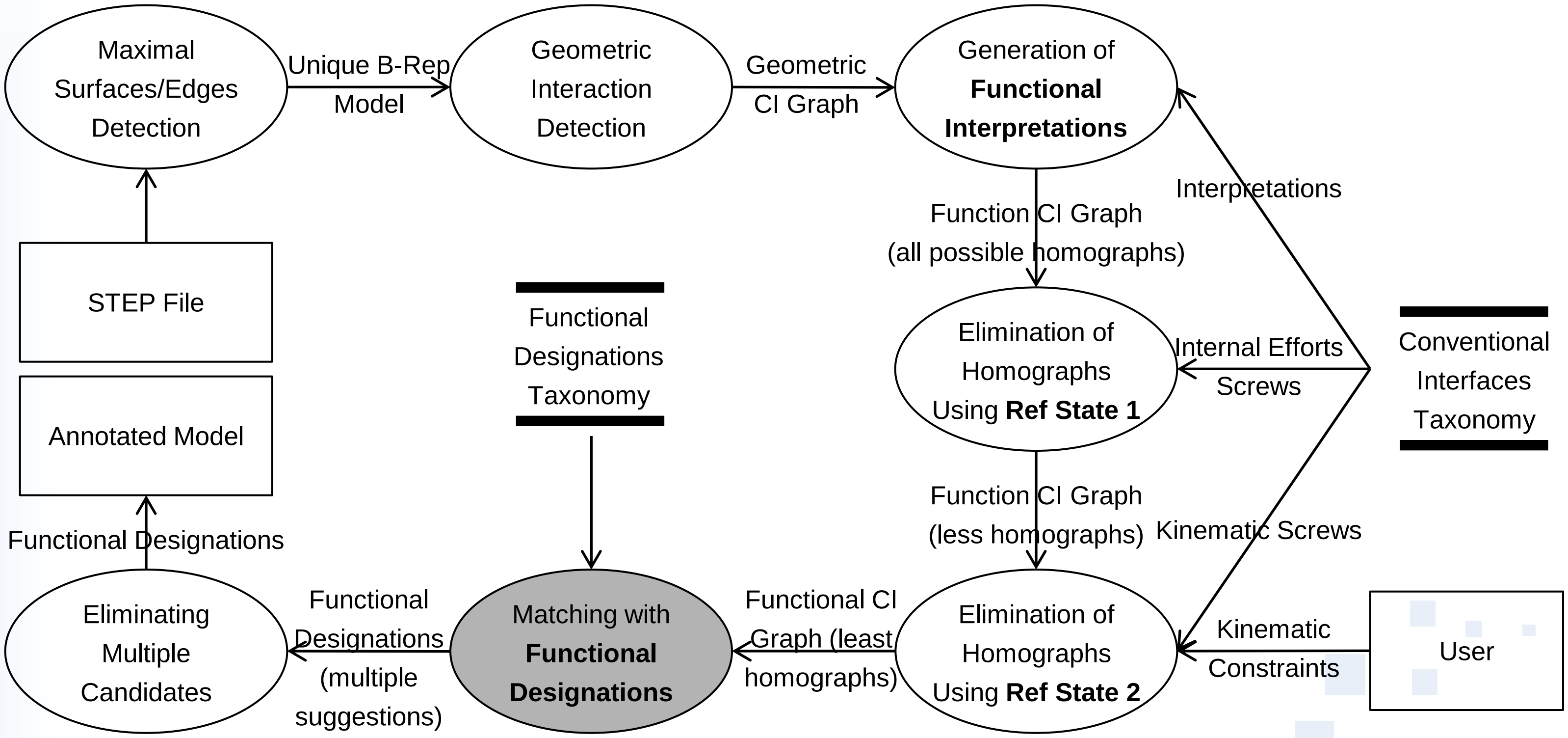
- Each interpretation is characterized by two structures defining constraints on each of internal efforts screw and kinematic screw.
- Constraints are: null (0), strictly positive (+), strictly negative (-), non-null (±), or arbitrary (x).

| Functional Interface | Internal Efforts | Kinematic |
|---|---|---|
| Threaded Linkage | $\begin{Bmatrix} \pm & 0 \\ x & x \\ x & x \end{Bmatrix}$ | $\begin{Bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}$ |
| Spline Link | $\begin{Bmatrix} 0 & 0 \\ x & x \\ x & x \end{Bmatrix}$ | $\begin{Bmatrix} x & x \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}$ |
| Planar Support | $\begin{Bmatrix} + & 0 \\ 0 & x \\ 0 & x \end{Bmatrix}$ | $\begin{Bmatrix} 0 & x \\ x & 0 \\ x & 0 \end{Bmatrix}$ |

# Mechanical & Functional Analysis

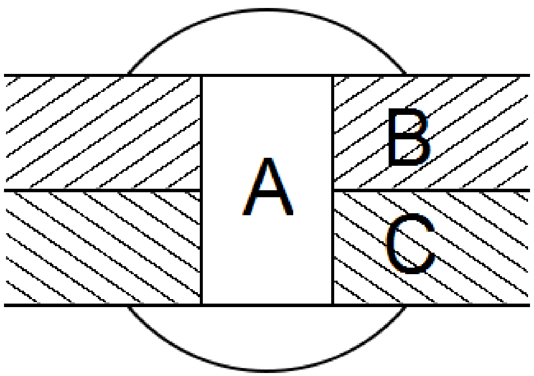
- The algorithm:
  - *Apply CI Taxonomy suggestion to the Geometric CI graph, replacing each edge (lexeme) by all its possible interpretations (homographs).*
  - *Initially, all nodes are open.*
  - *Until (all nodes are closed)*
    - *Among open nodes, choose a node with smallest number of valid assumptions.*
    - *Using Ref. State (1 or 2), eliminate interpretations that lead to inconsistency.*
    - *Recalculate number of valid assumptions.*
    - *If (number of valid assumptions is one, or no interpretation was eliminated)*
      - *Mark node as closed (not open).*
    - *If (no more valid interpretation)*
      - *Report inconsistent model.*
- Where
  - Our Graph is a pseudo-graph, i.e. parallel edges are allowed.
  - Number of valid assumptions is the product of number of remaining interpretations over all the interfaces involving a node.

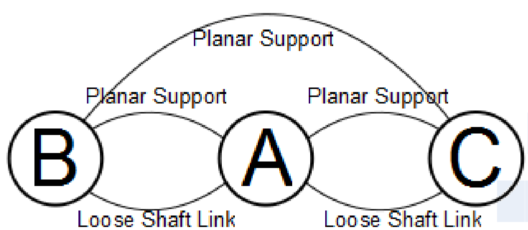
# Data Flow Diagram



# Matching with Functional Designations

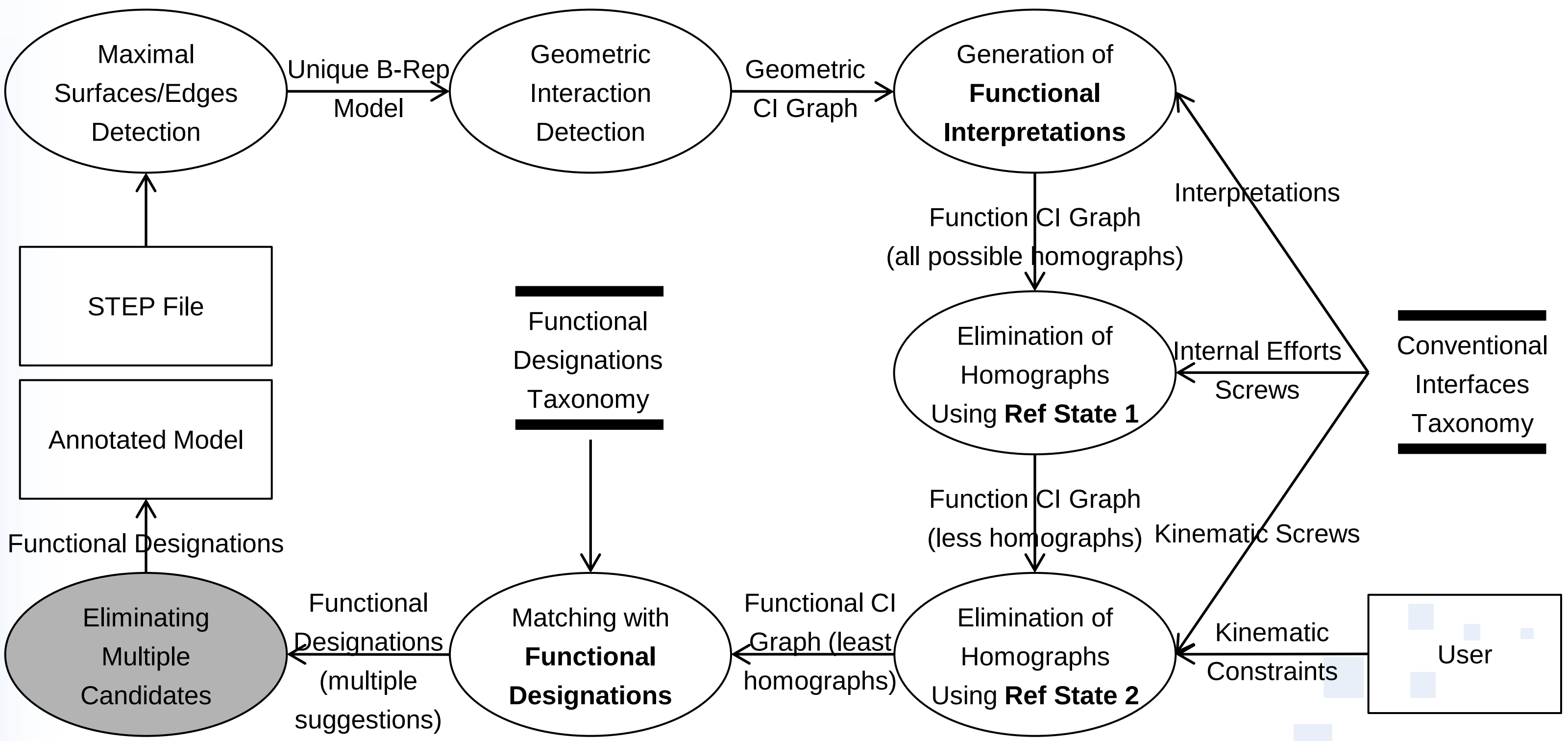
- Functional interfaces of a component will decide its Functional Designation. Basic shape properties can also be applied to filter candidates the earliest possible.
- This is done by the help of a Functional Designation Taxonomy.
  - Example: A rivet has two planar supports and two loose shaft links, that link the component to two different pieces which are in turn linked to each other or to a third piece through planar support(s).




A is a Rivet

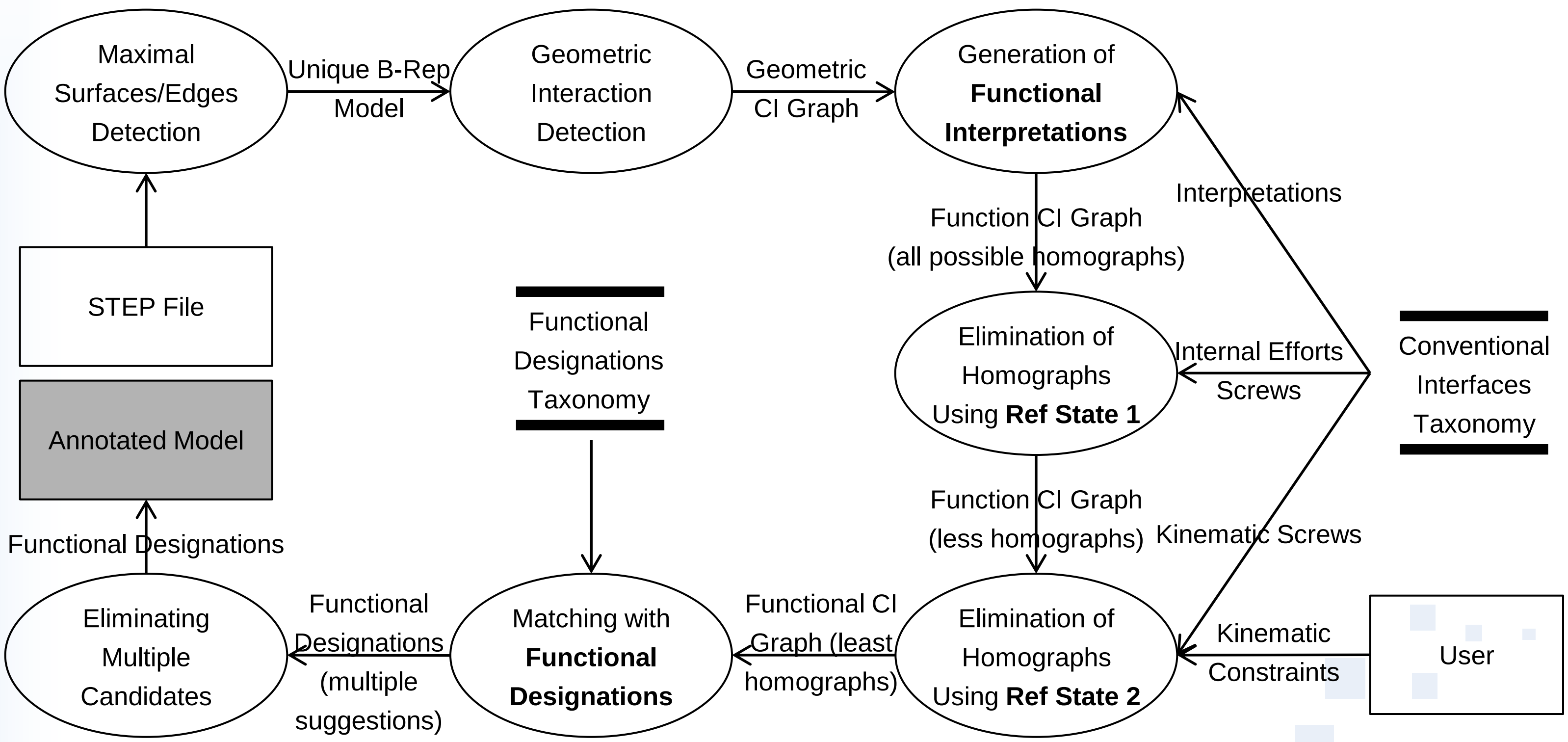
# Data Flow Diagram



# Eliminating Multiple Candidates

- Interfaces may lead more than one possible solution.
- Criteria are needed to select the most meaningful option:
  - Mechanical state: minimize the amount of functions per component.
  - Kinematic state: No internal mobility in the general case.

# Data Flow Diagram



# Conclusions

- Emphasis is put on the geometric interaction between objects (representing components) rather than the geometric properties of objects themselves.
- Analysis of DMUs shows the merit of this approach.