FACULTY OF ENGINEERING
UNIVERSITY OF ALBERTA

Mechanical Engineering

Interoperability and Productivity in Engineering IT

by
Dr. Yongsheng Ma, University of Alberta, Canada

Technocon 2010

http://www.technocon.ca/

THE CITY OF Edmonton

moving life forward

Industrial Requirements – Driving Trends

Mechanical Engineering

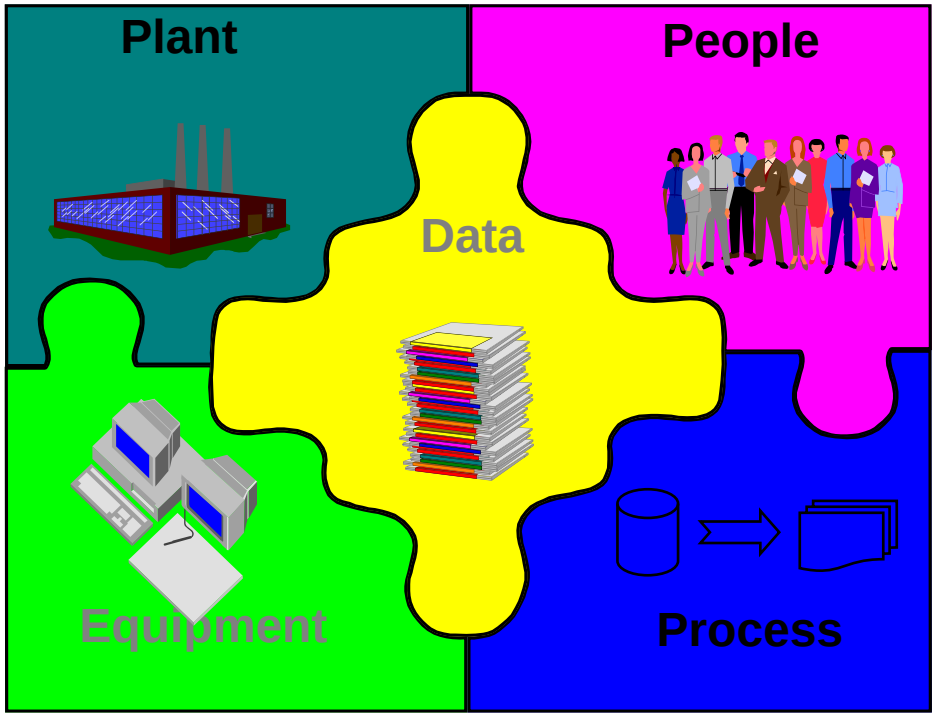
- **Concurrent and Collaborative engineering has been a vast and active research domain in engineering informatics**
- **OEM Enterprises have been transformed phenomenally from self-contained enterprises or hierarchical supply chains into open, dynamic and global enterprises.**

- **Such globalization of industries leads to the dramatic change of information management approach in enterprises.**
- **Severe competition in delivery time, fast new-product development and predictable quality reliability has resulted in more and more fully-digital design and manufacturing practices.**
- **Product lifecycle modeling, analysis and optimization**

Concurrent and Collaborative Product Lifecycle Engineering & Management

Mechanical Engineering

- Modeling products and processes is critical to the industry.
- Coherent integration and lifecycle management are the major challenges.
- A company's intellectual assets are simply in the form of puzzle pieces.



Computers are incredibly fast, accurate, and stupid; humans are incredibly slow, inaccurate, and brilliant; together they are powerful beyond imagination.

--Albert Einstein

Research on PLM system interoperability with a feature-object-based approach

Mechanical Engineering

Source: http://www.intergraph.com/default.aspx

OUTLINE

- **Current State of Art of Interoperability in PLM**: Data Exchange; Typical features; Advanced Features; Engineering Informatics
- **Feature objects**: Feature Modeling; Advanced features; Generic features; Feature object based systems
- **Proposed framework**: Data structures; System designs; Potential Solution

State of the Art – Data Exchange

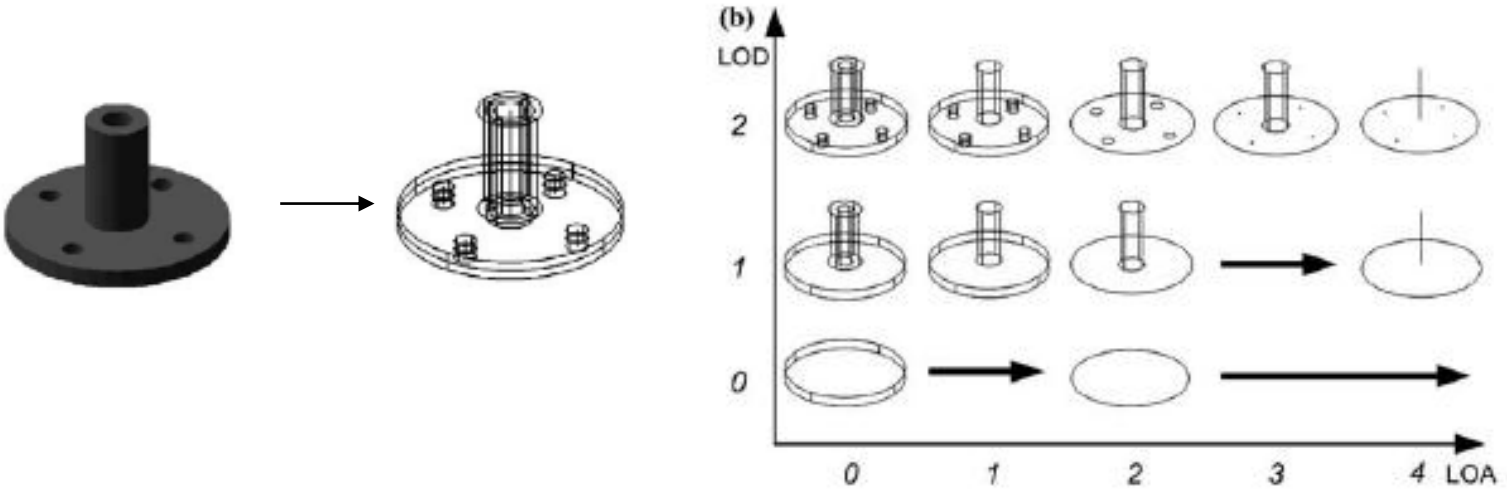
Mechanical Engineering

- Traditionally, interoperability refers to the ability of different software packages to exchange system data.
- IGES, STEP, SAT
- Standards for data format and communication protocols have been developed and adopted by different industries.
- So far, interoperability has only been investigated at the data level.
- Semantic information is lost during data exchange
- Data exchange creates a tremendous number of inconsistent copies of data files in collaboration

Source: http://www.tedesigns.ca/sample21-96id-21000-265-psi.html

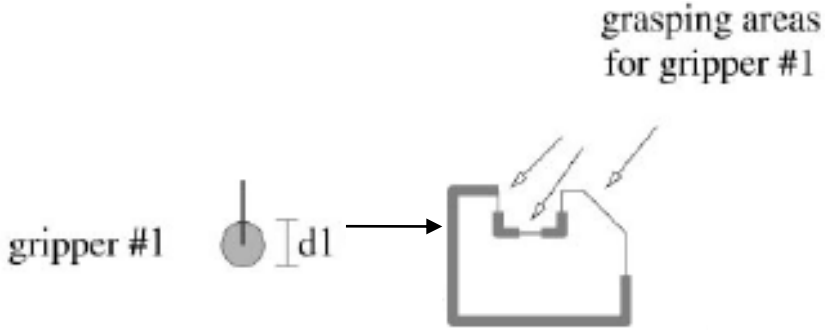
State of the Art – Typical Features ("hard coded"?)

Mechanical Engineering

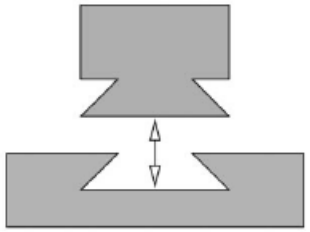


(a) Solid model (b) Merged model (c) Multiple representations of each feature

- Each idealization feature contains multiple representations, from [Lee 2005]



(a) A handling feature corresponding to gripper #1



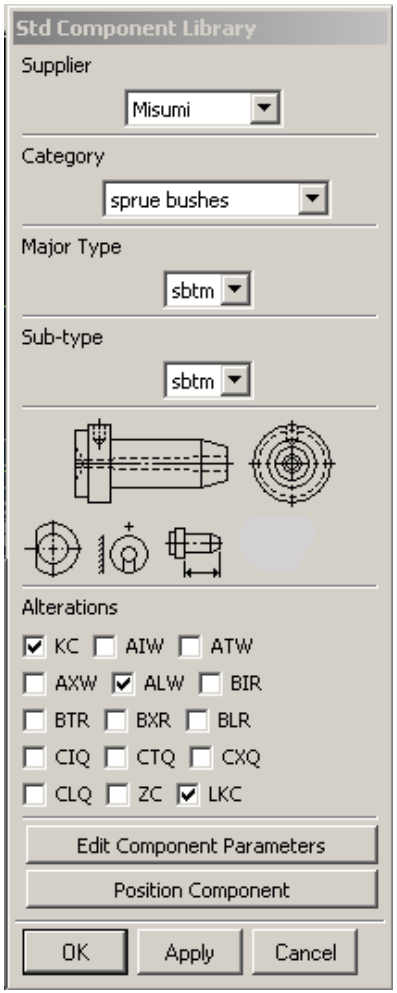
(b) A dove-tail connection feature

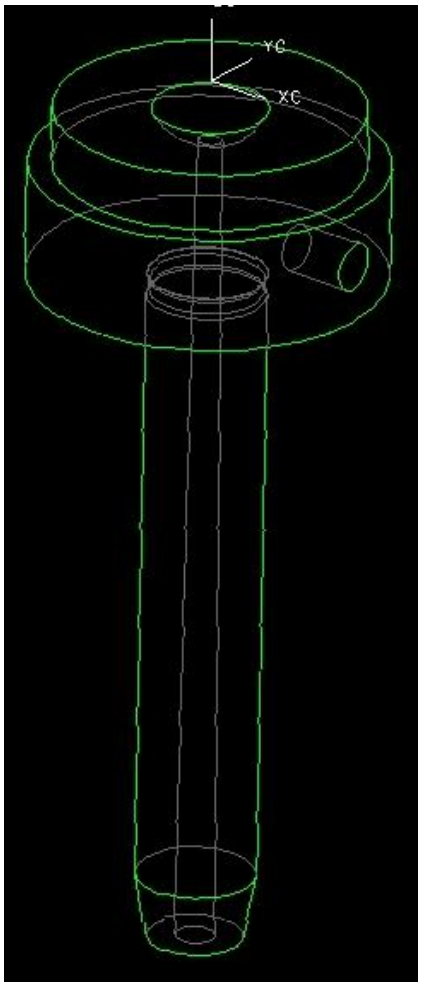
- Handling and connection features, from [Holland 2000]

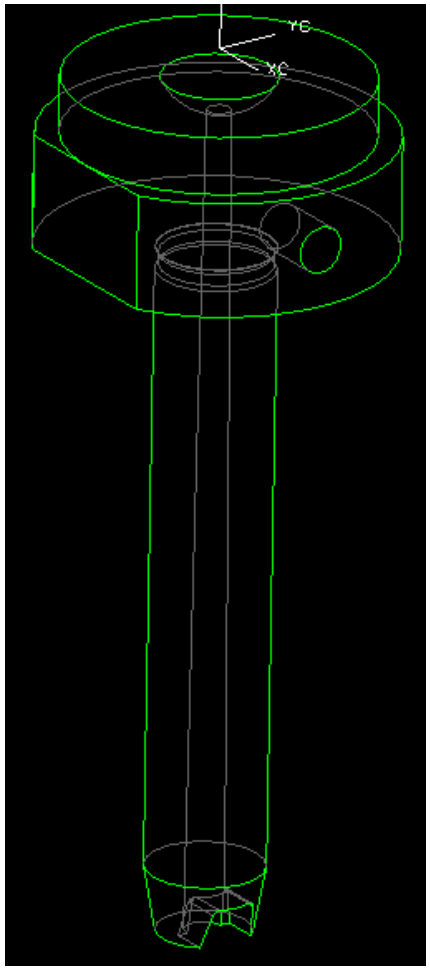
Feature Objects

Mechanical Engineering

- **Fine grain associative features applied at the component level** [Ma 2003]



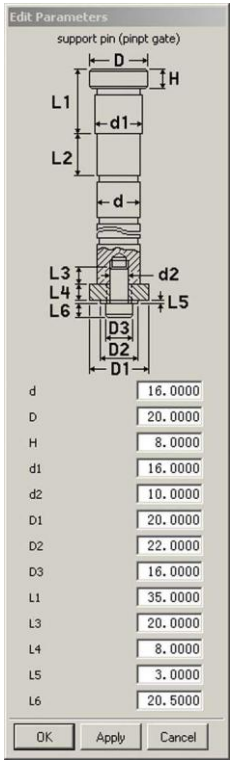


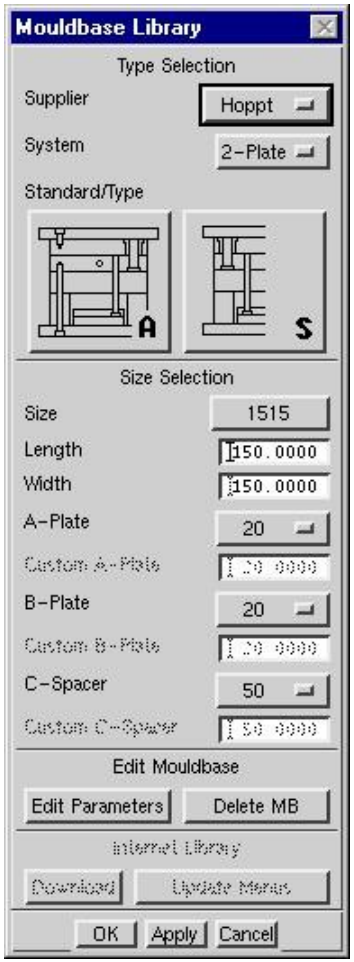


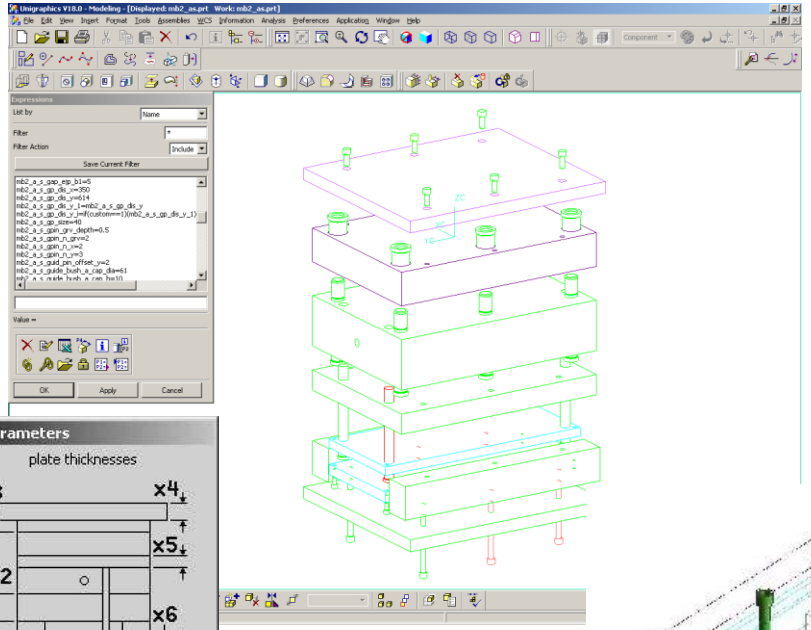
Feature Objects

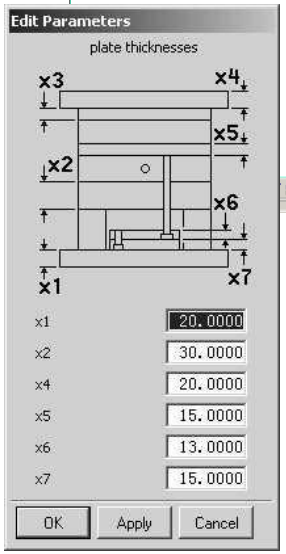
Mechanical Engineering

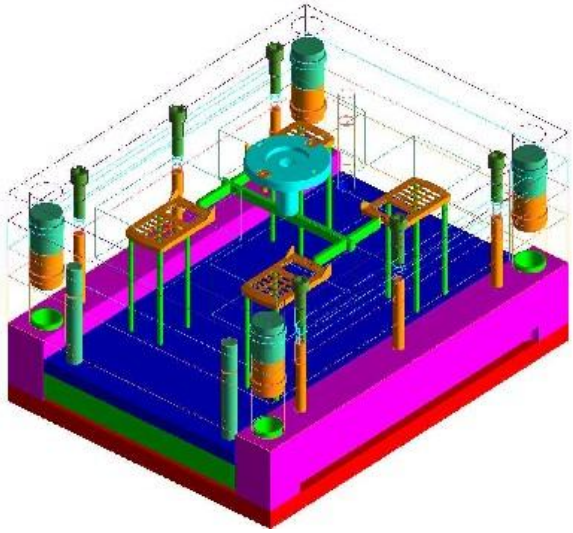
- Associative features and its modelling method – assembly design features [Ma 2005]









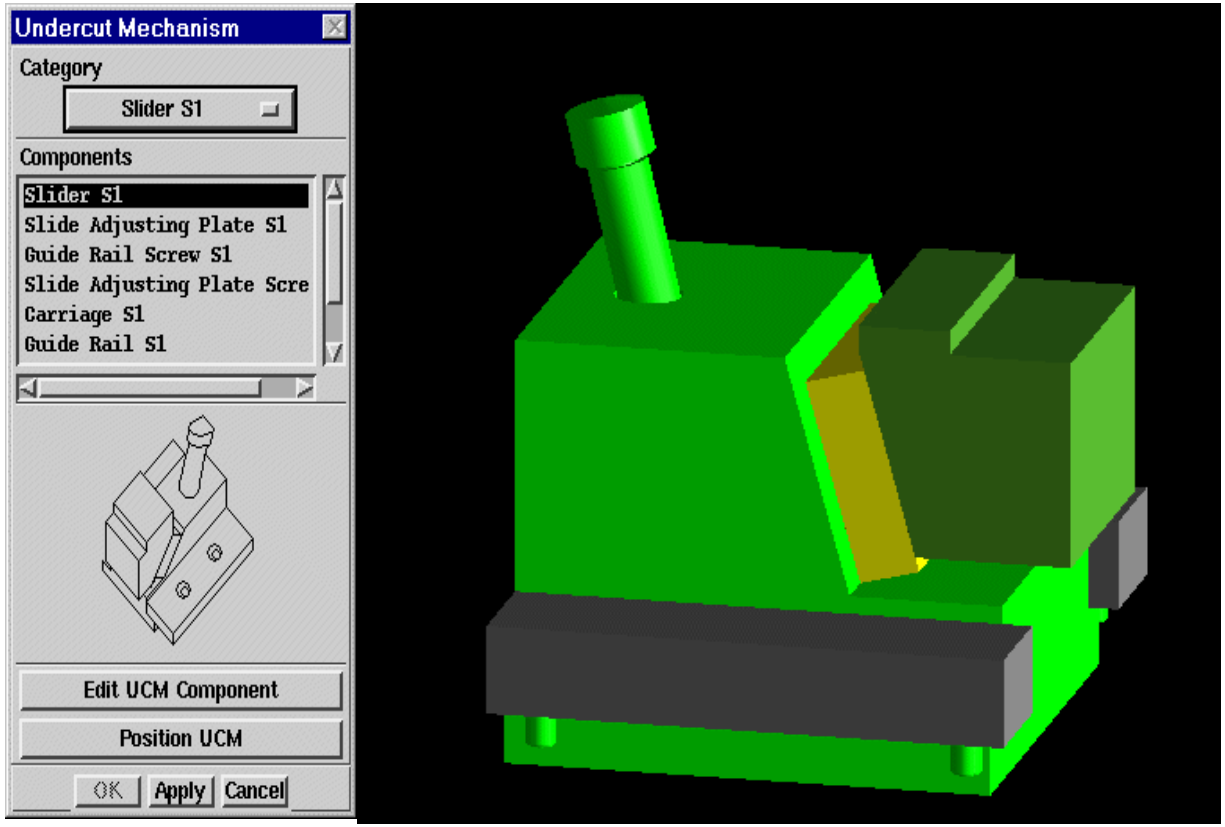


www.ualberta.ca/~yongshen

Feature Objects

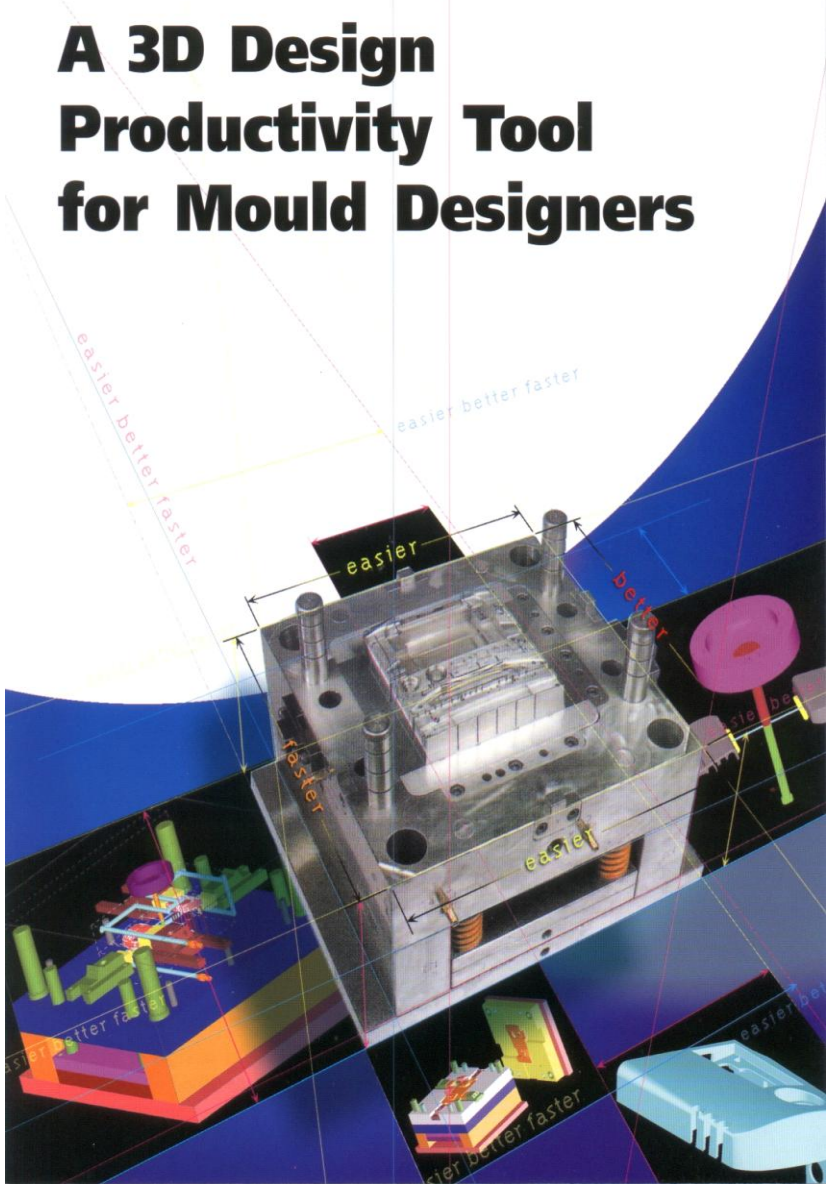
Mechanical Engineering

- Associative features and its modelling method – assembly design features



From QuickMould to MoldWizard

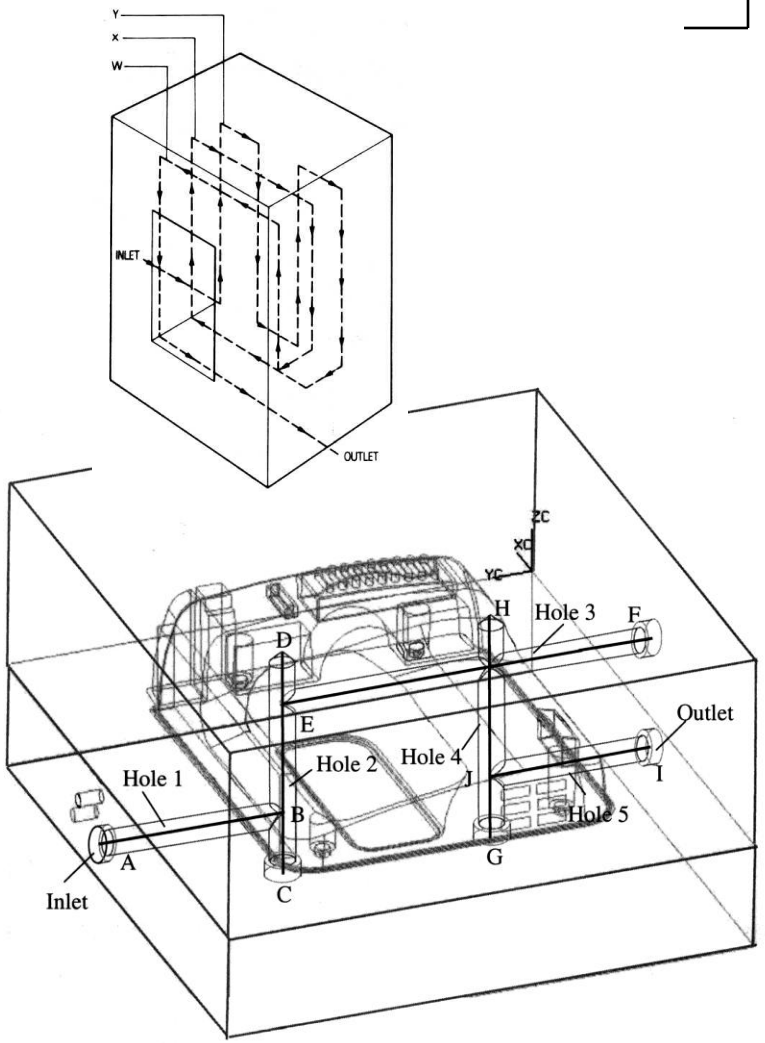
Mechanical Engineering



Fine-Grain Feature Objects

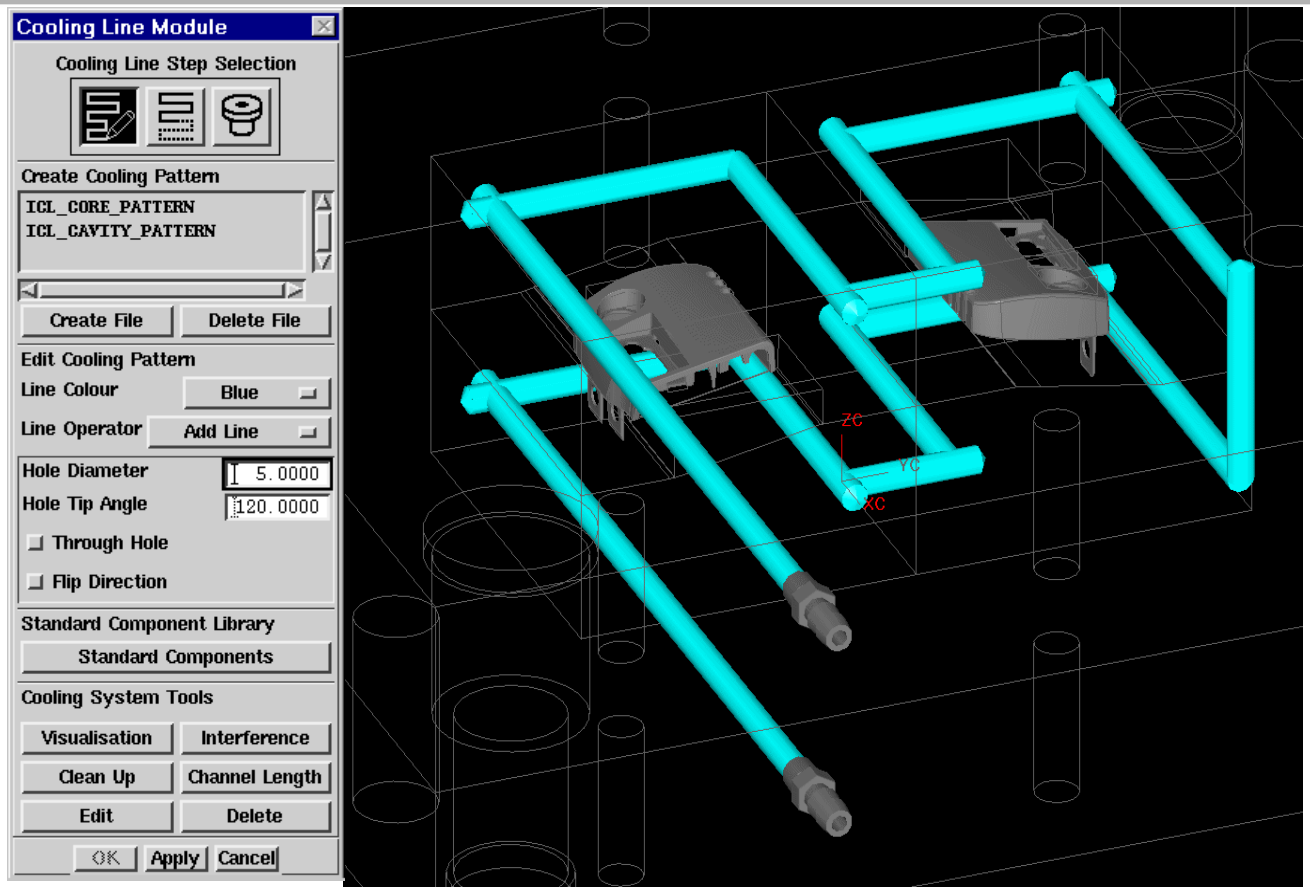
Mechanical Engineering

- Feature properties defines the geometric entities which behaviors defines the related constraints and logics in functioning methods throughout the lifecycle of any feature instance.
- An example associative feature, cooling channel pattern in plastic injection mold design was given in [Ma YS 2003]
- An initial sketch-based conceptual pattern in the early mould design stage is implemented and its downstream cooling '*hole*' features are derived from the pattern; and then the related assembly interfacing features and associated standard components at the manufacturing and assembly stages are associatively generated and managed via a well defined feature class model.



Fine Grain Feature Objects

Mechanical Engineering

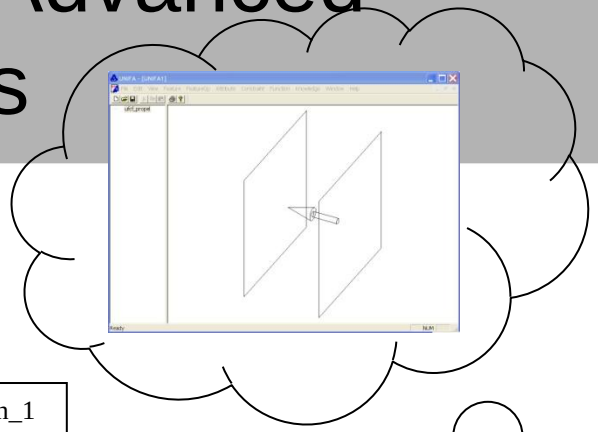


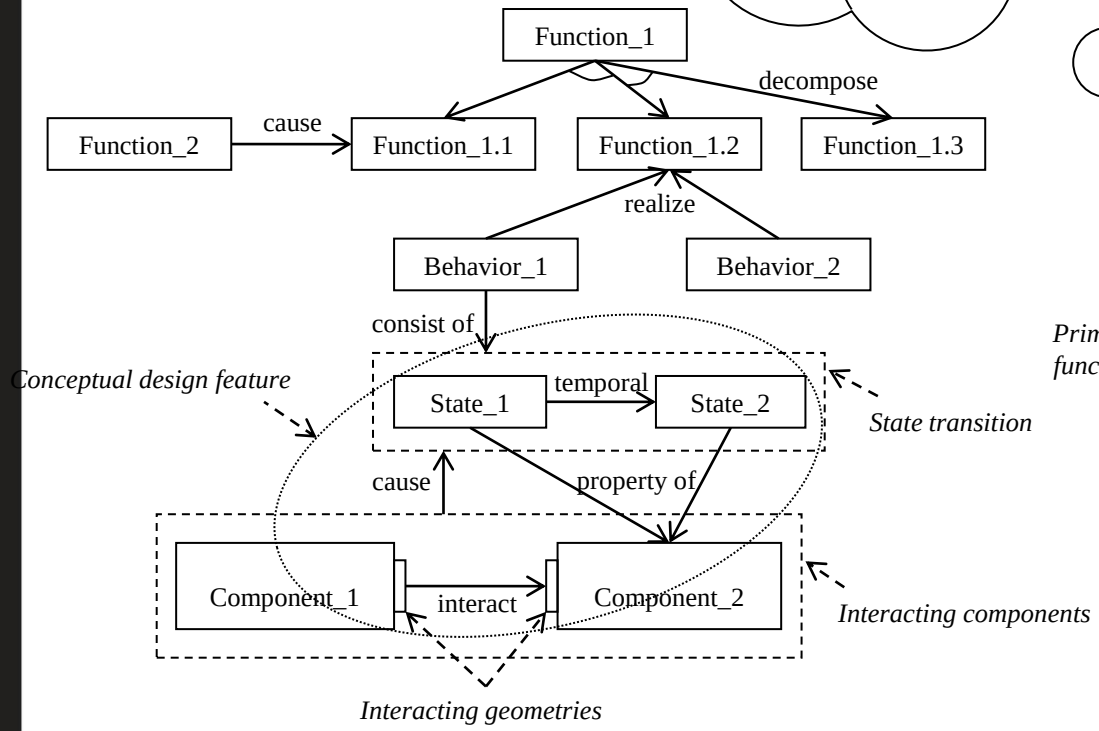
•The *associative feature* concept expands feature definitions of specific application related shapes into a set of well-constrained geometric entities.

• By using object-oriented approach, a feature type can be modeled in a declarative manner which basically consists of the properties and behaviors.

Evolvement of Advanced Feature Objects

Mechanical Engineering





Primitive functions

F1: Create an external undercut region of a molding

F1.1: Form the undercut region during the molding process

F1.1.1: Imprint the undercut region using the slider head

F1.1.2: Enclose the impression area

F1.1.3: Position the slider head

F1.1.4: Restrict the slider head

F1.2: Release the undercut prior to the ejection process

F1.2.1: Propel the slider body

F1.2.2: Guide the movement of the slider body

F1.2.3: Restrict the movement of the slider body

F2: Put the slider back to its molding position

F2.1: Propel the slider body

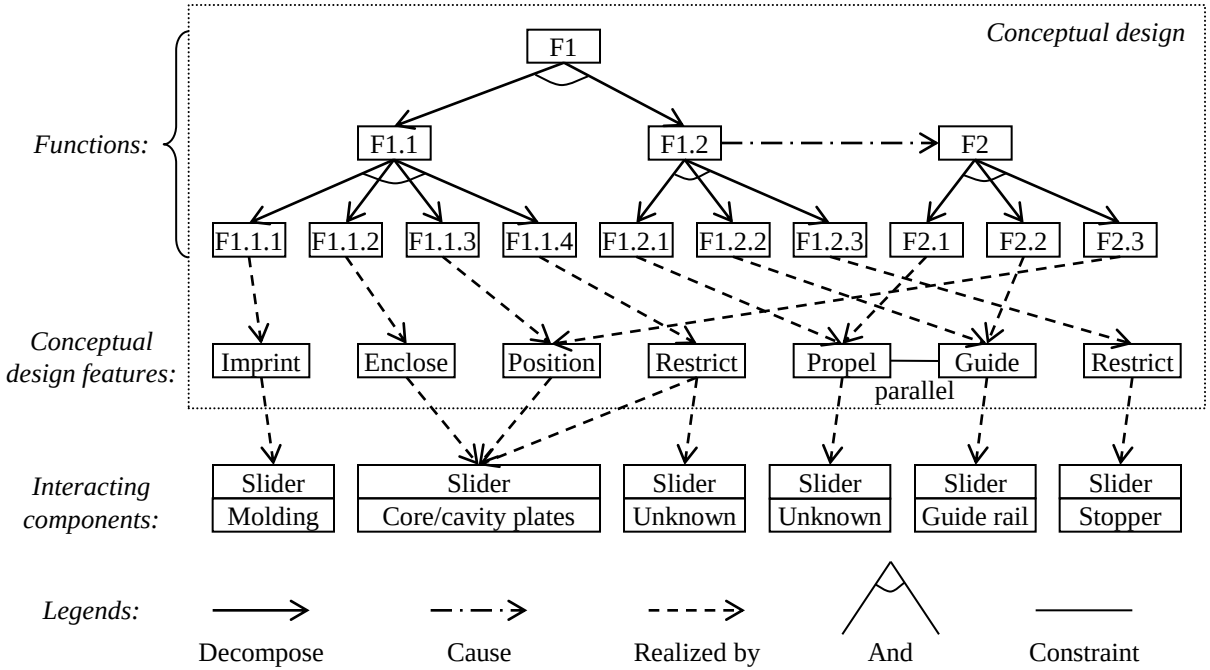
F2.2: Guide the movement of the slider body

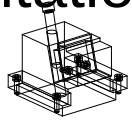
F2.3: Position the slider head

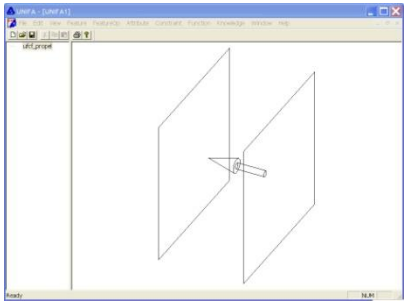
- Allowing multi-stage evaluation and evolvement
- Allowing cross-boundary referencing and sharing
- Allowing encapsulated constraint solver to be dynamically bonded into object methods

Evolvement of Advanced Feature Objects

– Fine Grain Access and multi-facet representation







- With the built-in object polymorphism capability, a systematic modeling scheme for a generic and abstractive parent feature class, with levels of specification as per application domain requirements, can be developed.
- Such a generic feature definition scheme unifies many traditionally defined, application-oriented feature definitions and supports XML representation and fine grain database repository.
- Under the associative feature concept, where the associative constraints across multiple phases of applications of a product life cycle, complicated engineering features (patterns) and engineering intent can be implemented.

Objectives

Mechanical Engineering

- This presentation addresses a research approach proposing a system design and a set of generic methods so that to embed engineering knowledge and to achieve interoperability at the feature level in an open collaborative engineering environment.

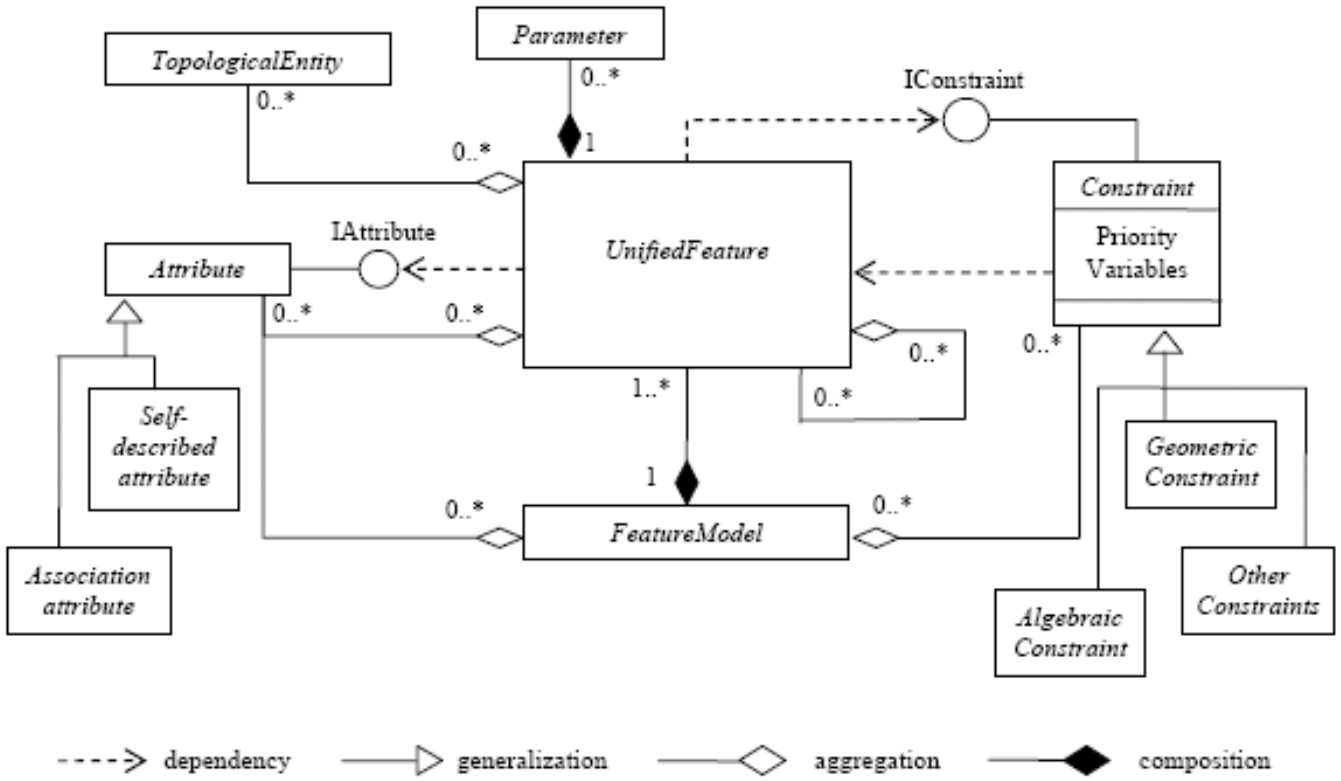
- Ideally, the proposed new approach would allow knowledge rules to be embedded into the constraints of features supported by the complex associations of a multi-application engineering repository.

- Potentially, the system proposed offers user-defined feature types that support flexibility in feature-based information definition, sharing and mapping.

Proposed Framework – Data Structure

Mechanical Engineering

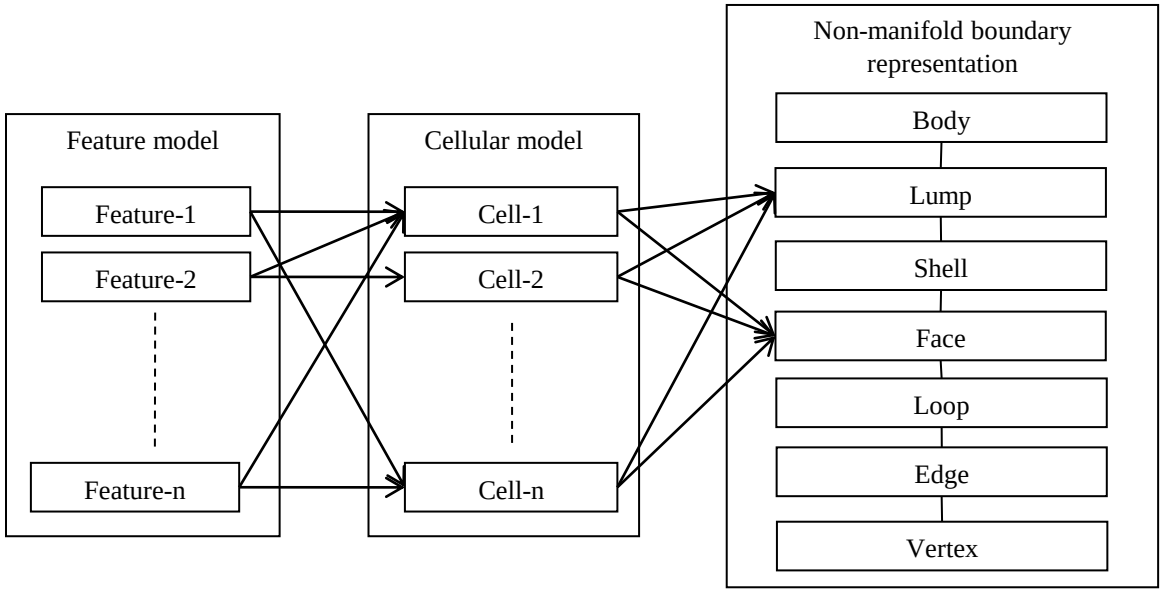
- Unified feature model with constraint management modules



Proposed Framework – Data Structure

Mechanical Engineering

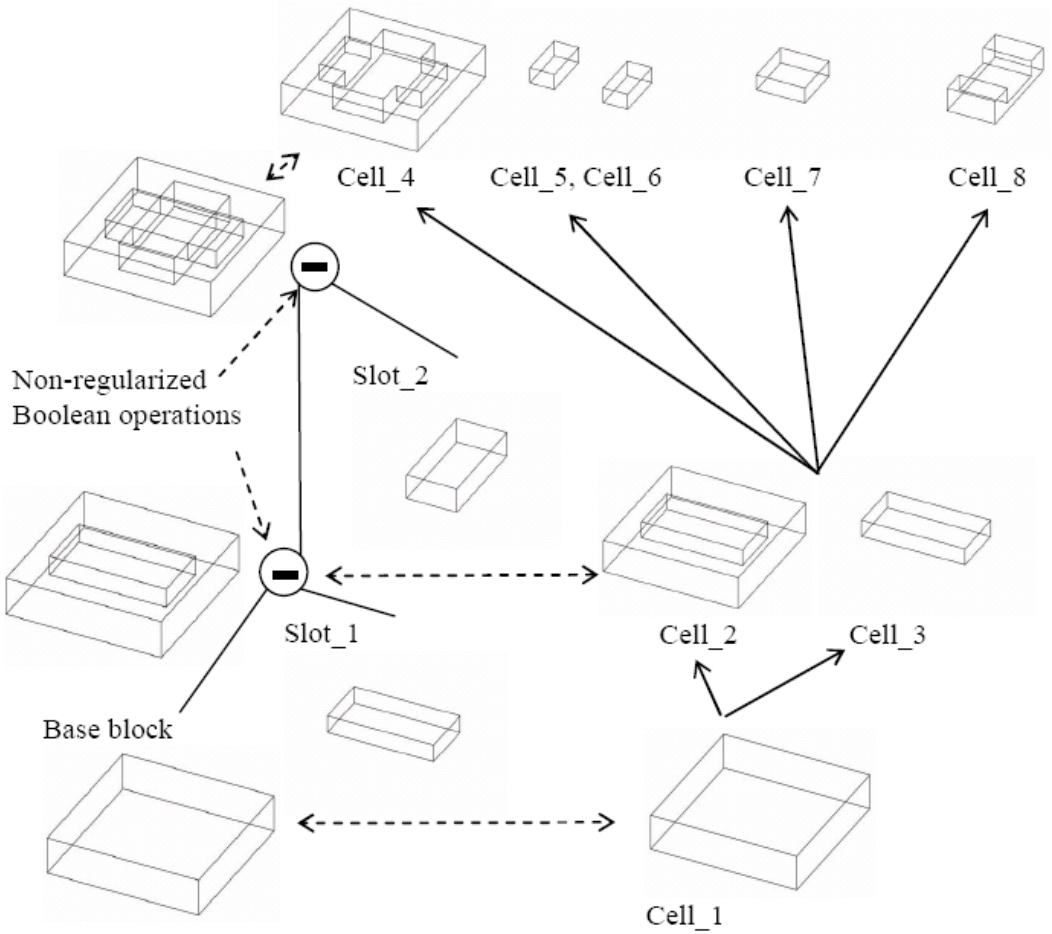
- Relations between feature model, cellular model and the boundary representation



Proposed Framework – Data Structure

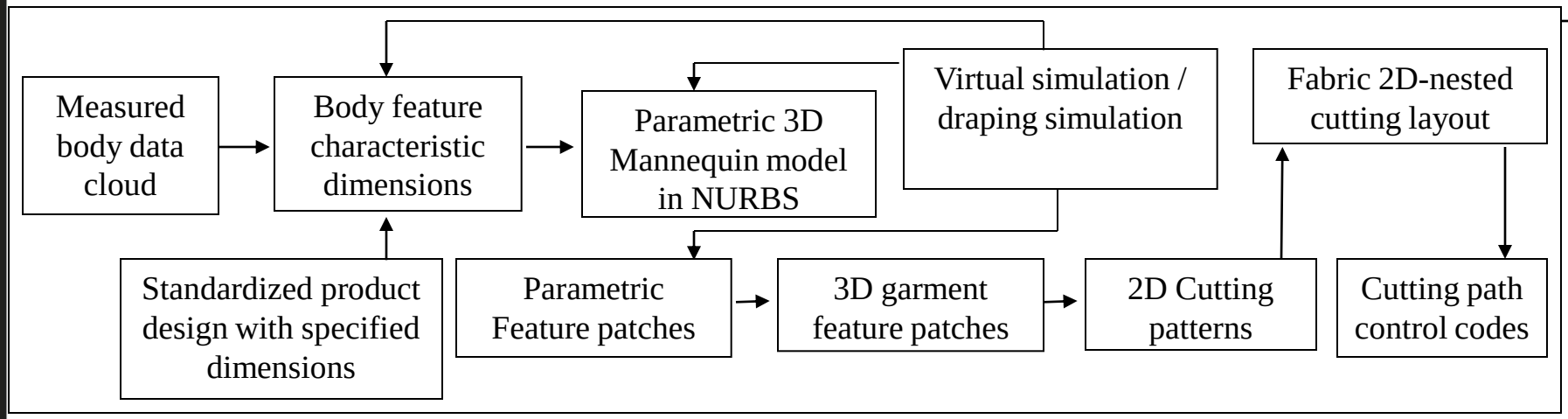
Mechanical Engineering

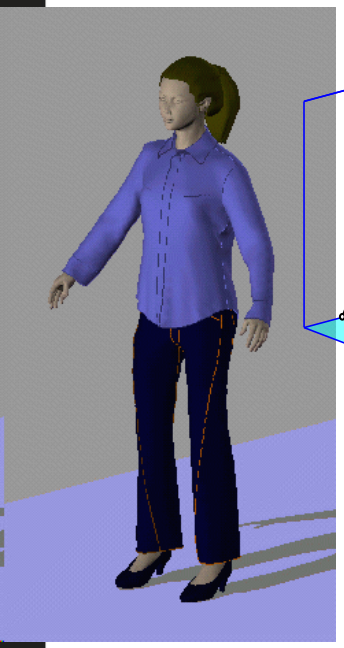
- Multi-facet cellular model and the boundary representation evaluation

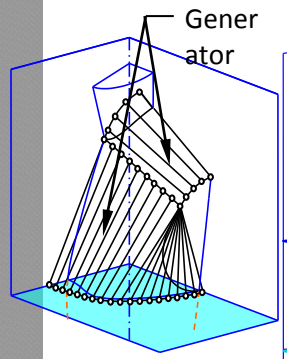


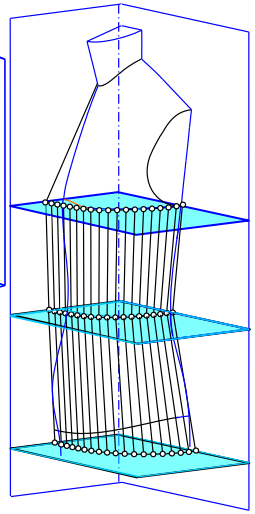
Application Field – Garment Design

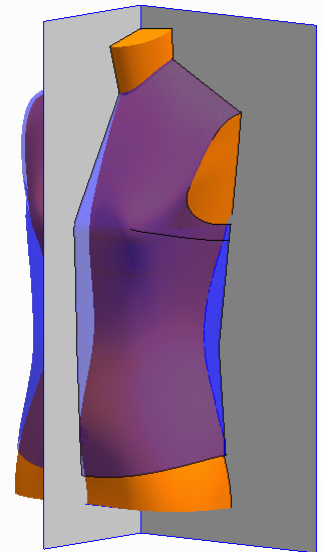
Mechanical Engineering

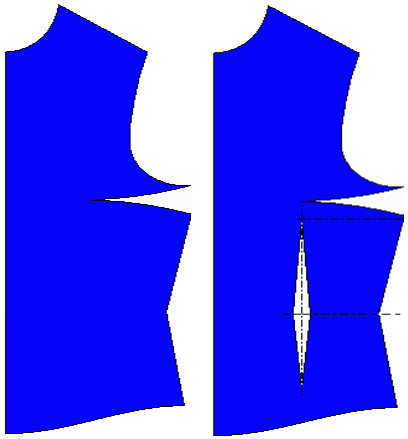


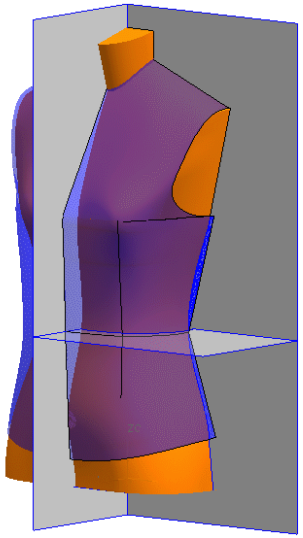






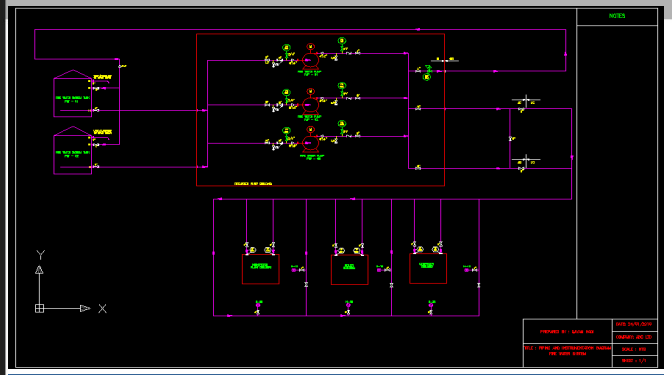


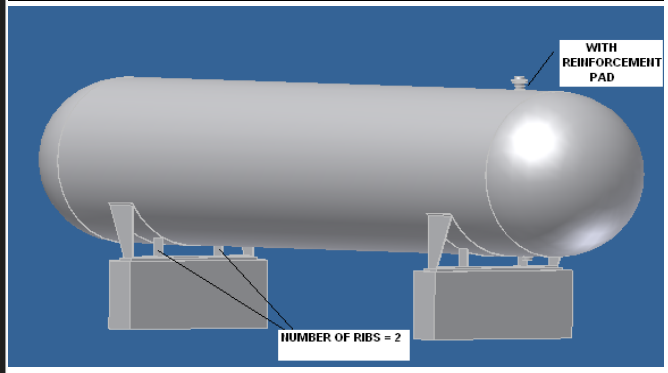


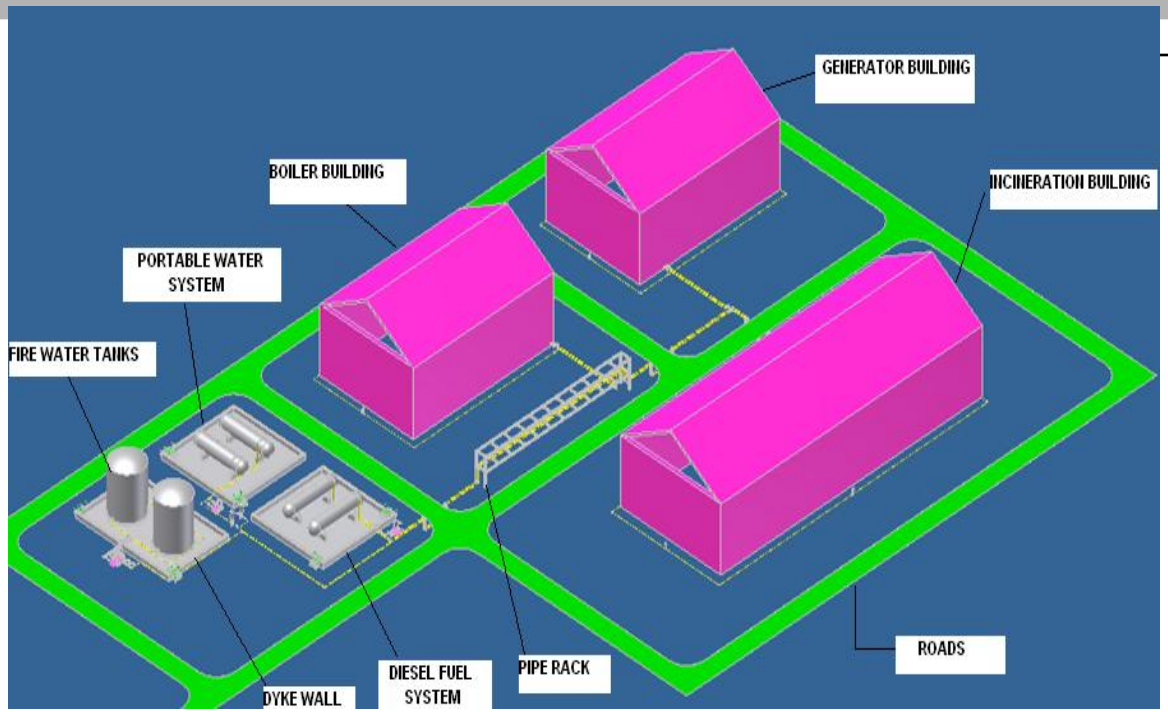


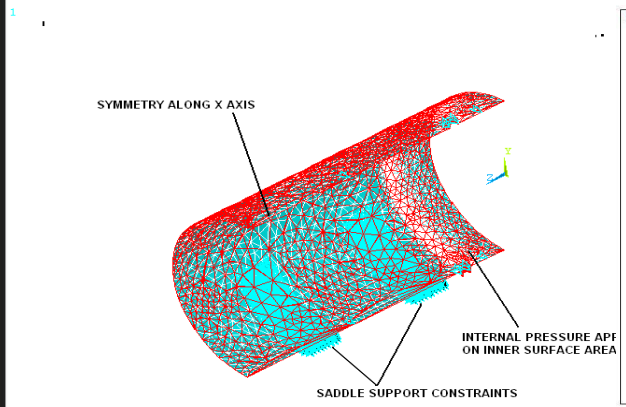
Application Field – Water and Fuel Supply System Design

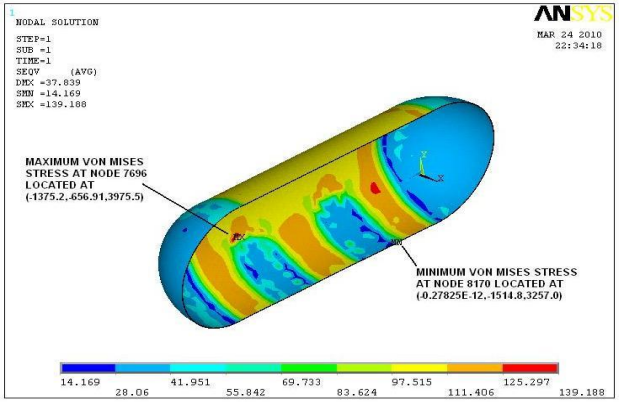
Mechanical Engineering

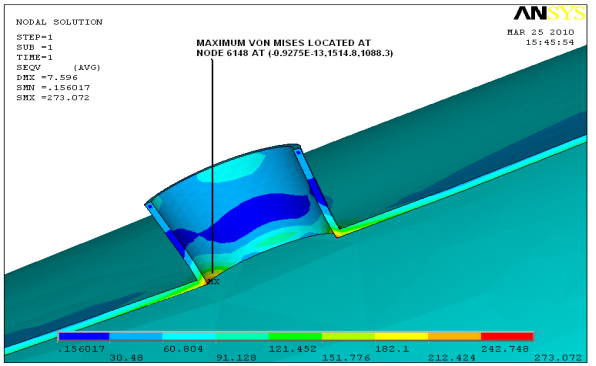








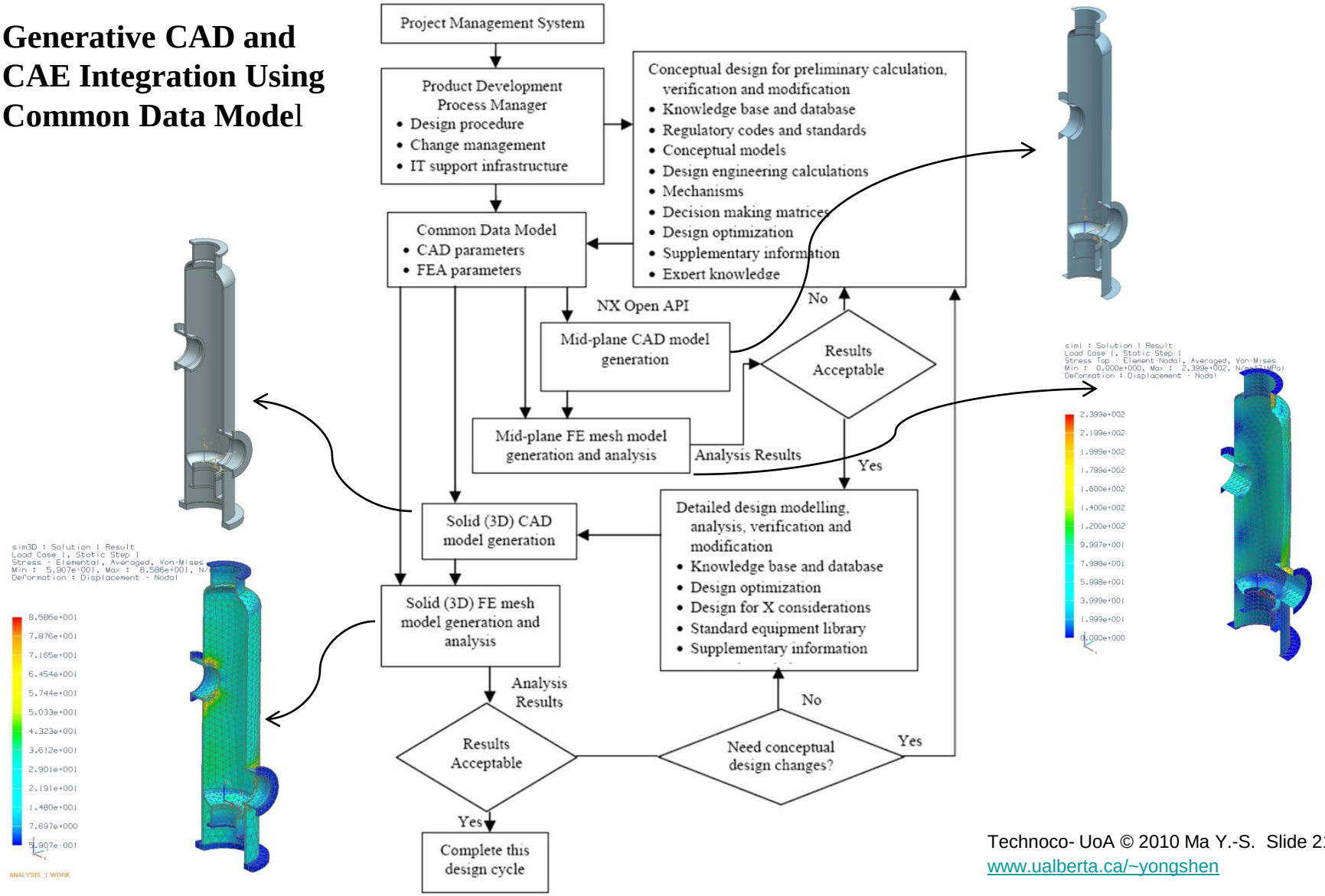




Application Field – Pressure Vessel Design

Mechanical Engineering

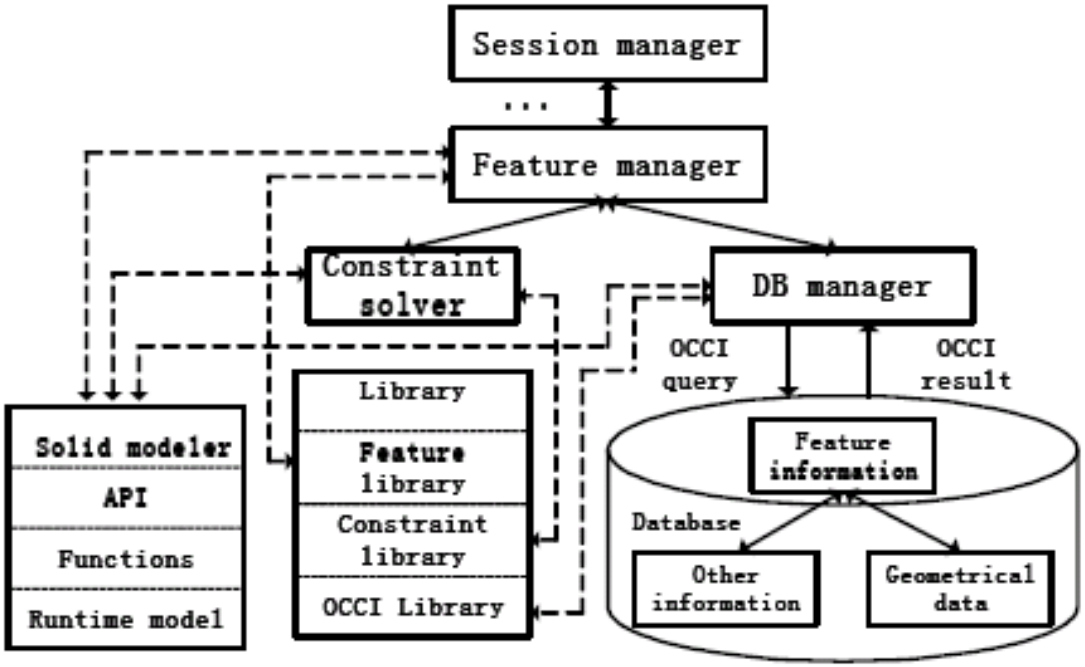
Generative CAD and CAE Integration Using Common Data Model



Proposed Framework – Data Structure

Mechanical Engineering

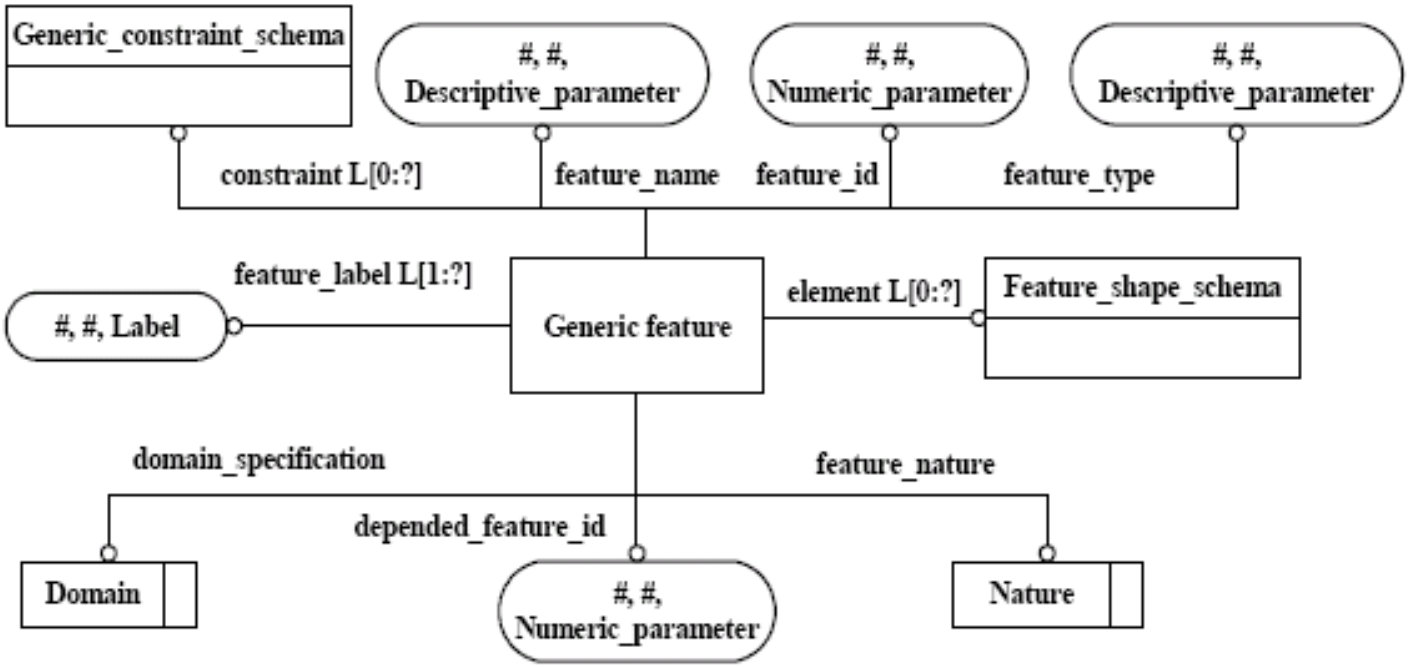
- Generic feature model in cellular model and the boundary representation in a feature oriented repository



Proposed Framework – Data Structure

Mechanical Engineering

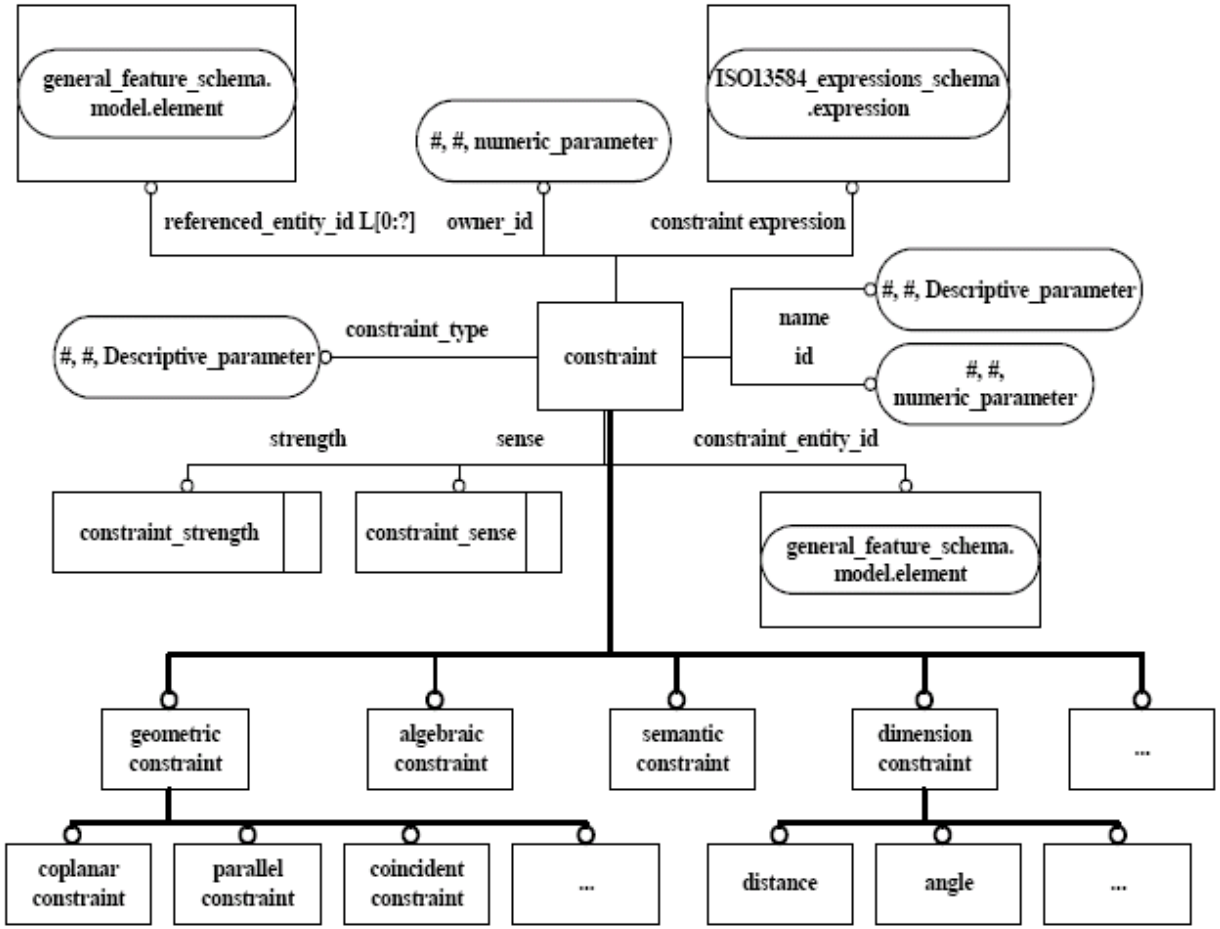
- Generic feature model in cellular model and the boundary representation in a feature oriented repository



Proposed Framework – Data Structure

Mechanical Engineering

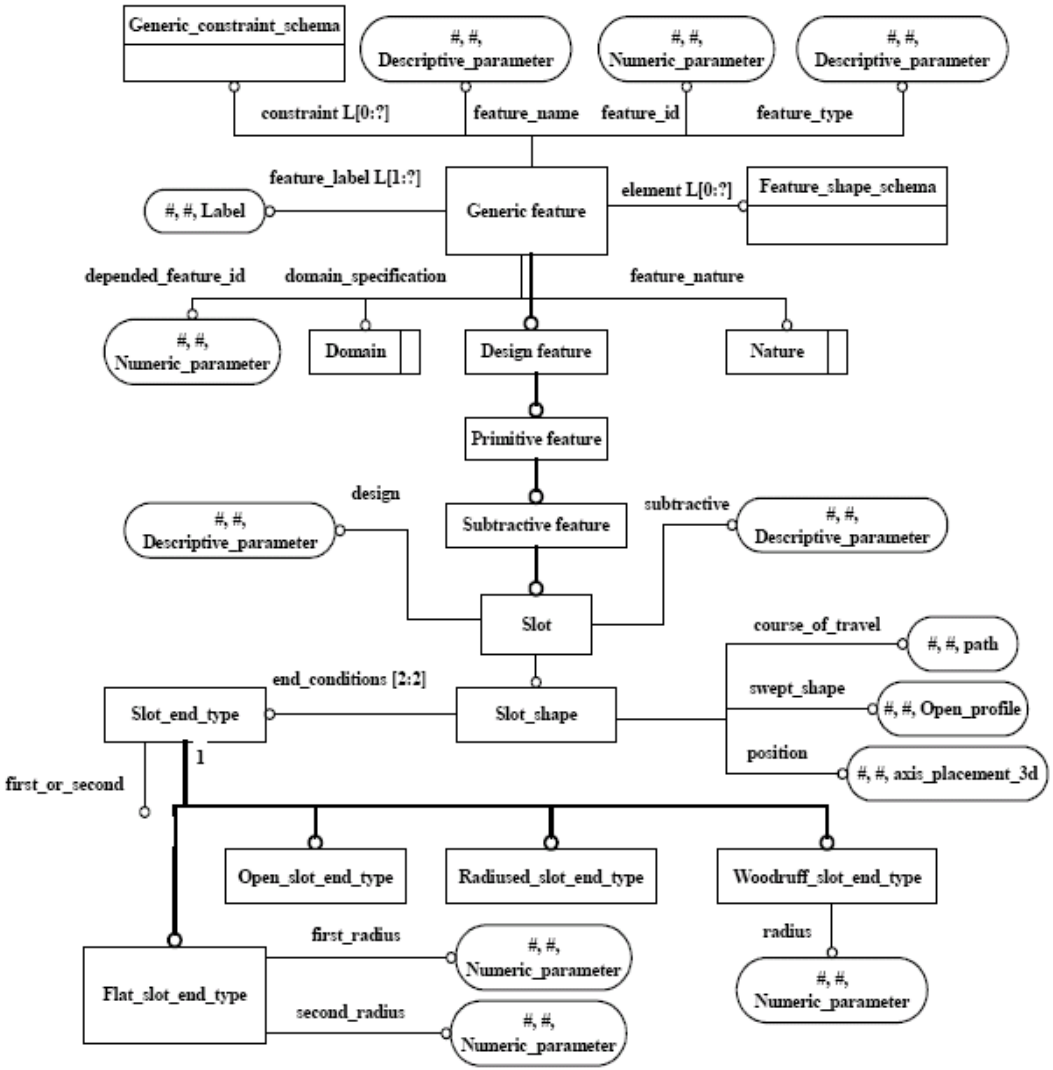
- Generic constraint model in a feature oriented repository



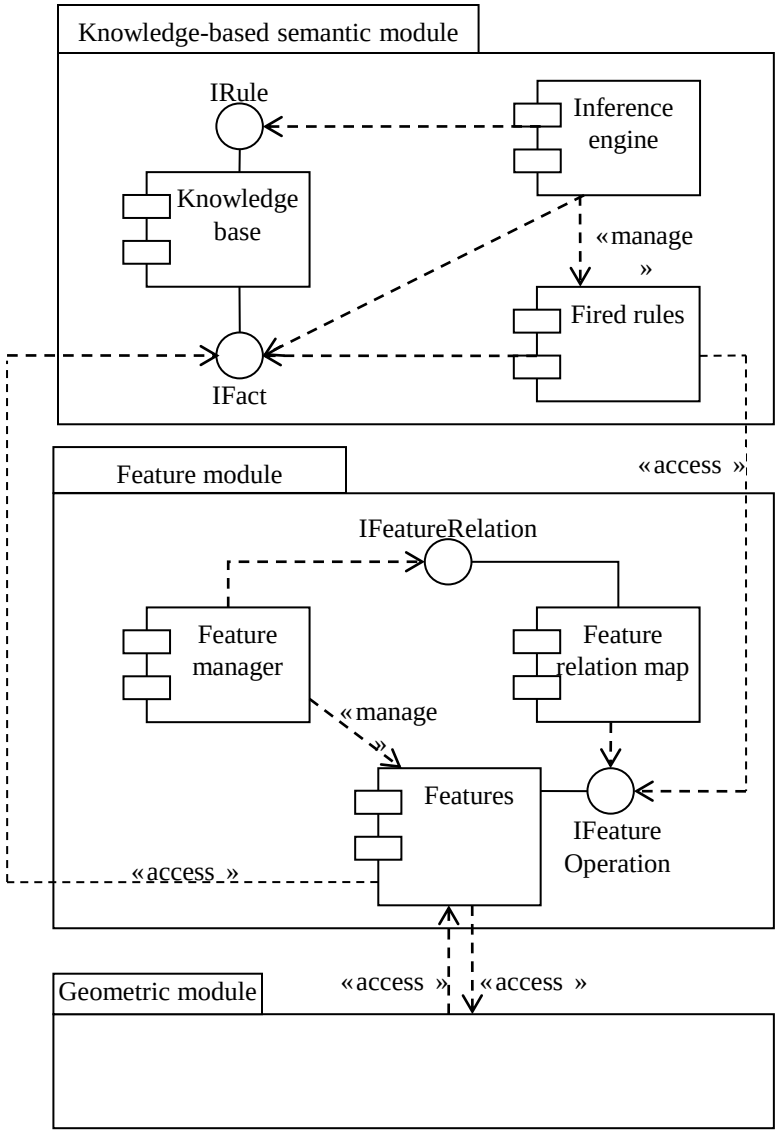
Proposed Framework – Data Structure

Mechanical Engineering

- feature model in a feature oriented repository



Proposed Framework – System Design

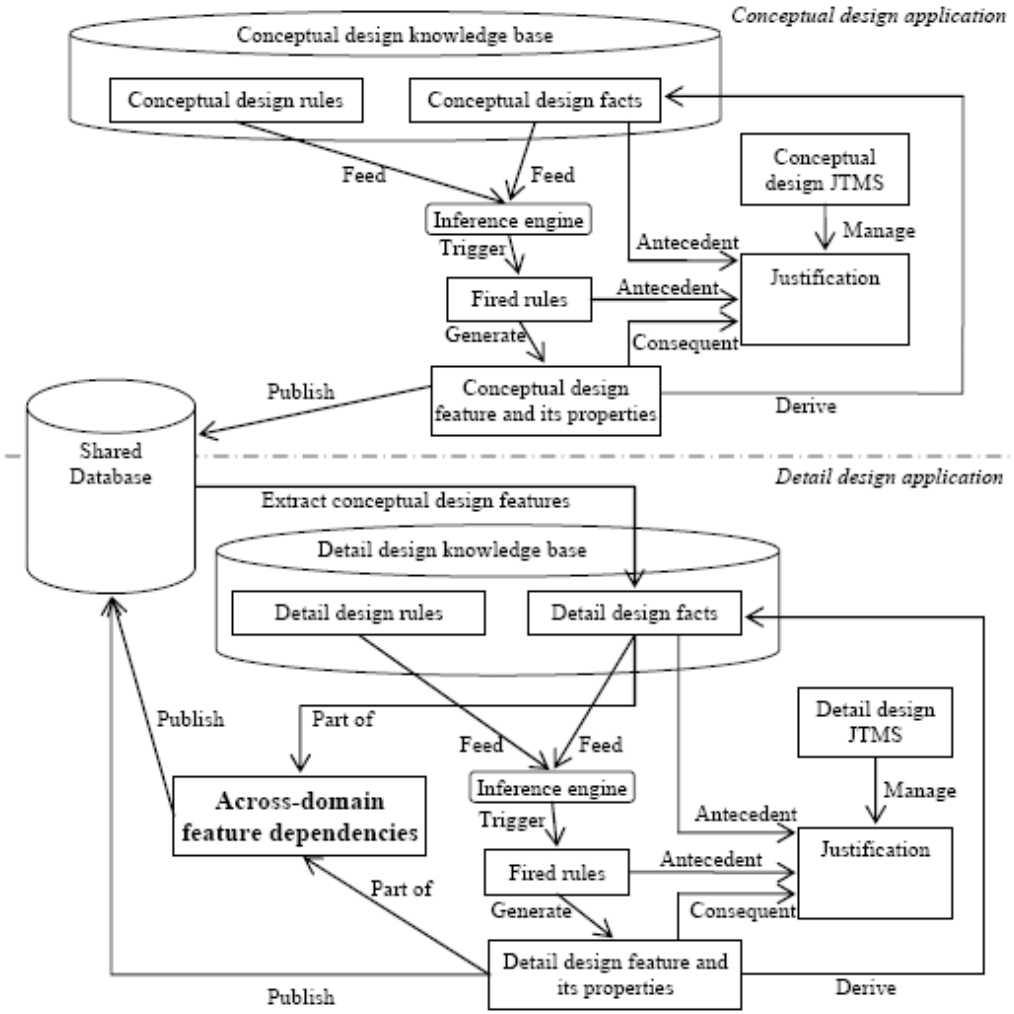


Unified integration structure and mechanisms

Proposed Framework – System Design

Mechanical Engineering

Feature interaction modules



Proposed Framework – Web Based Collaboration Service

Mechanical Engineering

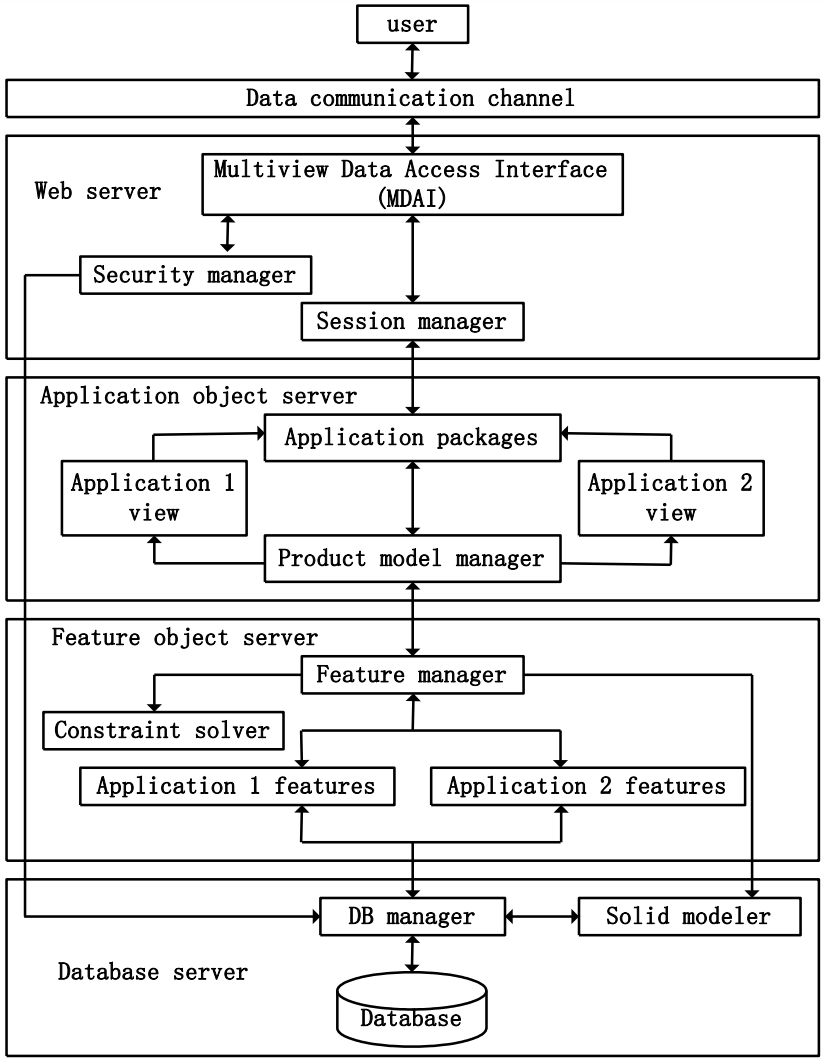
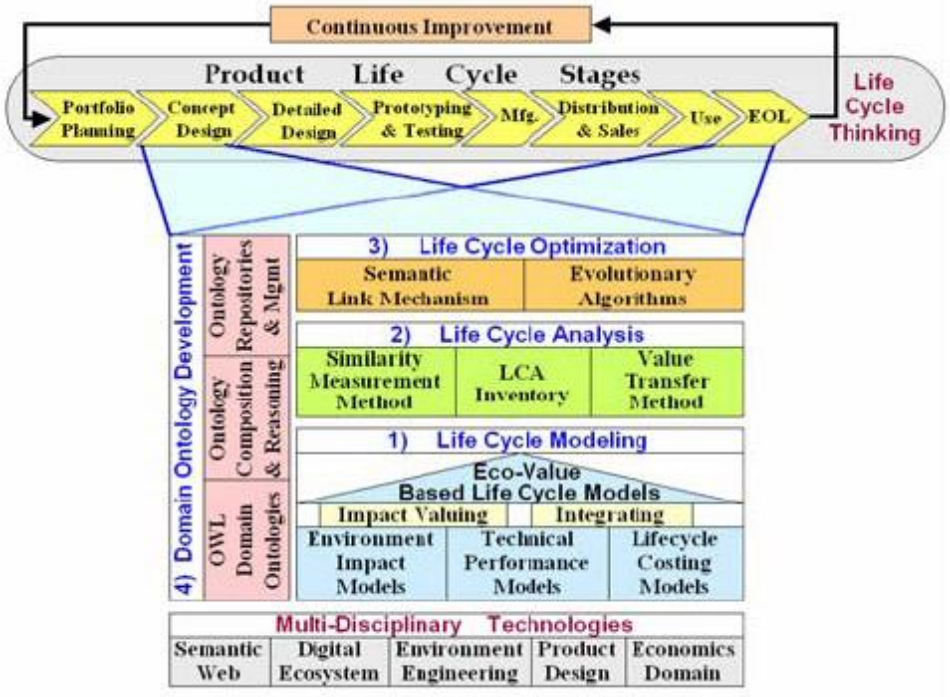
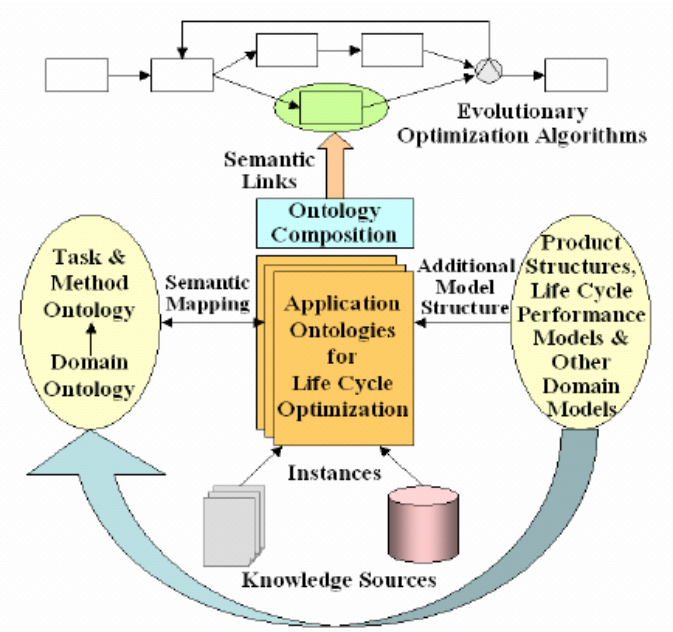


Figure 1. Overall system architecture

Proposed Framework – PLM Web Based Collaboration Services

Mechanical Engineering





Generic feature-based design & manufacturing – an open approach

Mechanical Engineering

•**Goal**: to reduce system incompatibility in order to support an effective and efficient collaborative engineering for multiple players, applications, and stages of product lifecycles. Eventually, as a result of this research, industry productivity and competitiveness can be enhanced on a significant scale.

•**Method**: Representing and manage knowledge rules to be embedded into the constraints of features together with the complex associations of different aspects of a product or project.

•**System design**: An open approach is to be taken, which offers self-defining feature types that support flexibility in feature-based information definition, sharing and mapping. The research methodology will be based on object-oriented software engineering methods and web-service-oriented architecture design

•**Novelty**: to develop a set of generic methods to embed engineering knowledge and achieve interoperability at the feature level in an open collaborative engineering environment.

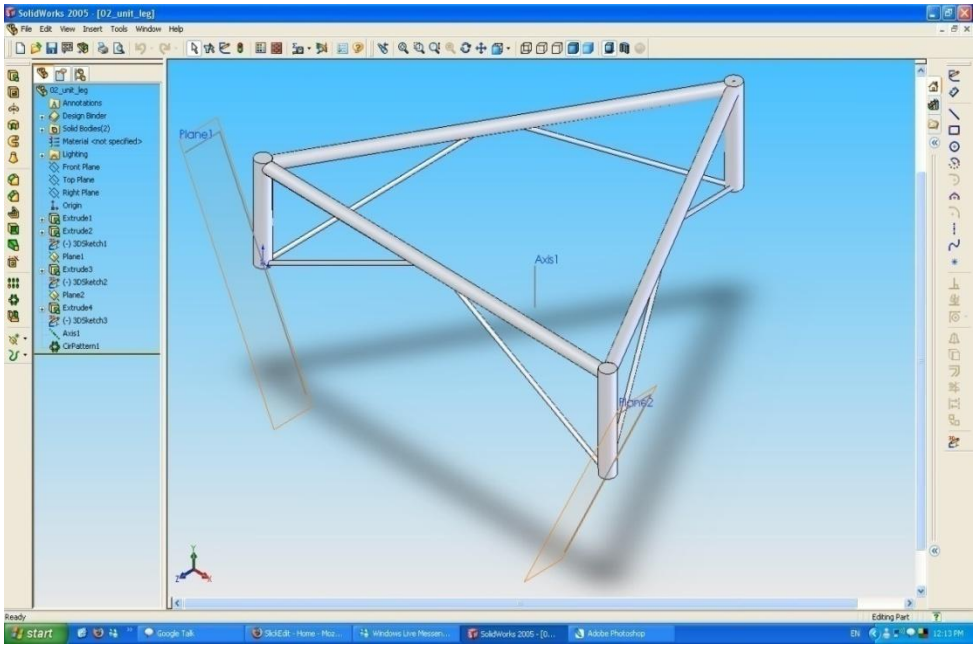
Case Study

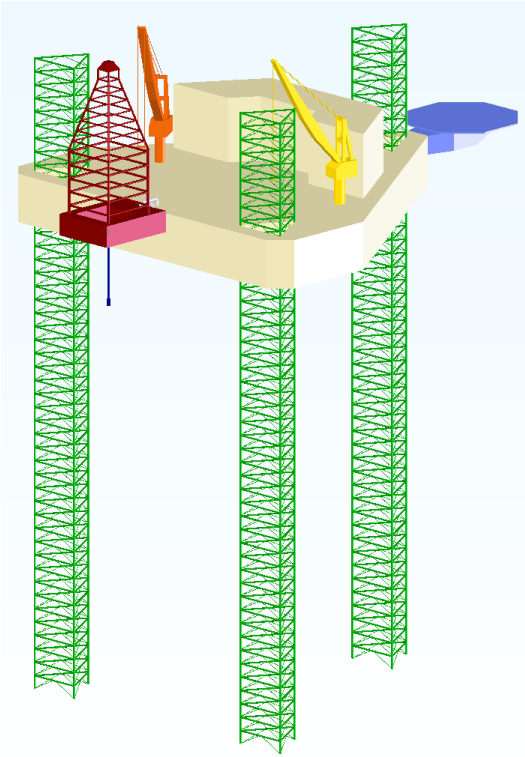
Mechanical Engineering

- Project manager determines main design parameters through Web-based App.
- Parametric relationship of jack-up rig components for CAD models
- Main parameters updated, new models created instantly
- Allow early assessment with little effort

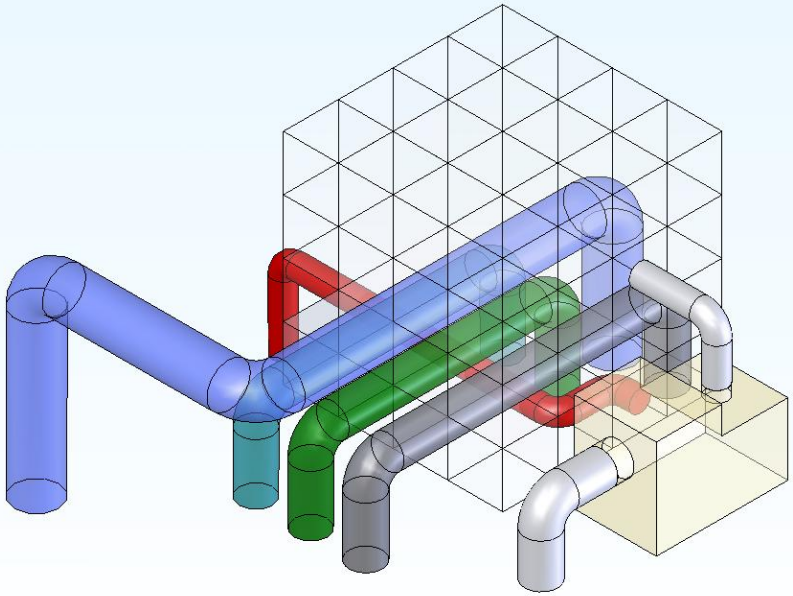
Case Study

Mechanical Engineering

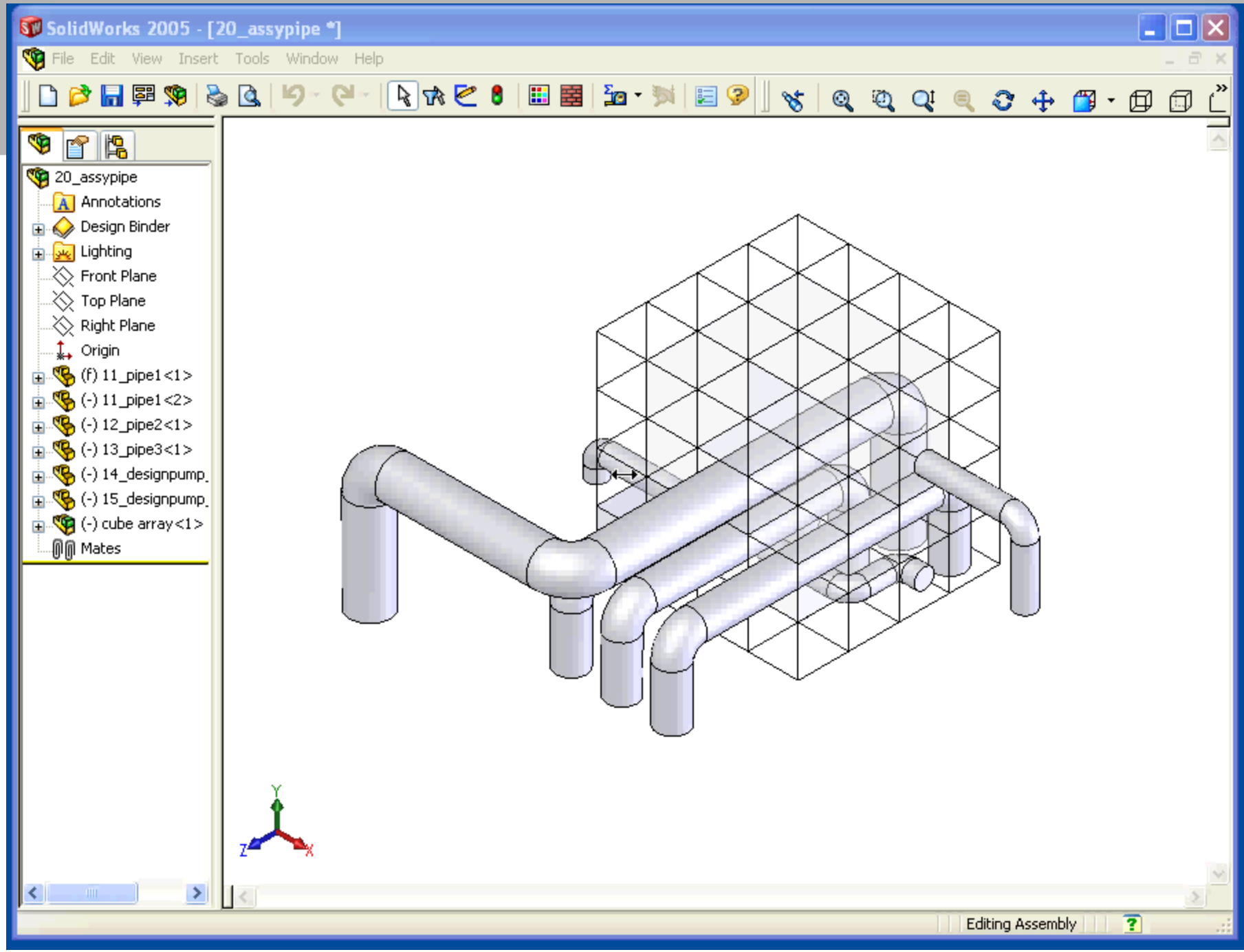




Case Study – Oil Rig Space

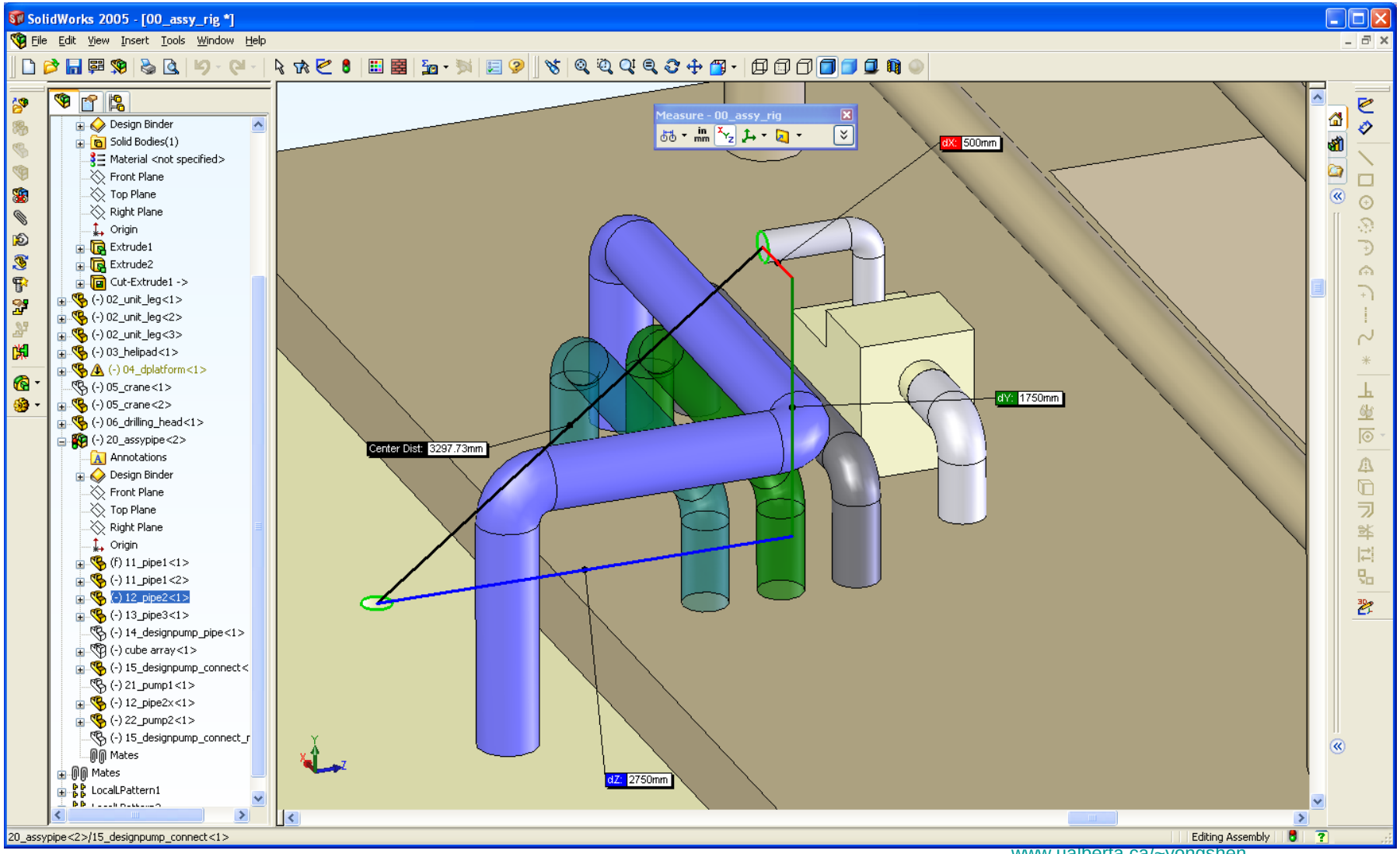


- Original design
- Pump fails and no exact replacement, new model used
  - Different pipe connection
- New pipe required
  - Complex piping
  - Space constraints

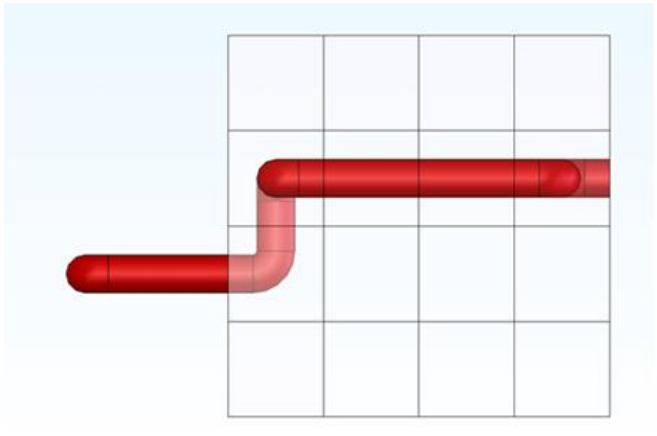

SolidWorks 2005 - [20_assypipe *]
File Edit View Insert Tools Window Help
20_assypipe
Annotations
Design Binder
Lighting
Front Plane
Top Plane
Right Plane
Origin
(f) 11_pipe1<1>
(-) 11_pipe1<2>
(-) 12_pipe2<1>
(-) 13_pipe3<1>
(-) 14_designpump
(-) 15_designpump
(-) cube array<1>
Mates
Y
Z
X
Editing Assembly

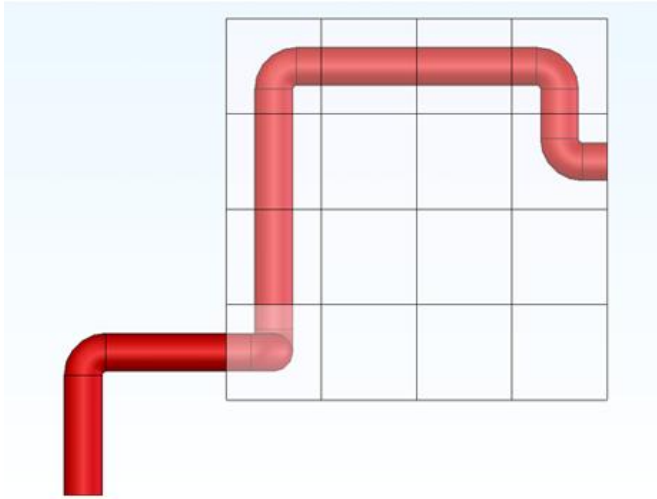
Automatic SMG Creation

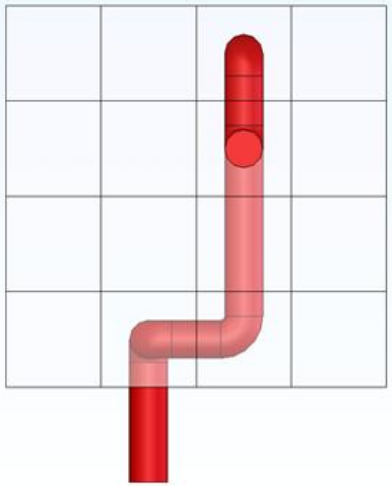
Mechanical Engineering



Case Study





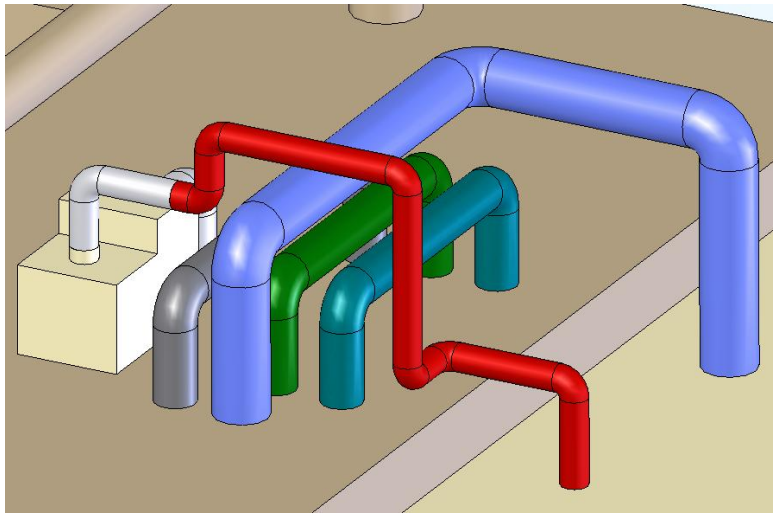


C:\WINDOWS\system32\cmd.exe

```
 1 Test empty_space[0][0][0] = 1  whilst occupied[0][0][0] =  0
 2 Test empty_space[0][0][1] = 1  whilst occupied[0][0][1] =  0
 3 Test empty_space[0][0][2] = 1  whilst occupied[0][0][2] =  0
 4 Test empty_space[0][0][3] = 1  whilst occupied[0][0][3] =  0
 5 Test empty_space[0][1][0] = 2  whilst occupied[0][1][0] =  0
 6 Test empty_space[0][1][1] = 1  whilst occupied[0][1][1] =  0
 7 Test empty_space[0][1][2] = 1  whilst occupied[0][1][2] =  0
 8 Test empty_space[0][1][3] = 1  whilst occupied[0][1][3] =  0
 9 Test empty_space[0][2][0] = 2  whilst occupied[0][2][0] =  0
10 Test empty_space[0][2][1] = 2  whilst occupied[0][2][1] =  0
11 Test empty_space[0][2][2] = 2  whilst occupied[0][2][2] =  0
12 Test empty_space[0][2][3] = 1  whilst occupied[0][2][3] =  0
13 Test empty_space[0][3][0] = 1  whilst occupied[0][3][0] =  0
14 Test empty_space[0][3][1] = 1  whilst occupied[0][3][1] =  0
                              ][2] = 1  whilst occupied[0][3][2] =  0
                              ][3] = 1  whilst occupied[0][3][3] =  0
                              ][0] = 1  whilst occupied[1][0][0] =  0
                              ][1] = 0  whilst occupied[1][0][1] =  2
                              ][2] = 1  whilst occupied[1][0][2] =  0
                              ][3] = 1  whilst occupied[1][0][3] =  0
                              ][0] = 1  whilst occupied[1][1][0] =  0
                              ][1] = 0  whilst occupied[1][1][1] =  2
                              ][2] = 1  whilst occupied[1][1][2] =  0
                              ][3] = 1  whilst occupied[1][1][3] =  0
                              ][0] = 0  whilst occupied[1][2][0] =  3
                              ][1] = 0  whilst occupied[1][2][1] = -6
                              ][2] = 2  whilst occupied[1][2][2] =  0
                              ][3] = 1  whilst occupied[1][2][3] =  0
                              ][0] = 1  whilst occupied[1][3][0] =  0
                              ][1] = 1  whilst occupied[1][3][1] =  0
                              ][2] = 1  whilst occupied[1][3][2] =  0
                              ][3] = 1  whilst occupied[1][3][3] =  0
                              ][0] = 1  whilst occupied[2][0][0] =  0
                              ][1] = 0  whilst occupied[2][0][1] =  2
                              ][2] = 0  whilst occupied[2][0][2] =  2
                              ][3] = 1  whilst occupied[2][0][3] =  0
                              ][0] = 1  whilst occupied[2][1][0] =  0
                              ][1] = 0  whilst occupied[2][1][1] =  2
                              ][2] = 0  whilst occupied[2][1][2] =  2
                              ][3] = 1  whilst occupied[2][1][3] =  0
                              ][0] = 0  whilst occupied[2][2][0] =  3
                              ][1] = 0  whilst occupied[2][2][1] = -6
                              ][2] = 2  whilst occupied[2][2][2] =  2
                              ][3] = 1  whilst occupied[2][2][3] =  0
                              ][0] = 0  whilst occupied[2][3][0] =  3
                              ][1] = 0  whilst occupied[2][3][1] =  3
                              ][2] = 0  whilst occupied[2][3][2] = -6
                              ][3] = 1  whilst occupied[2][3][3] =  0
                              ][0] = 1  whilst occupied[3][0][0] =  0
                              ][1] = 0  whilst occupied[3][0][1] =  2
                              ][2] = 1  whilst occupied[3][0][2] =  0
                              ][3] = 1  whilst occupied[3][0][3] =  0
                              ][0] = 1  whilst occupied[3][1][0] =  0
                              ][1] = 0  whilst occupied[3][1][1] =  2
                              ][2] = 1  whilst occupied[3][1][2] =  0
                              ][3] = 1  whilst occupied[3][1][3] =  0
                              ][0] = 1  whilst occupied[3][2][0] =  0
                              ][1] = 0  whilst occupied[3][2][1] =  2
                              ][2] = 2  whilst occupied[3][2][2] =  0
                              ][3] = 1  whilst occupied[3][2][3] =  0
                              ][0] = 0  whilst occupied[3][3][0] =  3
                              ][1] = 0  whilst occupied[3][3][1] = -6
                              ][2] = 1  whilst occupied[3][3][2] =  0
                              ][3] = 1  whilst occupied[3][3][3] =  0
G:\FYPswAPI\swFYP2\sldworks_api_session\Debug>_
```

Case Study

- Retrieved SMG data from Feature Database
- Input new pump inlet information
- New pipe generated with SMG algorithm
  - Design revision
- Inter-temporal collaboration



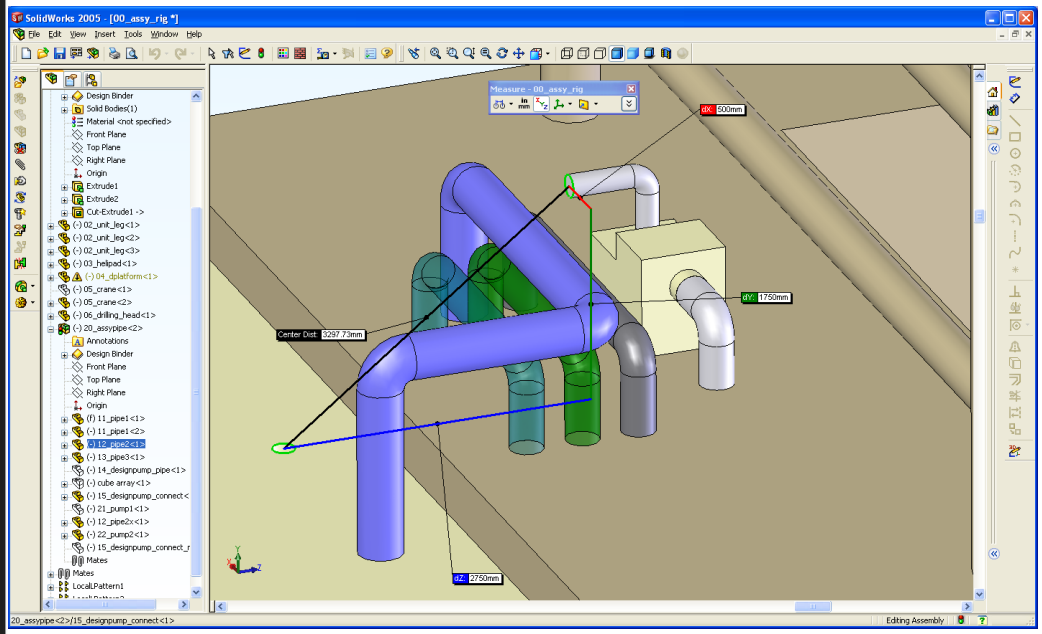
A case of pipeline design optimization in oil rig design

Mechanical Engineering

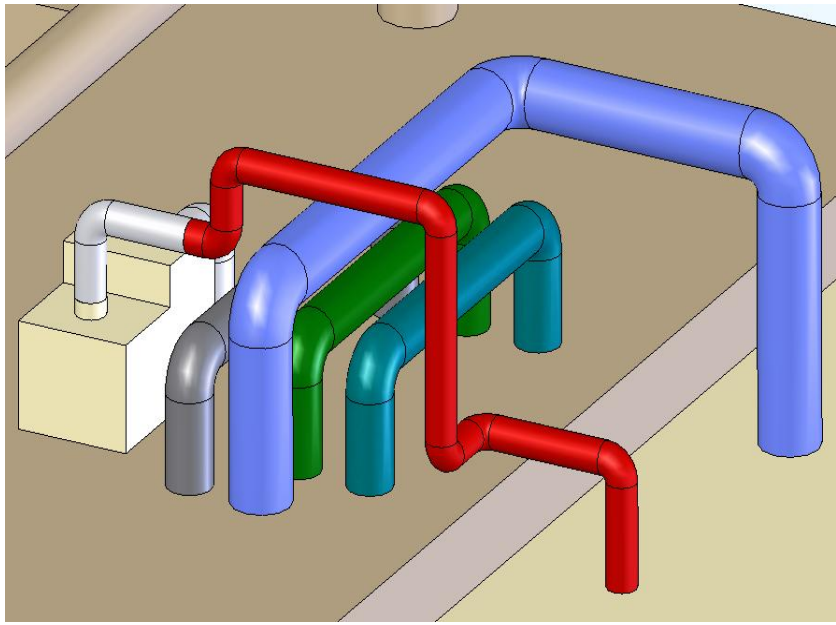
R1: '*the minimum crossing space between pipelines has to be more than x meters*',

R2: '*the use of lower space is prioritized for the ease of maintenance*',

R3: '*reserving maximum space for operational use*',

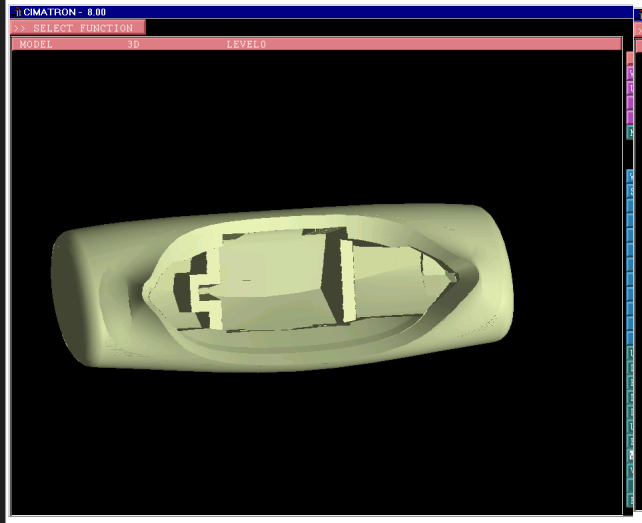


(a) A smart design environment with specified constraints

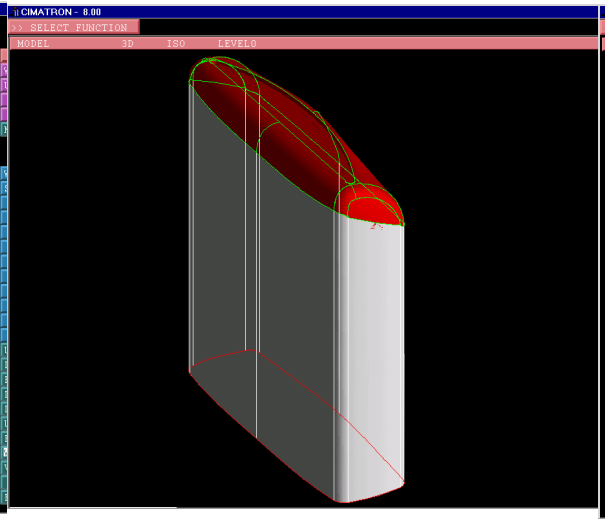


(b) A resulted piping path after optimization

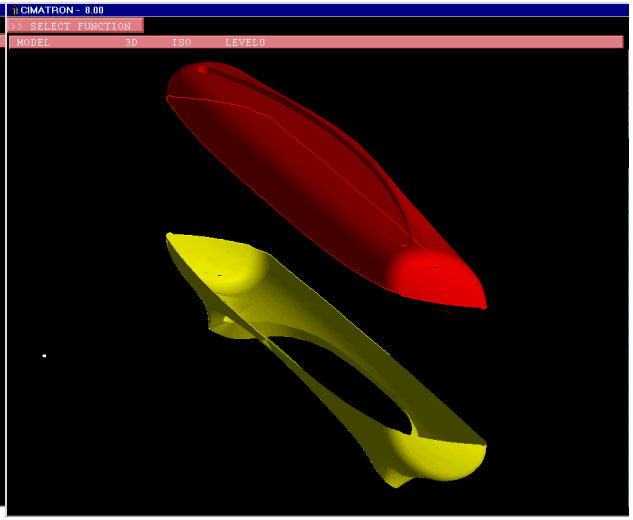
Sharing, deriving, reuse of feature information



(a) The rubber part model in association to the master product model and the metal frame of the iron rest



(b) Split faces for core and cavity inserts for rubber mold design



(c) Electrode model fro EDM machining

Feature Objects - Flexible Fine Grain Associative Features

Mechanical Engineering

- **Fine grain associative design and manufacturing**
- The speaker champions fine grain feature-based collaborative engineering.
- Fine grain association refers to the relations created or used for certain engineering purpose among engineering entities without the limitation of access, even to entities below 3D solid or part level.



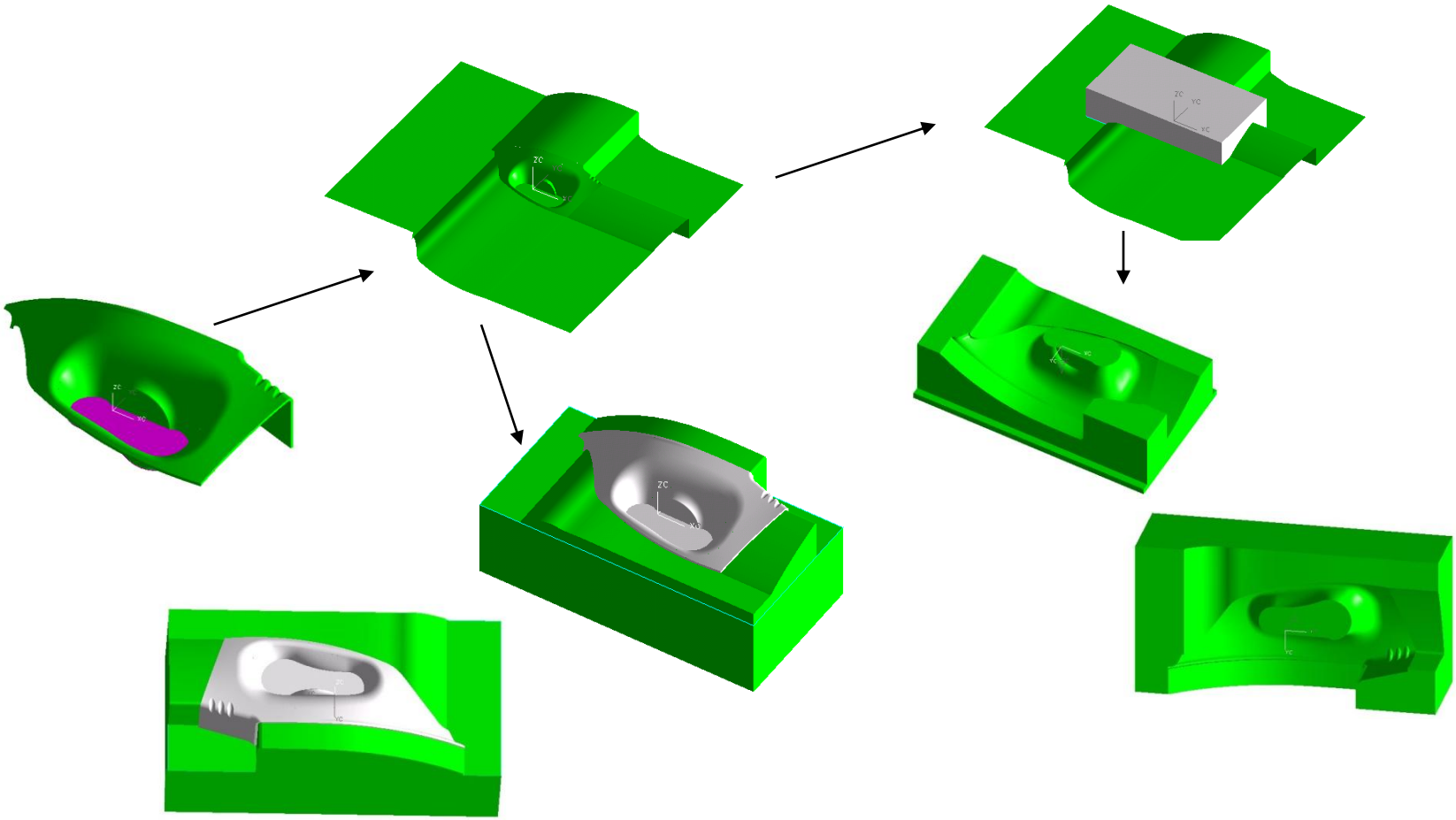

- Associative design and manufacturing support design automation and simulation for field engineering
- Associated computer aided solutions offer predictability and compressed delivery time. A systematic strategy helps industry to save millions of dollars.




Associative feature-based modeling approach

Mechanical Engineering

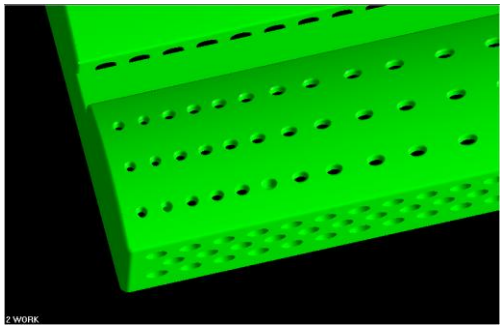
- Associative features and its modelling method – Parting features [Ma 2002]

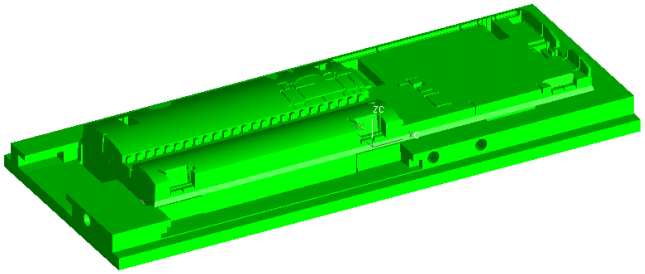


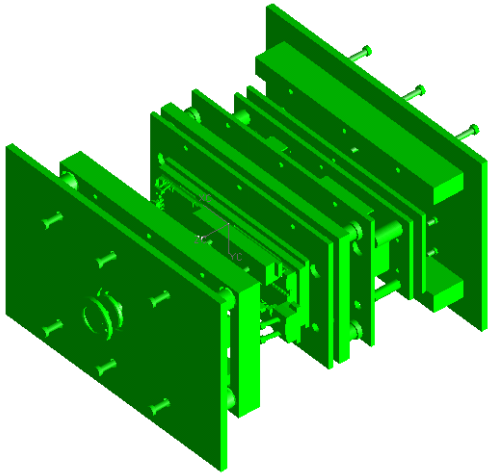
Unified feature-based modeling approach

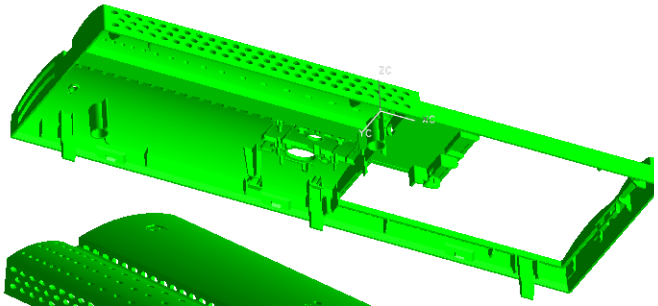
Mechanical Engineering

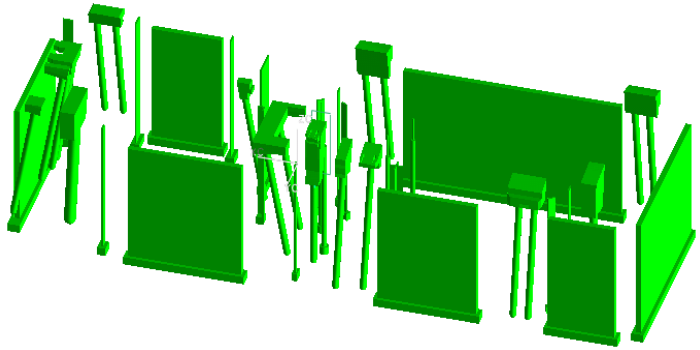
- Associative features and its modelling method – mould design [Ma 2002]

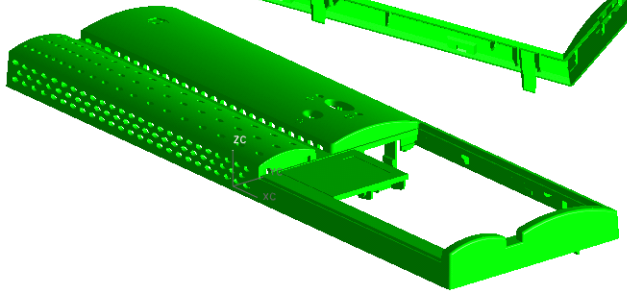


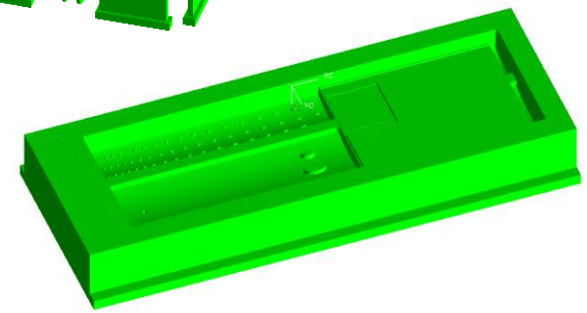








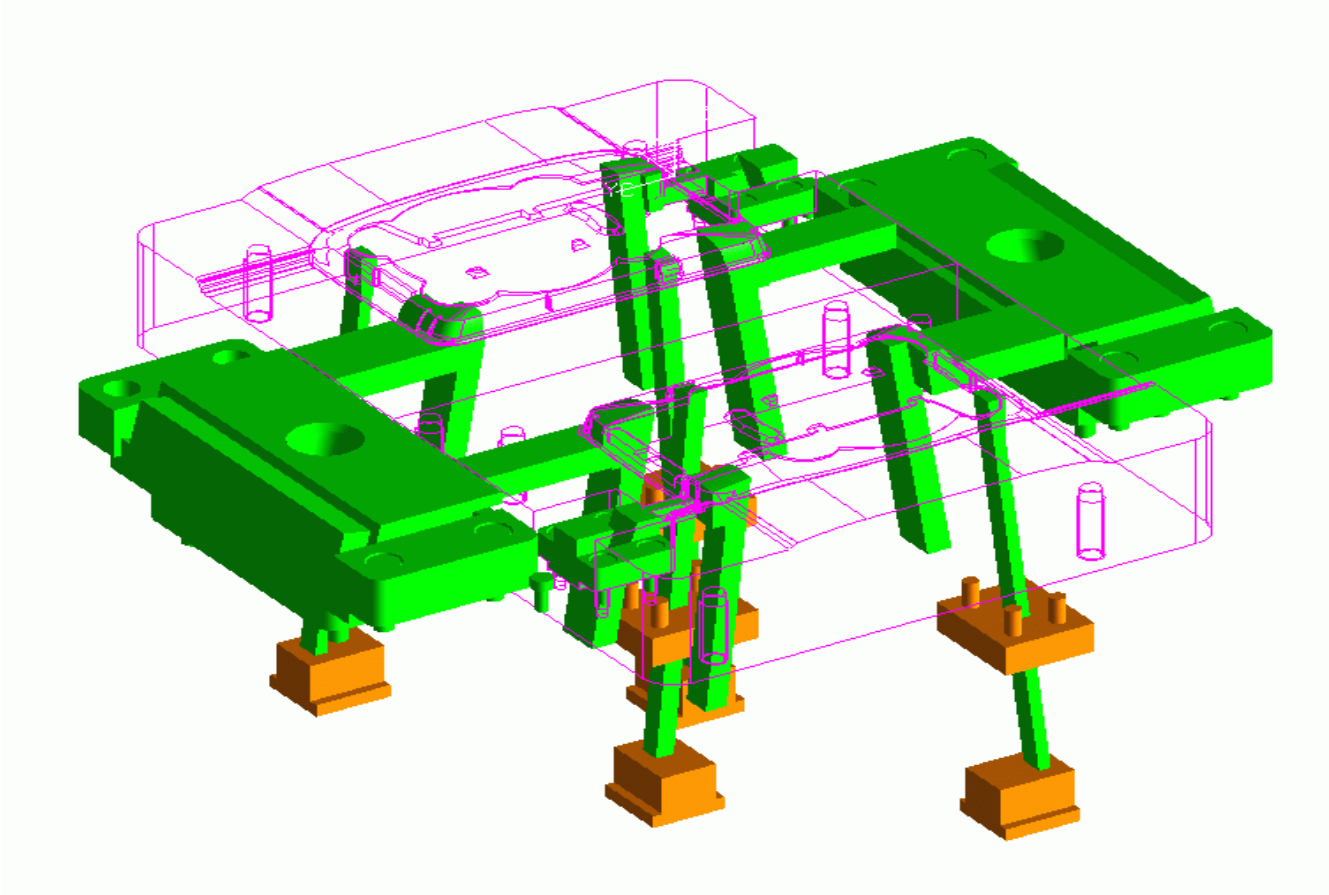




Feature information fusion

Mechanical Engineering

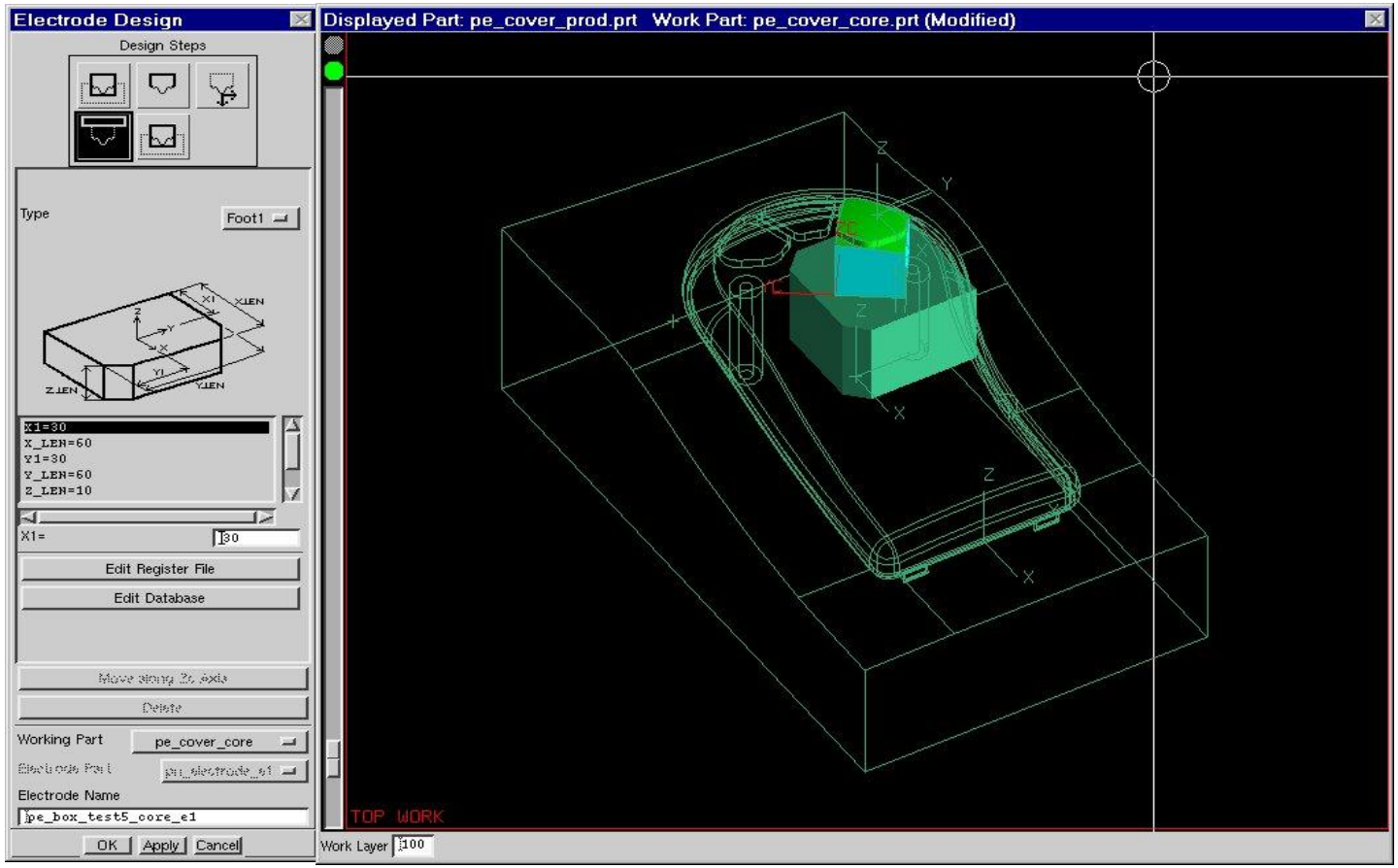
- Associative features and its modelling method – assembly design features



Unified feature-based modeling approach

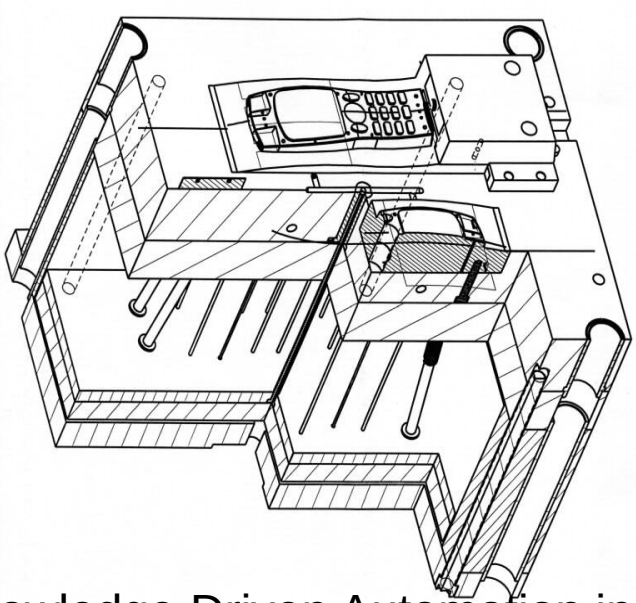
Mechanical Engineering

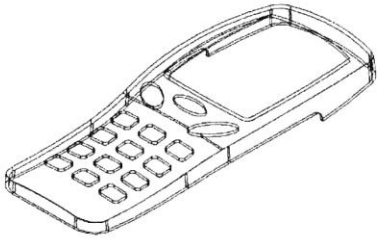
- Associative features and its modelling method – EDM Electrode [Ma 2001]



Associative Design and Manufacturing

Mechanical Engineering



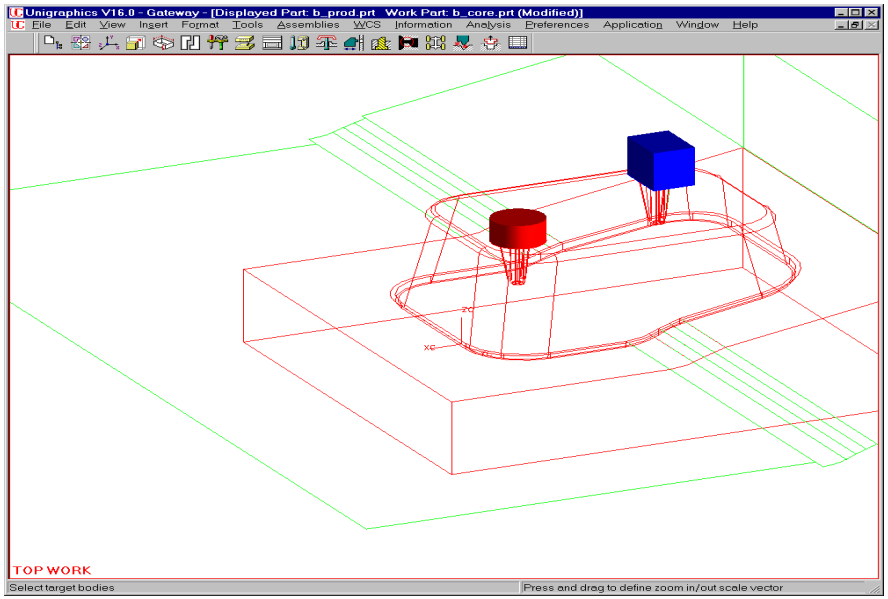


Knowledge Driven Automation in Productivity Improvement

Developing a unified and associative feature-based collaborative engineering platform is a challenge in engineering informatics.

Interoperability associated with multiple parties and systems is the focus point.

- A core-insert creation process in plastic mold design



Engineering Informatics for Intellectual Asset Management

Mechanical Engineering

- With the virtual engineering collaborations, globally-distributed designers and engineers at different stages of product lifecycles use different semantics and engineering patterns;
- yet they all work on a common product with different derived ‘views’ of their relevant working scopes.
- Contemperary interoperability can be described as the interpretability of data types and related information by different computer systems.

Source:http://www.plm.automation.siemens.com/en_us/products/by_lifecycle.shtml

- Consistency and change management are still challenges of research
- a new level of interoperability has to be investigated to create a solution for pervasive collaboration based on advanced engineering informatics and Web technologies.