Programming in Pro/ENGINEER Wildfire

- Interactive graphical programming is carried in Pro/E at two different levels.
  - At the higher level, C++ program are supported through Pro/ENGINEER API Toolkit.
  - At lower level, a micro programming environment, Pro/E PROGRAM Tool, is supported.
- These programming environments serve different needs.
- It is very typical for a CAD system to support at least these two levels of interactive graphics programming.

Pro/ENGINEER API Toolkit

- Pro/ENGINEER API Toolkit allows customers to extend, automate, and customize a wide range of Pro/ENGINEER design-through-manufacturing functionality.
- Pro/ENGINEER API Toolkit consists of a library of functions:
  - an application-programming interface (API), written in the C programming language. These functions are typically used by MIS organizations to create applications that run in parallel with Pro/ENGINEER and to integrate product information with the customer's corporate MRP/ERP systems.
  - applications used extensively by companies participating in PTC's Cooperative Software Partner (CSP) program to interface their commercial information management products with Pro/INTRALINK.
  - Normally, participation of a three day tutorial on the API Toolkit is needed to get the API Toolkit function module.
Typical Toolkit Applications

- The extensive Pro/ENGINEER API Toolkit provides programmatic access for creating, interrogating, and manipulating almost every aspect of the engineering model and its data management.
  - automating the creation of complex features
  - automating the production of Pro/ENGINEER deliverables, such as BOMs, drawings, and manufacturing operations
  - improving product quality by performing design rule verification based on inputs from an external, knowledge-based system.

Specific Pro/ENGINEER API Toolkit Function

- Customization of the Pro/ENGINEER-Foundation menu system
- Datum, solid, and manufacturing feature creation
- Assemblies
- Drawing automation
- Access to model geometry
**Pro/INTRALINK Access**

- The Pro/E API Toolkit provides complete access to the information within the Pro/INTRALINK environment, allowing customers to further leverage the product information contained within Pro/INTRALINK.
- Specifically, this functionality allows:
  - Integration with MRP/ERP Systems
  - Custom client applications, such as Web integrated clients
  - Triggered verification, notification and enforcement of business process actions

---

**Use of Pro/E API Toolkit**

**Product Capabilities:**
- Create automated, single-use or derived designs by geometric and parametric constraints
- Extend the Pro/ENGINEER user interface with custom processes seamlessly embedded in the interface
- Customize the Pro/ENGINEER menu system
- Collaborate between Pro/ENGINEER applications
- Access peer-to-peer communications for better application diagnoses

**Customer Benefits:**
- Integrate expert systems and knowledge-based applications into the Pro/ENGINEER environment.
- Improve product quality with design rule verification based on inputs from an external, knowledge-based systems.
The Pro/E Program environment, on the other hand, support quick and relatively straightforward interactive graphical programming in Pro/E for every users. The programming environment is simply Pro/E and Microsoft Notepad or Word. One can enter the Pro/E PROGRAM environment, by clicking Tools > Program... from the pull-down menu in the Pro/E PART or ASSEMBLY mode. To show or edit the program, one can click Show Design or Edit Design from the PROGRAM menu.

A Typical Pro/E PROGRAM Routine

Any of the followings:
- Input variables
- Relations
- IF-ELSE clauses
- Lists of all the features, and parts
- INTERACT statements
- MASSPROP statement

After the Pro/E PROGRAM routine is edited, the user will be asked whether the changes are to be incorporated (in the message window at the bottom). To proceed, enter Y. If N entered, the program will not be executed and changes will be lost.
Input Variables

- The INPUT statement must define the name and type of the variable. Variable names must always begin with a character. The following variable types are supported:
  - Number
    - String: This enables the user to enter parameters or model names.
    - Logical (YES_NO): Enter either Y or N.
  - An example:
    - INPUT THICKNESS NUMBER
      "Enter wall thickness for the cylinder"
      END INPUT

Relations

- All valid relations in a Pro/ENGINEER model can be entered in a Pro/PROGRAM.
- An example:
  - \( d0 = d6 \times 2 \)
- Here, \( d0 \) and \( d6 \) are dimension ID name.
**IF-ELSE Clauses**

- Conditional statements, i.e. IF _ ELSE, can be used to create a program branch.

  For example:
  ```plaintext
  ADD PROTRUSION.....
  IF d1 > d2
  ADD HOLE ...
  END ADD
  ENDIF
  ADD CUT.....
  END ADD
  ```

- So, when d1 is smaller than d2, a CUT is added, instead of a HOLE.

---

**Lists of Features and Parts**

- The program that Pro/E PROGRAM brings up simply includes all feature building commands used in creating the model and the properties of these features.

- All features and parts are listed in the program.

- For instance, the **ADD feature by EXTRUSION** operation is recorded as:
ADD FEATURE (initial number 8)
INTERNAL FEATURE ID 106
PARENTS = 100(#7)
PROTRUSION: Extrude

<table>
<thead>
<tr>
<th>NO.</th>
<th>ELEMENT NAME</th>
<th>INFO</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Feature Name</td>
<td>Defined</td>
</tr>
<tr>
<td>2</td>
<td>Extrude Feat type</td>
<td>Solid</td>
</tr>
<tr>
<td>3</td>
<td>Material</td>
<td>Add</td>
</tr>
<tr>
<td>4</td>
<td>Section</td>
<td>Defined</td>
</tr>
<tr>
<td>4.1</td>
<td>Reference Sketch</td>
<td>F7(SKETCH_2)</td>
</tr>
<tr>
<td>5</td>
<td>Feature Form</td>
<td>Solid</td>
</tr>
<tr>
<td>6</td>
<td>Direction</td>
<td>Side 2</td>
</tr>
<tr>
<td>7</td>
<td>Depth</td>
<td>Defined</td>
</tr>
<tr>
<td>7.1</td>
<td>Side One</td>
<td>Defined</td>
</tr>
<tr>
<td>7.1.1</td>
<td>Side One Depth</td>
<td>None</td>
</tr>
<tr>
<td>7.2</td>
<td>Side Two</td>
<td>Defined</td>
</tr>
<tr>
<td>7.2.1</td>
<td>Side Two Depth</td>
<td>Variable</td>
</tr>
<tr>
<td>7.2.2</td>
<td>Value</td>
<td>70.00</td>
</tr>
</tbody>
</table>

SECTION NAME = Sketch 2
FEATURE'S DIMENSIONS:
d11 = 70.00

END ADD

Additional operations can be added, and this ADD operation can be changed.

**INTERACT**

- INTERACT statements provide a placeholder for creating interactive part. They can be inserted anywhere within the FEATURE ADD - END ADD.

- Here is an example,
  ADD PROTRUSION......
  IF d1 > d2
  ADD HOLE.....
  ELSE
  INTERACT
  END IF
  ADD CUT.....

- In this example, an alternate set of features will be created if d1 is not greater than d2. The ADD CUT command has to be input by the user.
**MASSPROP**

- The MASSPROP statement is used to update mass properties each time geometry changes. Format is as follows:
  
  ```
  MASSPROP
  END MASSPROP
  ```

---

**Other Operations for Feature Editing**

- **Changing feature dimension**
  The dimensions of features in the program can be updated by a DIMENSION statement with:
    
    ```
    MODIFY d# = value
    ```

- **Editing Errors**
  Common editing errors include:
  - Having an IF statement without an END IF statement or vice versa
  - Typing a variable name incorrectly in a relation or a condition
  - Reordering a child before the parent
  - Deleting a parent feature
An Example Part and the PROGRAM Window

The Pro/Program for this Part Model
- Start Pro/E
- Open the Part Model file: part5.prt
- Use Pull Down Menu Tool > Program...

In the PROGRAM Window
- **Show Design** and **Edit Design** options will display the Pro/Program that is used to create the displayed part model.
- **Edit Design** option allows you to make changes to the model through “Programming Logic” rather than through “drawing and modeling”. Automated tasks can be achieved. If you exit from the Edit window and answer “Yes” in the message window at the bottom of the screen to the prompt: “Do you want to incorporate your changes into the model?” The programmed change will be added to the existing model. You can start from a simple template model to write various programs.
- The **J-Link function** allows you to load in Java codes.
The List of the Pro/E PROGRAM for this Part Model

<table>
<thead>
<tr>
<th>No.</th>
<th>Element Name Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Feature Name Defined</td>
</tr>
<tr>
<td>2</td>
<td>Constraint Defined</td>
</tr>
<tr>
<td>3</td>
<td>Constraint Defined</td>
</tr>
<tr>
<td>4</td>
<td>Feature #1 Defined</td>
</tr>
<tr>
<td>5</td>
<td>Point Defined</td>
</tr>
<tr>
<td>6</td>
<td>Point Type Defined</td>
</tr>
</tbody>
</table>

NAME = TOP

END ADD

ADD FEATURE (entity number 1)
INTERNAL FEATURE ID 1

DATUM PLANE

<table>
<thead>
<tr>
<th>No.</th>
<th>Element Name Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Feature Name Defined</td>
</tr>
<tr>
<td>2</td>
<td>Constraint Defined</td>
</tr>
<tr>
<td>3</td>
<td>Constraint Defined</td>
</tr>
<tr>
<td>4</td>
<td>Feature #1 Defined</td>
</tr>
<tr>
<td>5</td>
<td>Point Defined</td>
</tr>
<tr>
<td>6</td>
<td>Point Type Defined</td>
</tr>
</tbody>
</table>

NAME = RIGHT

END ADD

ADD FEATURE (entity number 3)
INTERNAL FEATURE ID 1

DATUM PLANE

<table>
<thead>
<tr>
<th>No.</th>
<th>Element Name Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Feature Name Defined</td>
</tr>
<tr>
<td>2</td>
<td>Constraint Defined</td>
</tr>
<tr>
<td>3</td>
<td>Constraint Defined</td>
</tr>
<tr>
<td>4</td>
<td>Feature #1 Defined</td>
</tr>
<tr>
<td>5</td>
<td>Point Defined</td>
</tr>
<tr>
<td>6</td>
<td>Point Type Defined</td>
</tr>
</tbody>
</table>

NAME = FRONT

END ADD

ADD FEATURE (entity number 4)
INTERNAL FEATURE ID 1

PARENTS = (1) (2) (4)

PROTRUSION: Extrude

---

<table>
<thead>
<tr>
<th>No.</th>
<th>Element Name Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Feature Name Defined</td>
</tr>
<tr>
<td>2</td>
<td>Extrude Feature type Solid</td>
</tr>
<tr>
<td>3</td>
<td>Material Defined</td>
</tr>
<tr>
<td>4</td>
<td>Section Defined</td>
</tr>
<tr>
<td>5</td>
<td>Sketch Defined</td>
</tr>
<tr>
<td>6</td>
<td>Feature Form Solid</td>
</tr>
<tr>
<td>7</td>
<td>Direction Sides</td>
</tr>
<tr>
<td>8</td>
<td>Depth Defined</td>
</tr>
<tr>
<td>9</td>
<td>Side One Defined</td>
</tr>
<tr>
<td>10</td>
<td>Side One Depth Defined</td>
</tr>
<tr>
<td>11</td>
<td>Side Two Defined</td>
</tr>
<tr>
<td>12</td>
<td>Side Two Depth Defined</td>
</tr>
<tr>
<td>13</td>
<td>Upsilon</td>
</tr>
</tbody>
</table>

NAME = BLOCK

SECTION NAME = 52D0001

FEATURES DIMENSIONS:

g1 = 20.00

END ADD

ADD FEATURE (entity number 5)
INTERNAL FEATURE ID 28
PARENTS = 7(4)

PROTRUSION: Extrude

---

<table>
<thead>
<tr>
<th>No.</th>
<th>Element Name Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Feature Name Defined</td>
</tr>
<tr>
<td>2</td>
<td>Extrude Feature type Solid</td>
</tr>
<tr>
<td>3</td>
<td>Material Defined</td>
</tr>
<tr>
<td>4</td>
<td>Section Defined</td>
</tr>
<tr>
<td>5</td>
<td>Sketch Defined</td>
</tr>
<tr>
<td>6</td>
<td>Feature Form Solid</td>
</tr>
<tr>
<td>7</td>
<td>Direction Sides</td>
</tr>
<tr>
<td>8</td>
<td>Depth Defined</td>
</tr>
<tr>
<td>9</td>
<td>Side One Defined</td>
</tr>
<tr>
<td>10</td>
<td>Side One Depth Defined</td>
</tr>
<tr>
<td>11</td>
<td>Side Two Defined</td>
</tr>
<tr>
<td>12</td>
<td>Side Two Depth Defined</td>
</tr>
<tr>
<td>13</td>
<td>Upsilon</td>
</tr>
<tr>
<td>14</td>
<td>Value</td>
</tr>
</tbody>
</table>

NAME = ROUND END

SECTION NAME = 52D0002

OPEN SECTION

FEATURES DIMENSIONS:

g1 = 5.00

END ADD

ADD FEATURE (entity number 6)
INTERNAL FEATURE ID 53
PARENTS = 7(4) 28(4)

CUT: Extrude
Program in AutoCAD

- Widely Used and Easy to Implement
- Powerful Interactive Graphics Programming Tools
  - AutoLISP
  - ADS – AutoCAD Development System
  - API – Advanced Programming Interface
  - Script Files
  - Menu Systems
- Extensive Tutorial Materials (@MECH410/520 web page)
Summary

- Interactive Graphics Programming is traditionally carried out using graphics routines in a special package.
- Today all CAD systems offer different levels of Interactive Graphics Programming capabilities.
- This unique capability of CAD systems allows them to be further developed into most convenient and more productive design tools.
  - Customization
  - User interface improvement (menu, etc.)
  - Repetitive and complex tasks
  - Guidelines for design
- It is essential not to limit one’s capability of using CAD system simply as a modeling program or drafting tool.