Agenda Connector Training

• **Introduction**
  • What are Connectors? Why to use them? How do they work?

• **New connector panel structure**
  • Where to create the connectors?
  • Final connection routines for
    • Spots / Seams
    • Bolts

• **Live demonstrations**
  • Connector Browser
  • Part Replacement (reconnect rules)
  • Spot, Seam, Bolt Mesh Imprint and resolve conflicts
  • Seam Partitioning
  • et. al
Connectors - What are they?

- They are geometric entities (not FE) primarily used to create spot- and seamwelds, but also used to create adhesives, bolts and masses.
- They are simply a database of information defining specific requests for connections at specific locations.
- They are a medium to create both solver independent and/or solver dependent connections between geometric and/or FE entities.
- They can be created interactively or from different master connection files (.xml, .mcf, .vip).
Connectors - Why to use them?

- **Connectors hold definition of:**
  - connection locations (spots, seams, adhesives, bolts)
  - linked entities
  - re-connect rules
  - et al.

- **Connectors manage realized FE type and entities like**
  - rigids, CWELDs, MAT100s, hexa adhesives, et al.

- **Allows switching between different FE weld types**

- **Allow quick part reconnection**
  - replacement by names or IDs
Connector – How do they work?

Initial situation:

<table>
<thead>
<tr>
<th>Entity Type</th>
<th>Entity ID</th>
<th>Entity Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link Entity</td>
<td>1</td>
<td>Comp</td>
</tr>
<tr>
<td>Link Entity</td>
<td>2</td>
<td>Comp</td>
</tr>
<tr>
<td>Connecter Location</td>
<td>10</td>
<td>Point</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Entity Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>“top”</td>
</tr>
<tr>
<td>“bottom”</td>
</tr>
</tbody>
</table>
Connector – How do they work?

Connector Definition:

**Link Entity**
- Entity Type: Comp
- Entity ID: 1
- Entity Name: “top”

**Unrealized Connector (yellow)**
- Entity Type: Connector
- Entity ID: 7
- Link entities: Comp 1, Comp 2
- Thickness: 2

**Link Entity**
- Entity Type: Comp
- Entity ID: 2
- Entity Name: “bottom”
Connector – How do they work?

Connector Realization:

**Link Entity**
- Entity Type: Comp
- Entity ID: 1
- Entity Name: “top”

**Realized Connector (green)**
- Entity Type: Connector
- Entity ID: 7
- Link entities: Comp 1, Comp 2
- Thickness: 2
- FE Config: 70 (acm detached)
- FE Tolerance: 5.0 (enough for projection)

**Link Entity**
- Entity Type: Comp
- Entity ID: 2
- Entity Name: “bottom”
Connector – How do they work?

Connector Realization:

**Link Entity**
- Entity Type: Comp
- Entity ID: 1
- Entity Name: “top”

**Realized Connector (green)**
- Entity Type: Connector
- Entity ID: 7
- Link entities: Comp 1, Comp 2
- Thickness: 2
- FE Config: 70 (acm detached)
- FE Tolerance: 5.0 (enough for projection)

**Connector created weld element**
- Entity Types: Elements (Hex8, RBE3)

**Link Entity**
- Entity Type: Comp
- Entity ID: 2
- Entity Name: “bottom”
Connector – How do they work?

Connector Realization:

**Link Entity**
- Entity Type: Comp
- Entity ID: 1
- Entity Name: “top”

**Realized Connector (green)**
- Entity Type: Connector
- Entity ID: 7
- Link entities: Comp 1, Comp 2
- Thickness: 2
- FE Config: 70 (acm detached)
- FE Tolerance: 5.0 (enough for projection)

**Connector created weld element**
- Entity Types: Elements (Hex8, RBE3)

**Link Entity**
- Entity Type: Comp
- Entity ID: 2
- Entity Name: “bottom”
Connector – How do they work?

- **There are three different connector states**
  - **Yellow** unrealized connector
    - new created or imported connector before the first realization
    - editing the connector definition (i.e. add or delete a link entity from the connector), the connector removes the welds it created, and reverts back to an unrealized state.
    - manual unrealization
  - **Green** successful realized connector
  - **Red** failed connector
    - If the creation of the FE representation is unsuccessful (due to low tolerance, insufficient link entities, etc.) the connector icon is displayed as failed (red).
Connector – How do they work?

- **Separating the FE representation from the connector definition comes with a lot of advantages**
  - Connector can easily be re-realized as a weld of a different configuration without redefining it.
  - If you edit the connector definition (i.e. add or delete a link entity from the connector), the connector removes the welds it created, and reverts back to an unrealized state.
  - If the creation of the FE representation is unsuccessful (due to low tolerance, insufficient link entities, etc.) the connector icon is displayed as failed (red).
- **An unrealized connector is yellow, a realized connector is green, and a failed connector is red.**
Connectors - Where to create them?

- **Spot, Bolt, Seam, Area Panels**
  - Main connector panels
  - They are subdivided in at least three subpanels, which behave very similar:
    1. subpanel for a **combined** connector creation and realization
    2. subpanel for a pure connector creation
    3. subpanel for a (re)-realization of already existing connectors
  - Whereas the edit or the partition subpanels are different or belong exclusively to a certain connector type the tree listed subpanels look and behave very similar. Three columns can be identified. Here marked with colored boxes.
Connectors - Where to create them?

• The **blue column** contains everything related to
  • connector creation
  • and link detection.

• **In this column are defined:**
  • location
  • link candidates
  • number of layers
  • tolerance

• **Dependent on the connector type (spot, bolt, seam, area) and the selected locations additional options are provided.**
Connectors - Where to create them?

- The **orange column** contains everything related to:
  - realization type
  - post script or
  - property assignment

- **In this column are defined:**
  - realization type specific definitions like certain dimensions or number of elements, etc.
  - which property the FE representation should get or if a certain property script should be used to manage those property assignments
Connectors - Where to create them?

- The **yellow column** contains everything related to the final connection to the link entities.
- There are different routines which describe in more detail how the connection between the FE representation and the link elements should be achieved.
  - The routines differ depending on the connector type; similar methods are offered for spots, seams and area connectors, but bolt connectors need completely different methods.
Final connection routines for Spots, Seams and (Areas)

- The connection routines are clearly structured.
- Not all realization types are allowed to be used with all routines.
- Mesh independent has to be used for realization types, where the final connection is realized per solver definition.
Examples for Spots – Adjust realization
Examples for Spots – Adjust mesh

Spotline
Examples for Seams – Adjust realization

- Connector fails, if testpoints use the same nearest nodes for the element creation. Therefore a larger spacing maybe senseful.
Examples for Seams – Adjust mesh

- Note, that remesh with quadseams does not lead necessarily to clear t-edges.
Final connection routines for Bolts

- The connection routines are clearly structured.
- Not all realization types are allowed to be used with all routines.
- For all mesh independent bolt realization types a cylinder is defined. No bolt hole detection is performed, because no holes are required. All nodes inside the cylinder are considered to be part of the bolt realization.
Examples for bolts – mesh independent
Examples for bolts – mesh dependent

Initial situation

Consider existing holes only
Examples for bolts – mesh dependent

Initial situation

Create hole, if non
Examples for bolts – mesh dependent

Initial situation

Use hole, if available
Examples for bolts – mesh dependent

Initial situation

Fill and remesh hole, if available
Examples for bolts – mesh dependent

Initial situation

Adjust hole position (2D)
Examples for bolts – mesh dependent

Initial situation

Adjust realization
Examples for bolts – mesh dependent

hole consideration cylinder

applicable diameter

inapplicable diameter
Examples for bolts – mesh dependent

Initial situation

Create hole diameter
Examples for bolts – mesh dependent

Initial situation

Create and adjust hole diameter
Examples for bolts – mesh dependent

Initial situation

Adjust hole diameter
Examples for bolts – mesh dependent

Initial situation

No hole connection diameter
Examples for bolts – mesh dependent

- All above shown principal bolt creation methods can be combined with
  - no. of nodes around hole
    - preserve
    - exact number
  - washer creation
    - factor radius
    - exact width
  - fill hole
    - pie pieces
    - certain pattern
Connector Browser

- Full interactive display control
  - link entity browser (upper part)
  - connector browser (lower part)
- Access to the main connector panels
  - spot
  - bolt
  - seam
  - area
  - apply mass
  - fe absorb
Connector Browser

Standard action buttons in the connector browser behave exactly like the action buttons in the model browser unless one of the view option toggle buttons is pressed down. Their style changes then.

The view option toggle buttons define exactly what should be shown, hidden or isolated during using the standard action buttons.

The advanced action buttons become active after the appropriate selection. Options are:

- Find Between Connectors With:
  - minimum two selected links
  - exact selected links
  - all selected links

- Find Twin Connectors With:
  - minimum two links
  - exact links
Innovation Intelligence®

1D connections
- nearest node
- various imprints
Spotweld nugget with shell coating
Spotweld nugget with shell coating

*acm (shell gap + coating)*

- This realization creates one hexa cluster per connectors and realizes a node to node connection to the linked shell meshes (shell coating). Different patterns are available. This is driven by the number of hexas.
- The appearance can be influenced via the diameter and the washer layer activation.
Spotweld with rigid spider and circle segments
Spotweld with rigid spider and circle segments

**pie (rigid spider)**

- This realization creates one circle mesh out of a certain number of segments per link (shell mesh). Each circle mesh gets stiffen by a rigid spider which middle nodes are joined by a further straight rigid element.
- The appearance can be influenced via the diameter, the number of segment elements and the washer layer activation.
2D connections
• various imprints
Lap weld with caps and quad transition
Lap weld with caps and quad transition

seam-quad (angled+capped+L)
- This realization type is intended to be used together with the quad transition option.

appearance influenced by
- pitchsize
- weld angle (0°-60°, default 60°)
- run-off angle (0°-45°, default 10°)
- cap angle (45°-90° default 75°)
- pattern type
T weld with caps and quad transition
Lap weld with caps and quad transition

seam-quad (angled+capped+T)
- This realization type is intended to be used together with the quad transition option.

appearance influenced by
- pitchsize
- weld angle (0°-60°, default 60°)
- weld height (default 5.0)
- run-off angle (0°-45°, default 10°)
- cap angle (45°-90° default 65°)
- pattern type
2D connections
• conflicting imprints
Conflicting imprints
Conflicting imprints – 1st strategy

Resolve conflicting imprints
• Smaller conflicts can be resolved automatically when realizing the connectors. This releases the overlapping elements and performs a normal remesh in that area. This is permitted as long as the overlapping area is smaller than half the regular quad transition element. However, if a conflict is too great it cannot be resolved in this fashion.
Conflicting imprints – 2nd strategy
Conflicting imprints – 2nd strategy
Conflicting imprints – 2nd strategy

**Edit – manual imprint**

In the second step, the conflicting elements can be manually modified with all of the functions that HyperMesh provides.

- Then the manual mesh imprint is performed.
2D connections
• bolt imprint
Partition of seam connectors into passed and failed pieces

Partition

- This functionality proves, if each test points has valid projections in the given tolerance. The connectors are sheared into failed and passed pieces and organized as predefined.
Partition of seam connectors into passed and failed pieces

Partition
• This functionality proves, if each test points has valid projections in the given tolerance. The connectors are sheared into failed and passed pieces and organized as predefined.
Partition of seam connectors into passed and failed pieces

Partition
• This functionality proves, if each test points has valid projections in the given tolerance. The connectors are sheared into failed and passed pieces and organized as predefined.
Partition of seam connectors into passed and failed pieces

Partition
• This functionality proves, if each test points has valid projections in the given tolerance. The connectors are sheared into failed and passed pieces and organized as predefined.
Partition of seam connectors into passed and failed pieces

Partition

• This functionality proves, if each test points has valid projections in the given tolerance. The connectors are sheared into failed and passed pieces and organized as predefined.
Partition of seam connectors into passed and failed pieces

Partition

- This functionality proves, if each test points has valid projections in the given tolerance. The connectors are sheared into failed and passed pieces and organized as predefined.
Hexa Positioning

- Thickness and offset consideration
- Thickness consideration only
- Mid thickness
- Const. thickness and \((t+\ell_0)/2\)
- Thickness consideration only
- Const. thickness and \((t+\ell_0)/2\)
Hexa Positioning

- All “gap“ realizations orientate on the exact shell position.
- Between new and former hexa positioning can be decided in the general connector options.
Thank you!