Smart 3D
Codeless Interference Rules

Prasad Mantraratnam

Smart3D Automation Team Lead & Product Owner

INTERGRAPH®

Codeless Interference Checking Rules

Typical Interference Checking Rule customization

- **Suppress certain objects from Clash Checking**
  - E.g.,– Test systems, Roads, Grade level, Huge objects by Range …
- **Suppress known false clash situations**
  - Piping parts clashing with Welds (maintenance aspect).
  - Slab penetrating objects (small bore pipe, steel etc).
  - Clashes within a given tolerance to member part ends.
  - Between objects in “Testing” system hierarchies.
  - Clashes within Imported Steel.
- **Categorize clashes**
  - as Hard/Soft/Ignore based on object type and aspect.

Interference Checking Rules - Current Scenario

Current Scenario - challenges

- Customizing Interference Checking rules requires programming knowhow.
- Need expertise in Smart3D's data model and VB6 programming model.
- Trivial in some cases, but can be quite involved in most cases.
- Out of the box rule may not suffice for everyone.

Codeless Interference Checking Rules Functionality

- Solves the above challenges and provides simple means.

Other Administration benefits

- Different rules for different plants, even though sharing same catalog.
- Clear log explaining which rule caused to ignore an object / clash.
- Ease of testing – Local IFC picks up changes to config file instantly.
- Very fine control on what rules act on – limit few rules to few object types etc.
- Rules can be combined.
- Admin can test new rules on live data set without disturbing ongoing ServerIFC or LocalIFC of other users.
- Full control on Restart / Warm-reboot options for your changes.
- Provides flexibility to even make code-based decisions.

New Functionality - Codeless Interference Checking Rules

- Available on eCustomer website – includes CodelessIFCRule.DLL, sample rule (eqvt to OOTB IFCRule.dll) and associated documentation.
- Facilitates writing Interference Checking Rules using **NO** code.
  - Driven by a “config” file (xml) - contains rules.
    - Rules in simple english text … easily interpretable – fully documented.
  - Several features to configure Pre / Post Processing Rules
    - Objects to Ignore for Clash checking
    - (false) Clashes to Ignore
    - Categorize Clashes (set Clash Type)
    - …
  - User extensible … user can add/modify rules entries as needed.
  - Easy to test and deploy.

- Further extendable, if required, by user’s custom “code based rules”.
- Further extendable, by Intergraph, to add more features without affecting user’s existing deployment of this solution.

Codeless Interference Checking Rules - Features

Ignore Objects For Clash Checking

(pre-processing – decide if object is to be checked for interference or not)

- ByObjectPG → e.g., ignore objects in GHOST PG.

```
<IgnoreObjectsForClashChecking>
    <ByObjectPG List="Testing,GHOST"
        Comment="Ignore all objects in Testing and GHOST PGs"/>
```

- ByObjectType → e.g., ignore assembly connections.

```
<ByObjectType List="Assembly Connections"
    Comment="Ignore all Assembly Connections"/>
```

- ByName
- BySystemPath
- ByParentSystemPath → e.g., ignore objects within test systems

```
<ByParentSystemPath Like="*\Testing\*"
    Comment="Ignore Objects under system(s) named 'Testing'"/>
```

- ByInterfaces
- ByAttribute(s) → Also possible to check attributes of related objects
- ByFilePath – For Reference Objects.
- ByRuleCombination → By a combination of above rules to satisfy.

Codeless Interference Checking Rules - Features

Ignore Clashes Between Objects

(post-processing – analyze a clash and suppress it if desired)

- ByObjectPG
- ByObjectType – e.g., suppress clash between welds/piping

```
<ByObjectType List1="Piping Welds"
              List2="Piping Components, Pipes, Piping Instruments, Piping Specialty Items"
```

- ByName …… BySystemPath …… ByParentSystemPath
- ByInterfaces …… ByAttribute …… (also related object attribute)

```
<ByAttribute Interface1="IJStructEDIData" Attribute1="OriginAppName" Operator1="EQ" Value1="Tekla"
             Interface2="IJStructEDIData" Attribute2="OriginAppName" Operator2="EQ" Value2="Tekla"
             Comment="Ignore clashes within steel imported from Tekla"/>

<ByAttribute Interface1="IJConstructionInfo" Attribute1="ConstructionType" Operator1="EQ" Value1=":17"
             Interface2="IJConstructionInfo" Attribute2="ConstructionType" Operator2="EQ" Value2=":17"
             Comment="Ignore clashes between 'existing, to be reused in place' objects"/>
```

- ByCommonAttributeValues – e.g., stick built stairs/ladders
- Interconnected – e.g., branch pipe off olet clashing header pipe Insulation
- SlabPenetration – e.g., small bore pipe penetrating slab without holes modeled
- WithinToleranceToMemberPartEnd – e.g., clash within tolerance to member end pt.

Codeless Interference Checking Rules - Features

Set Clash Type

(post-processing – categorize a clash)

- This allows you to categorize clashes as **Hard**, **Soft** and **Ignore** based on objects and the clashing aspects.

ByDefiningInterfacesAndAspect

```
<ByDefiningInterfacesAndAspect
   DefiningInterfacesList1="IJRtePiping"  Aspect1="Simple Physical"
   DefiningInterfacesList2="IJEquipmentFurnishings"    Aspect2="Maintenance“
  Type = "SOFT"/>
<ByDefiningInterfacesAndAspect
   DefiningInterfacesList1 = "IJRtePiping" Aspect1="*"
   DefiningInterfacesList2 ="*" Aspect2="Operation“
   Type="HARD"/>
```

Wild cards allowed wherever appropriate to condense several rules into one. Also can use LimitToObjectTypes option.

Codeless Interference Checking Rules - Deployment

- **Deploy to SymbolShare** → SymbolShare\Custom Symbols\ClashMgmt\
- **Run Update Custom Symbol Configuration** from Project Management.
- **Create configuration file for your Plant –** Copy example and edit to your requirements.
- **Bulkload CodelessIFCRule.xls** (to switch from existing IFCRule.dll to CodelessIFCRule.dll)
- **Test and fine-tune your rules** - Logging mechanism provides verbose info. LocalIFC detects changes to Configuration File immediately.

```
@10/2/2012 7:47:14 PM - ProcessObject ? [Assembly Connections] - 'GussetPlateAsmConn_1-1-0032'
   --> Satisfied - /CustomClashRules/IgnoreObjectsForClashChecking/ByObjectType[0] -->
   <ByObjectType List="Assembly Connections" Comment="Ignore all Assembly Connections"/>
ProcessObject -> No
```

```
@10/2/2012 7:48:25 PM - CreateInterference ? - [Piping Welds] - '402-P' & [Piping Components]
'Flange-0213'
   --> Satisfied - /CustomClashRules/IgnoreClashesBetweenObjects/ByObjectType[0] -->
   <ByObjectType List1="Piping Welds" List2="Piping Components, Pipes, Piping Instruments,
   Piping Specialty Items" Comment="Ignore clashes between Piping Welds and Piping parts"/>
CreateInterference -> No
```

- **Deploy in Production**

Codeless Interference Rules – Iterative Refinement