Center for e-Design

Validation of CAD Model Exchange

May, 2009

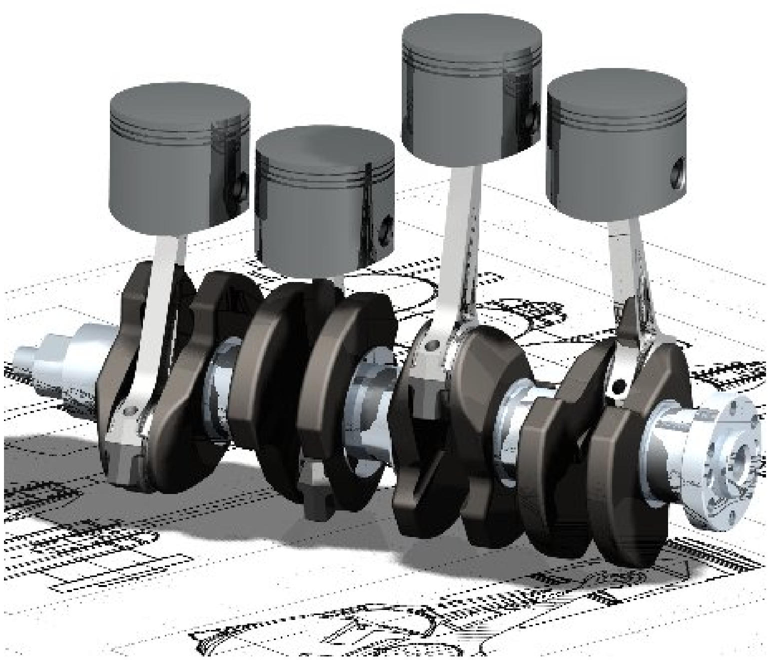
Presented by: John Altidor

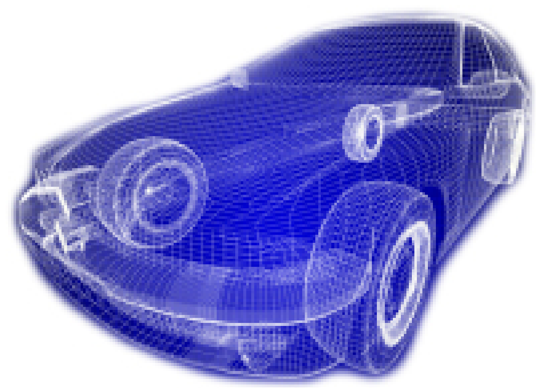
CAD Systems

- CAD (Computer-Aided Design) systems are software systems for producing computer models of objects.

- This research focuses on CAD systems that produce models of **physical** or **geometric** objects.

- CAD systems allow rigorous design and testing of objects **before physical construction**, which has enormous economic benefits.





CAD Model Construction

- CAD models are constructed by a sequence of operations performed on geometric primitives.
- CAD models are not only a single image or a polygonal mesh.

(Loading pics/Cad_fm01s.mov)

What is Interoperability?

- The ability of diverse systems, organizations, and other domains to work together (inter-operate).

- Examples of interoperability:
  - Programming Language Interoperability:
  - CAD System Interoperability:

What is Interoperability?

- The ability of diverse systems, organizations, and other domains to work together (inter-operate).

- Examples of interoperability:
  - Programming Language Interoperability:
    - Calling Java function from C++ (or vise-versa).
  - CAD System Interoperability:

What is Interoperability?

- The ability of diverse systems, organizations, and other domains to work together (inter-operate).

- Examples of interoperability:
  - Programming Language Interoperability:
    - Calling Java function from C++ (or vise-versa).
  - CAD System Interoperability:
    - Converting a model from one CAD system (SolidWorks) into another CAD system (SolidEdge).

Why is Data Exchange Important?

<u>30GB iPod:</u>

Linden, G., Kraemer, K.L., Dedrick, J., (2007). *Who Captures Value in a Global Innovation System? The case of Apple's iPod.*

Why is Data Exchange Important?

<u>30GB iPod:</u>

- hard drive
  - Toshiba (HQ- Japan, China)

Linden, G., Kraemer, K.L., Dedrick, J., (2007). *Who Captures Value in a Global Innovation System? The case of Apple's iPod.*

Why is Data Exchange Important?

30GB iPod:

- hard drive
  - Toshiba (HQ- Japan, China)
- video multimedia processor
  - Broadcom (HQ- US, Taiwan and Singapore)

Linden, G., Kraemer, K.L., Dedrick, J., (2007). *Who Captures Value in a Global Innovation System? The case of Apple's iPod.*

Why is Data Exchange Important?

30GB iPod:

- hard drive
  - Toshiba (HQ- Japan, China)
- video multimedia processor
  - Broadcom (HQ- US, Taiwan and Singapore)
- display driver
  - Renesas (Japan)

Linden, G., Kraemer, K.L., Dedrick, J., (2007). *Who Captures Value in a Global Innovation System? The case of Apple's iPod.*

Why is Data Exchange Important?

30GB iPod:

- hard drive
  - Toshiba (HQ- Japan, China)
- video multimedia processor
  - Broadcom (HQ- US, Taiwan and Singapore)
- display driver
  - Renesas (Japan)
- Mobile SDRAM
  - Samsung (Korea)

Linden, G., Kraemer, K.L., Dedrick, J., (2007). *Who Captures Value in a Global Innovation System? The case of Apple's iPod.*

Why is Data Exchange Important?

Over 400 parts

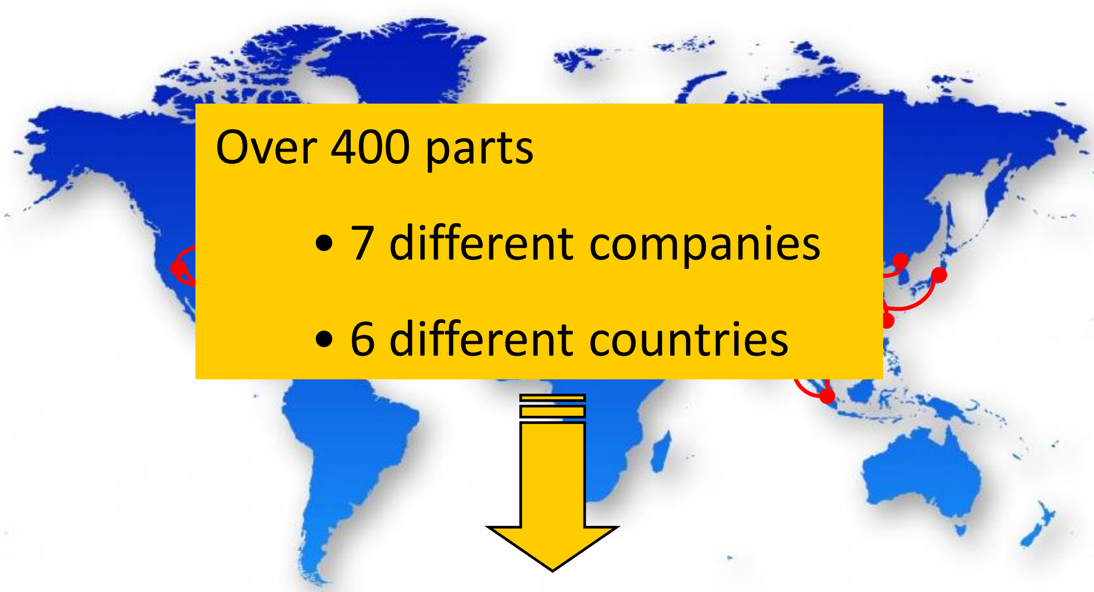
- 7 different companies
- 6 different countries

<u>30GB iPod:</u>

- hard drive
  - Toshiba (HQ- Japan, China)
- video multimedia processor
  - Broadcom (HQ- US, Taiwan and Singapore)
- display driver
  - Renesas (Japan)
- Mobile SDRAM
  - Samsung (Korea)

Linden, G., Kraemer, K.L., Dedrick, J., (2007). *Who Captures Value in a Global Innovation System? The case of Apple's iPod.*

Why is Data Exchange Important?



Over 400 parts

- 7 different companies
- 6 different countries

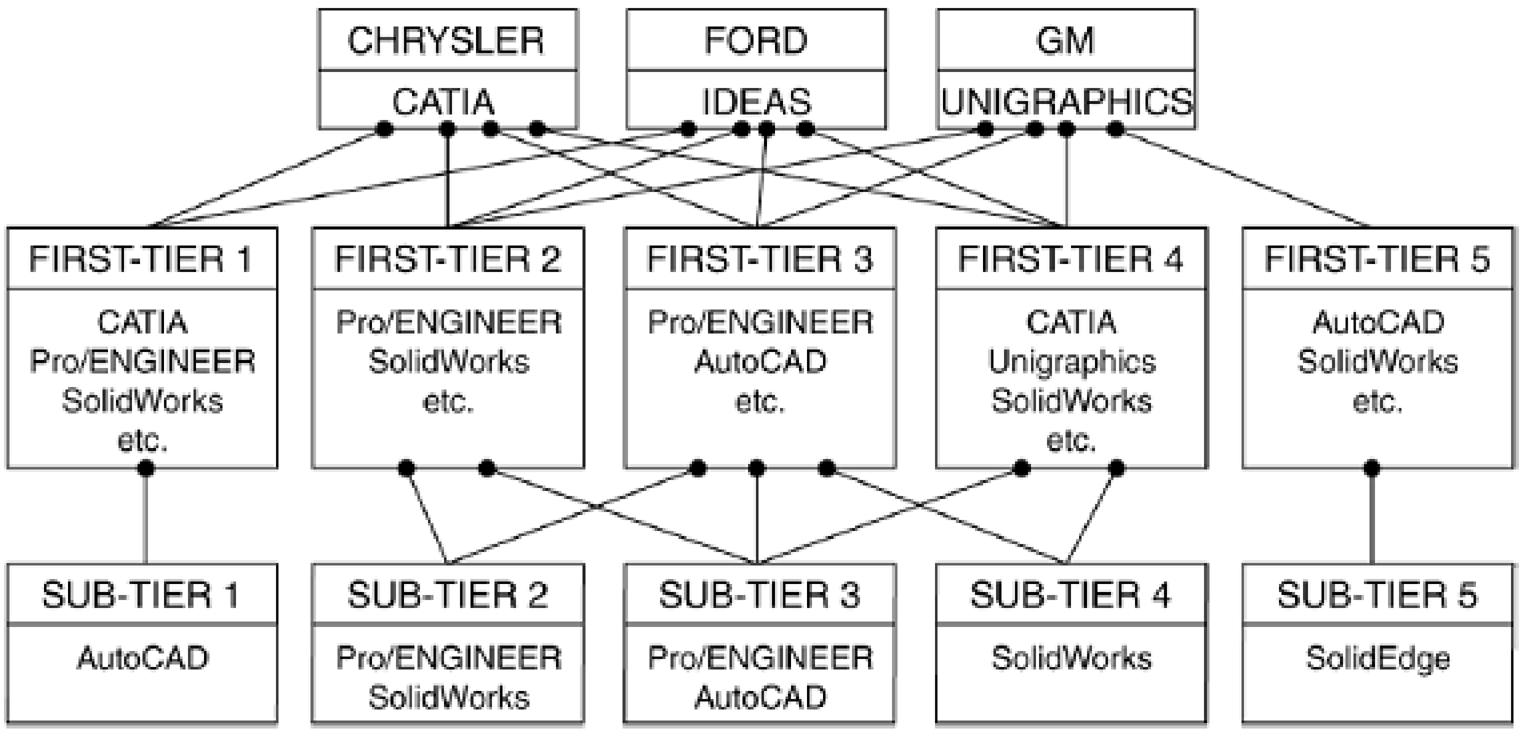
Effective communication between partners is paramount

<u>30GB iPod:</u>

- hard drive
  - Toshiba (HQ- Japan, China)
- video multimedia processor
  - Broadcom (HQ- US, Taiwan and Singapore)
- display driver
  - Renesas (Japan)
- Mobile SDRAM
  - Samsung (Korea)

Linden, G., Kraemer, K.L., Dedrick, J., (2007). *Who Captures Value in a Global Innovation System? The case of Apple's iPod.*

Typical OEM Supply Chain of CAD System Interoperation



Geometric Representations

- Traditionally, CAD models are almost entirely represented by geometric information.

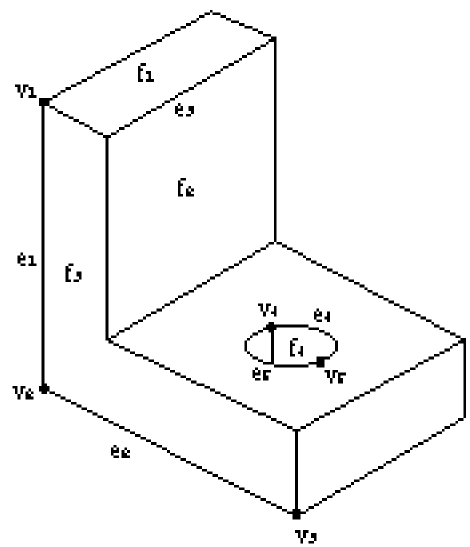
- Example: Boundary Representation: CAD Model is represented by a data structure giving information about each of the object's faces, edges and vertices and how they are joined together.

- Boundary representation models are composed of two parts: **topology** and **geometry**.

- **Topology:** records the connectivity of the faces, edges and vertices by means of pointers in the data structure.

- **Geometry:** describes the exact shape and position of each of the edges, faces and vertices.

Geometric Representations: Example



Standard for the Exchange of Product Model Data (STEP)

- Standard file format for representing CAD models.
- Only contains geometric representation of CAD models (e.g. B-REP). Example STEP file segment for SolidWorks:

```
#44 = FILL_AREA_STYLE_COLOUR ( '', #45 ) ;
#45 = COLOUR_RGB ( '',0.79, 0.82, 0.93 ) ;
#46 = CARTESIAN_POINT ( 'NONE',  ( 33.54, 37.17, 57.99 ) ) ;
#47 = DIRECTION ( 'NONE',  ( 0.00, 1.00, 0.00 ) ) ;
#48 = VECTOR ( 'NONE', #47, 1000.00 ) ;
```

- Although geometric information contains enough info to reconstruct the model as a whole, it does not preserve:

1. Model construction history
2. Design intentions
3. Semantics of the components that make up CAD models.

CAD Model Features

- A mechanism for capturing meta-data about CAD models such as design intentions and history.

- A **feature** is a higher order CAD entity that abstracts over the underlying geometry of a component in the model.

- Example: Given a 3D brick with a hole, the hole is considered a feature in the brick to reflect the manufacturing process used to create it, rather than referring to the hole by the mathematical term cylinder.

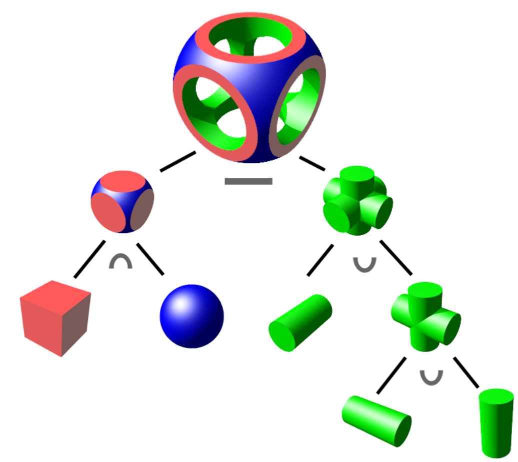
- Features have **attributes** for defining properties of a feature.

- Similar to a class in an OO language that abstracts over the underlying implementation of a component in a software system.

Feature-based representations

Many research efforts on feature form representation:

In ASU Features Testbed Modeler, features are defined in terms of parameters and rules about geometric shape. Interaction between features includes spatial relationship, volume-based constructive solid geometry (CSG) tree and Boolean operations.



A macro parametric approach was proposed to provide capabilities to exchange parametric information. This approach is to translate the macro recorded in the source system to a neutral macro format, and then translate it to the macro of the target system.

Problems with alternative feature representations

- Although the previously mentioned feature representations capture more meta-data about CAD models than geometric representation, they are not based off formal foundations.

- Without a formal basis, there are no guarantees on the correctness of their model conversions.

- Errors can arise during the model conversion process.

Problems during CAD Model Exchange

- Loss of information during CAD Model Conversion.

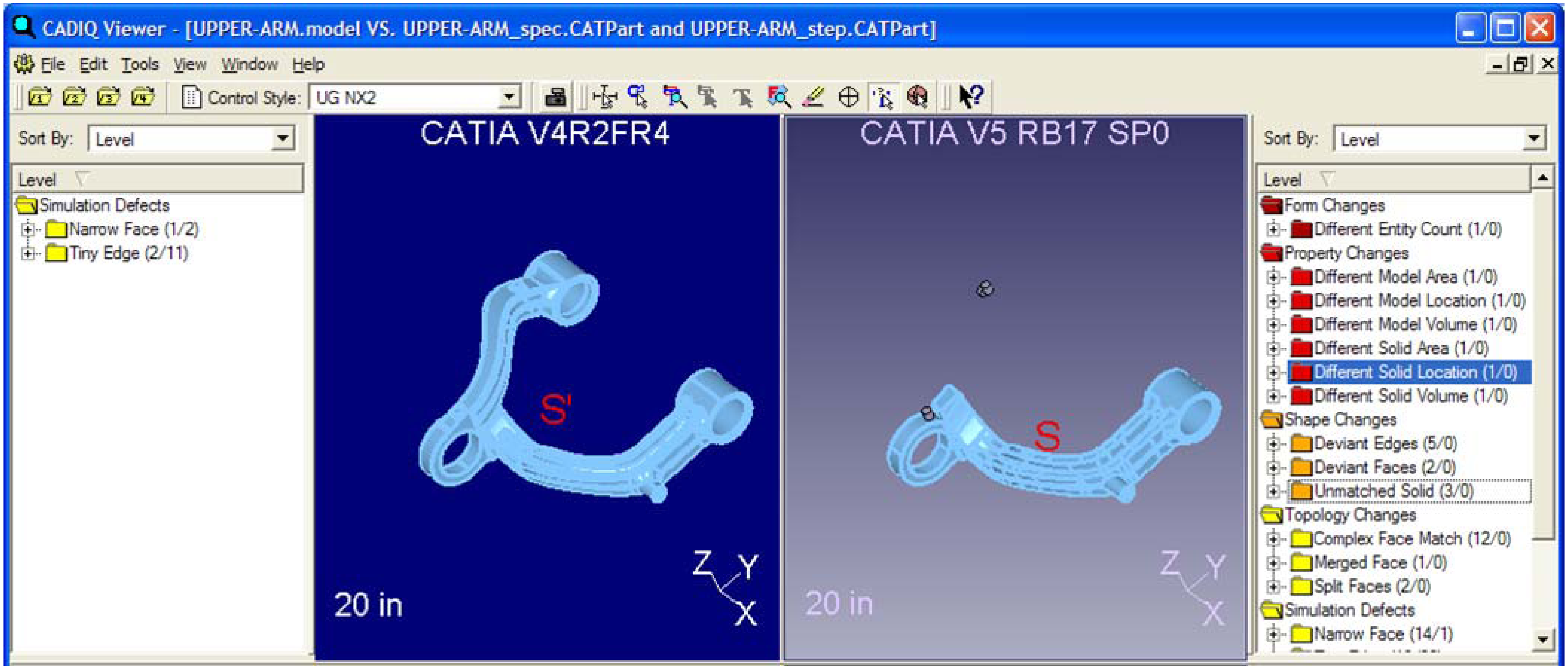


Photo from: Doug Cheney. 3D CAD Model Validation. **3D Collaboration & Interoperability '08**. Denver, CO. May 2008.

Problems during CAD Model Exchange cont.

- Distorting engineering design.

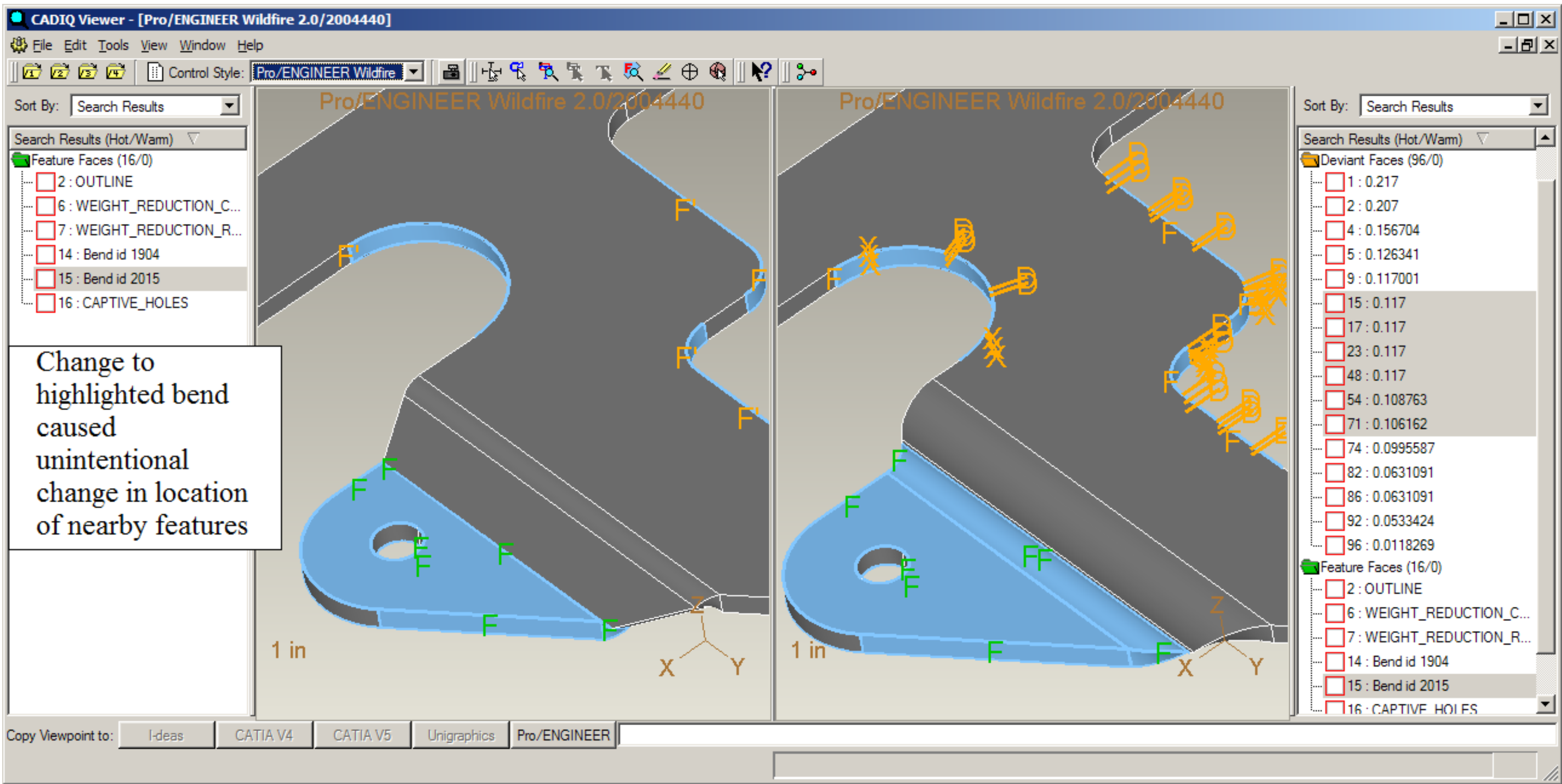


Photo from: Doug Cheney. 3D CAD Model Validation. **3D Collaboration & Interoperability ’08**. Denver, CO. May 2008.

Underyling Causes of Problems

- Human errors from tedious conversion process due to lack of automation.

- Information to preserve is not formally defined.

- No formal definition of what a **valid** CAD model exchange is.

Our Solution - A Semantic Representation of CAD Model

- CAD Models are represented in RDF (Resource Description Framework) as a semantic graph.

- RDF is built on top of the XML syntax which enables syntax-level interoperability.

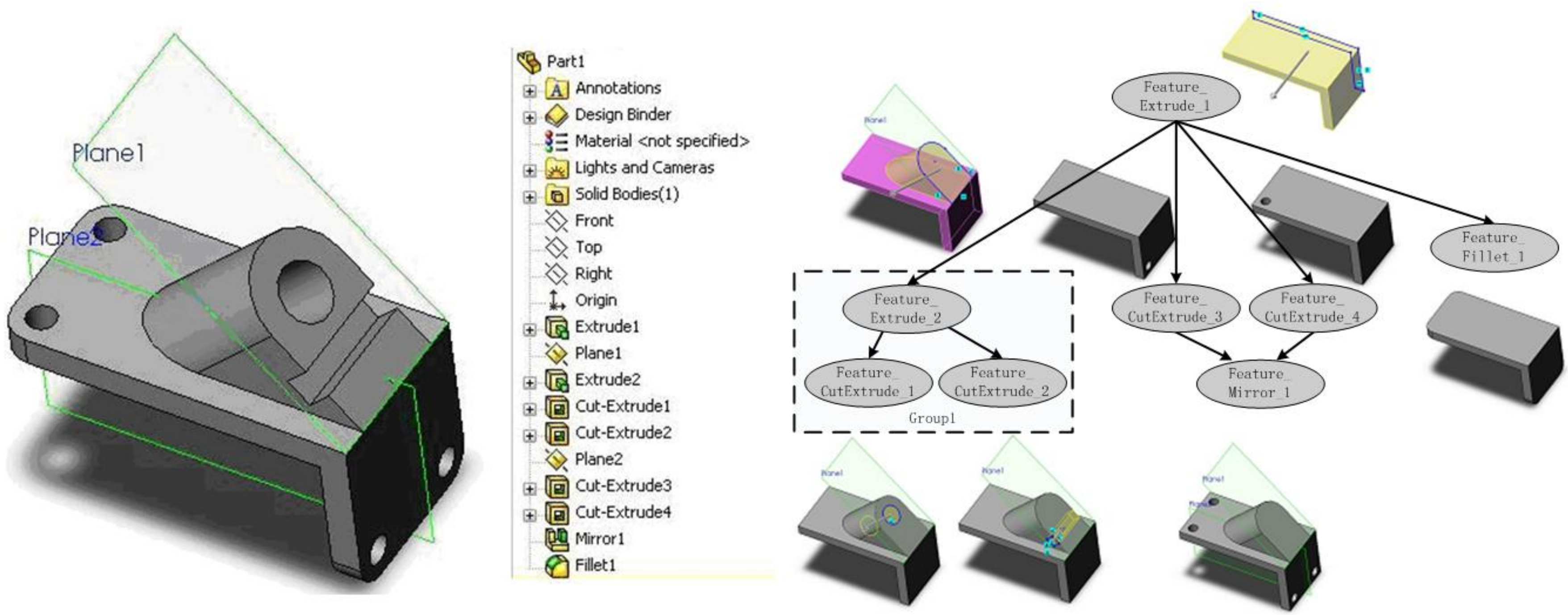
- RDF is part of the semantic web framework, which has mechanisms for semantic-level interoperability.

- Our feature-based approach differs from others by focusing on the **semantics** of CAD features.

- We utilize formal methods for analyzing interoperability and automated verification of CAD model exchange that lead to safety guarantees of our model conversions.

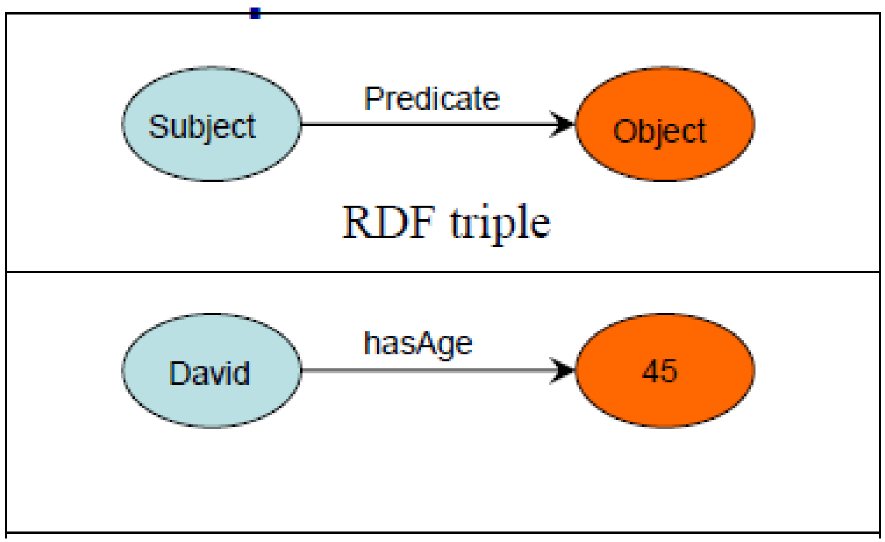
CAD Model Representation

- CAD features and operations are nodes in the graph.
- CAD model graphs represent combinations of and relationships between features in the model.



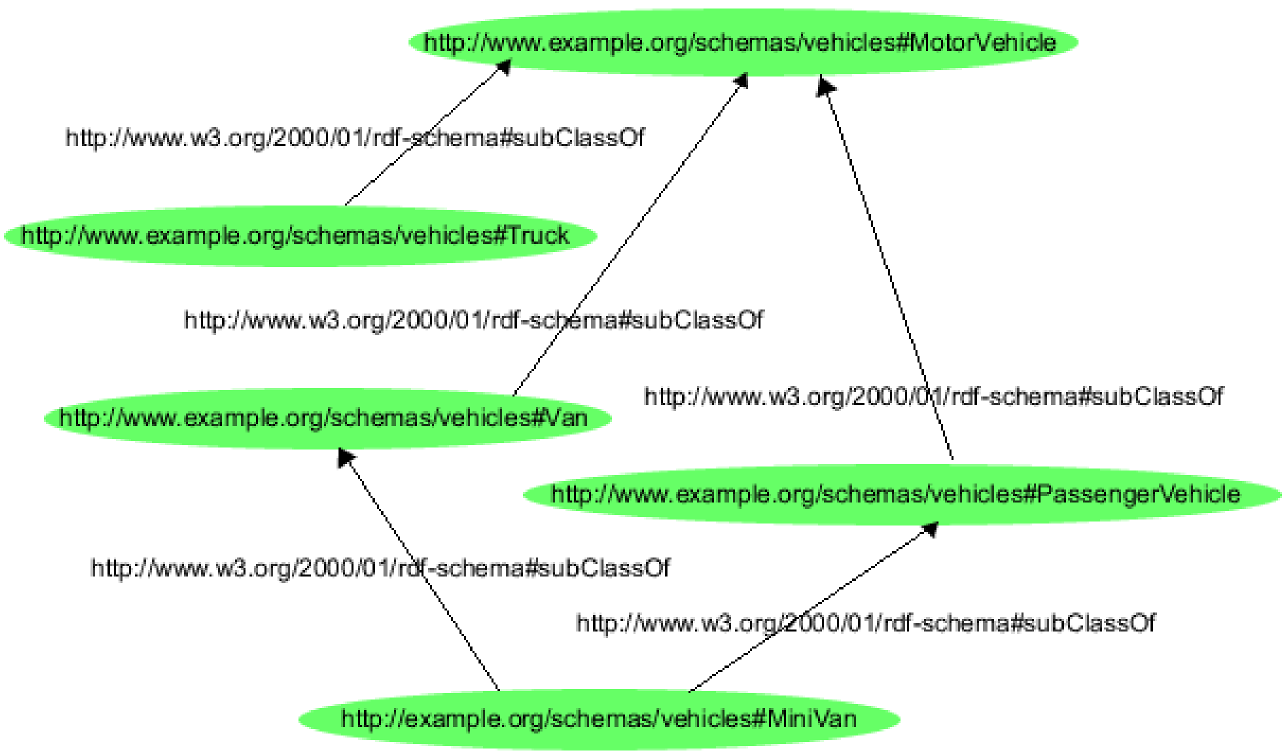
RDF Graphs

- RDF describes a graph of statements about **resources** (nodes).
- Each statement is an **subject-predicate-object** triple.
  - The subject is the **resource** from which the arc leaves.
  - The predicate is the **property** that labels the arc.
  - The object is the **resource or literal** pointed to by the arc.



RDF Classes

- Resources can be instances of classes, in RDF.
- Everything in RDF is a resource, so can have relationships between classes. Example: Subclass relationship:



Inference Rules over RDF Graphs

- We can specify **inference rules** to find more relationships between nodes in the graph.

- Rules have the following form.

$$\frac{\overbrace{J_1 \quad J_2 \quad \ldots \quad J_n}^{premises}}{\underbrace{J}_{conclusion}}\ \texttt{Rule Label}$$

Pizza Rules and Judgments

$$\frac{}{\texttt{pepperoni}\quad\texttt{meat}}\,\texttt{P} \qquad \frac{}{\texttt{ham}\quad\texttt{meat}}\,\texttt{H}$$

$$\frac{}{\texttt{tomato}\quad\texttt{veg}}\,\texttt{T} \qquad \frac{}{\texttt{onion}\quad\texttt{veg}}\,\texttt{O}$$

$$\frac{}{\texttt{plain}\quad\texttt{meat\_pizza}}\,\texttt{MP} \qquad \frac{}{\texttt{plain}\quad\texttt{veg\_pizza}}\,\texttt{VP}$$

$$\frac{\texttt{M}\quad\texttt{meat}\qquad\texttt{P}\quad\texttt{meat\_pizza}}{\texttt{add(M,P)}\quad\texttt{meat\_pizza}}\,\texttt{ADDM} \qquad \frac{\texttt{V}\quad\texttt{veg}\qquad\texttt{P}\quad\texttt{veg\_pizza}}{\texttt{add(V,P)}\quad\texttt{veg\_pizza}}\,\texttt{ADDV}$$

Exercise: Derive `add(tomato, add(onion, plain)) veg_pizza`

Example Proof Derivation Tree

$$\dfrac{\dfrac{}{\texttt{tomato veg}}\,\texttt{T} \qquad \dfrac{\dfrac{}{\texttt{onion veg}}\,\texttt{O} \qquad \dfrac{}{\texttt{plain veg\_pizza}}\,\texttt{VP}}{\texttt{add(onion, plain) veg\_pizza}}\,\texttt{ADDV}}{\texttt{add(tomato, add(onion, plain)) veg\_pizza}}\,\texttt{ADDV}$$

Applying Rules to RDF Graph

- Can apply inference rules to an RDF graph to infer new statements in the graph.

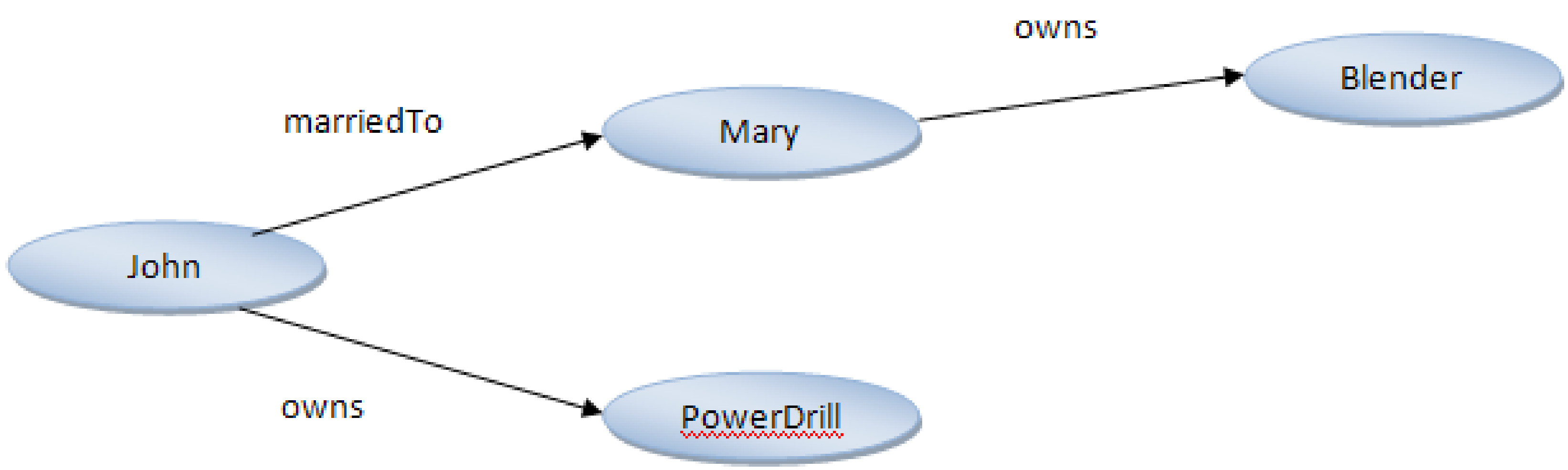
- Rules of marriage:

$$\frac{\texttt{P marriedTo Q}}{\texttt{Q marriedTo P}}\,\texttt{SYMM}$$

$$\frac{\texttt{P marriedTo Q} \quad \texttt{P owns I}}{\texttt{Q owns I}}\,\texttt{OWN}$$

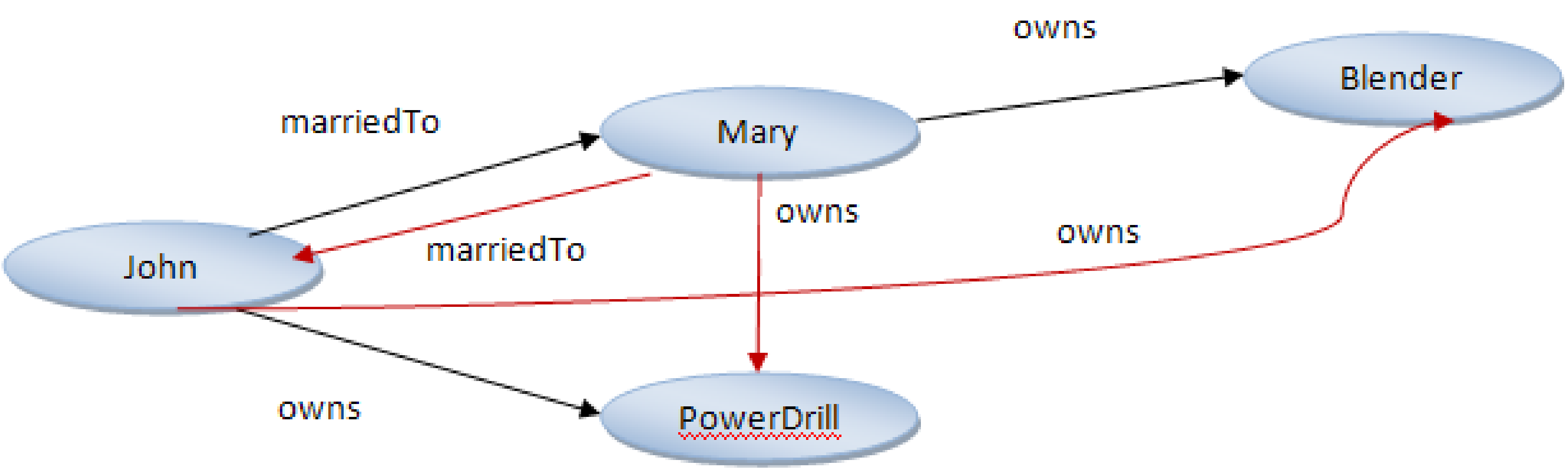
Application of Marriage Rules to RDF Graph

Original Graph:



Application of Marriage Rules to RDF Graph

Inferred Graph:



Feature Relationships for Safe CAD System Interoperability

- **Goal:** Determine when one feature from one CAD system can be replaced with another feature from another CAD system in a model without causing any incompability issues.

- Easy to infer when two features are known to be equivalent.

- Difficult when two features are not equivalent.

Subtyping

- Special case from programming languages: The subtype relation captures an "is-a" relation: When A is just a special type of B. If so, one conclude that an A can be used where a B is expected.

- Examples:
  1. Integer is a special type real number.
  2. Dog is a special type of animal.

- The relationship $\tau <: \tau'$ holds when: $Values(\tau) \subseteq_{\equiv} Values(\tau')$.

- Example: `int` $<:$ `double` because $Values(\texttt{int}) \subseteq_{\equiv} Values(\texttt{double})$

- $A \subseteq_{\equiv} B$ if and only if either of the two conditions hold:

  1. $A = \emptyset$.

  2. $\exists f : A \xrightarrow{1-1} B, \forall x \in A, x \equiv f(x)$.

- Can define same relationship between features.

Feature Subtyping & Valid Conversion

- A feature f_2 is a subtype (specialization) of another feature f_1 if f_1's attributes are an equivalent subset of f_2's attributes:

$$\frac{Attributes(f_1) \subseteq_{\equiv} Attributes(f_2)}{f_2 <: f_1}\ \texttt{SUBTYPE}$$

- Example:
  `ExtrudedHole[radius:real, depth:int]` $<:$ `Hole[radius:real]`

- Need inductive definition of $\subseteq_{\equiv}$ relation in order to derive inductive proofs with a semantic web reasoner.

Rules defining Equivalent Subset Relation

$$\frac{}{\emptyset \subseteq_{\equiv} A}\ \texttt{SUB\_EMPTY} \qquad \frac{A \subseteq_{\equiv} B}{A \subseteq_{\equiv} B \cup C}\ \texttt{SUB\_PROP}$$

$$\frac{A \subseteq_{\equiv} B \quad x \equiv x' \quad x \notin A \quad x' \notin B}{A \cup \{x\} \subseteq_{\equiv} B \cup \{x'\}}\ \texttt{SUB\_ADD}$$

$$\frac{}{A \subseteq_{\equiv} A}\ \texttt{SUB\_REFL} \qquad \frac{A \subseteq_{\equiv} B \quad B \subseteq_{\equiv} C}{A \subseteq_{\equiv} C}\ \texttt{SUB\_TRAN}$$

Practical Implications/Realizations

- Rigorous identification of safety criteria.

- Automated realization.
  - Automated verification that a feature mapping between two CAD systems is safe.
  - Automatic notification when intervention is needed between mappings that cannot be proven safe.

Future Work

- Case studies involving entire features sets of systems SolidEdge & SolidWorks.

- Rules reasoning over combinations of features, which can lead to one-to-many feature mappings.

- Involve a 3^{rd} tool (e.g. Pro-E) in further case studies.