# Long Term Knowledge Retention: Standards & Representation

## NIST March 15, 2006

# Archiving Manufacturing Knowledge

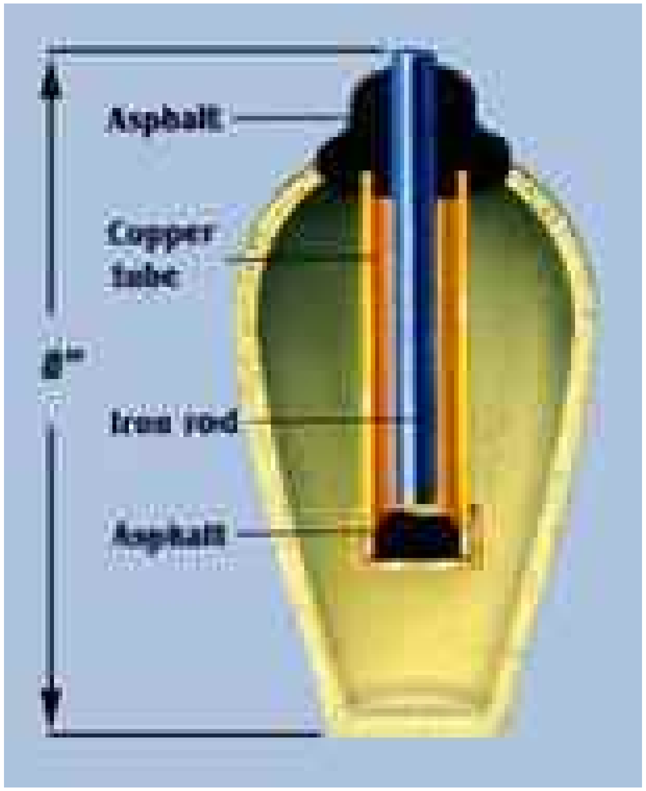
Dr. Frank Brown

University of Kansas

# The Bagdad Artifact
# 200 BC

- 

# What is it?

- 

- "It's a pity we have not found any wires.

- "It means our interpretation could be completely wrong."

# Metadata could Preserve Artifact Design

- This protrusion is the Annode
- This protrusion is the Cathode
- Here is a wire attachment protrusion
- ...

# Knowledge-based Archiving of Manufacturing Part Shape

- Conducted by NNSA and NARA in 2003
- Goal:
  - Archive a manufacturing part shape, with authenticating features, in the NARA Electronic Records Archives (ERA) prototype.
  - Retrieve the archived part shape from the NARA ERA prototype and authenticate.

# Approach

- Start with ISO 10303 STEP AP-203 and transform into knowledge form.
- <u>Deduce</u> authenticating aspects of part shape (features) by reasoning over the part geometry and topology.
- <u>Represent</u> the part shape in the W3C Ontologic Web Language (OWL) format for archiving.
- Archive the OWL part, with its authenticating features, in the NARA ERA research prototype.
- Retrieve the OWL part from the ERA and authenticate it by again deducing its shape features and comparing them to the shape features previously archived.

# Scope of our Work

- AIRIT
  - design/development of knowledge logics and axioms
  - design/development of inference systems
  - design/development of translation systems among
    - (STEP/ Logics used for inference / DL / OWL)
- KCP-NNSA
  - domain knowledge about part features
  - Noel Christiansen made first cut of axioms
  - operational aspects of overall experiment of transmitting/archiving data
    - transmittal to and from NARA ERA prototype

# Design Geometry Inferencing

- Representation of a Part:
  - Step ISO 10303 AP-203 Topology and Geometry
    - vertices, edges, loops, faces, points, curves, surfaces
- Representation of Knowledge about Parts
  - axioms describing different kinds of part features
    - bosses, open pockets, cutouts, through holes, face chains
- Automatic Deduction System to infer Knowledge
- Meta System to easily implement Automatic Deduction Systems
  - Logistica

# Axiom for Bosses

```
•  (   (boss ?id :protrusion-entrance ?protrusion-entrance
•                :sides ?sides
•                :top ?top)
•      if
•       (protrusion-entrance ?protrusion-entrance
•               :border-faces entrance-sides)
•       (external-face ?top
•               :border-faces top-sides)
•       (maximal-face-chain ?sides
•               :faces boss-sides
•               :convexity-of-left-edge 'concave
•               :is-loop '#t)
•       (and (floor-planar ?top)
•            (are-equal entrance-sides boss-sides)
•            (are-equal top-sides boss-sides)
•            (are-vertical boss-sides))   )
```

# Maximal Face Chain Idea

```
(maximal-face-chain ?sides
          :faces boss-sides
          :convexity-of-left-edge 'concave
          :is-loop '#t)
iff
     (face-chain ?sides
          :faces boss-sides
          :convexity-of-left-edge 'concave
          :is-loop '#t)
     (not (exists (?sides2 boss-sides2)
               (and (face-chain ?sides2
                    :faces boss-sides2
                    :convexity-of-left-edge 'concave
                    :is-loop '#t)
                  (sub-chain boss-sides bosssides2) ))
```

# Need for FOL and SOL

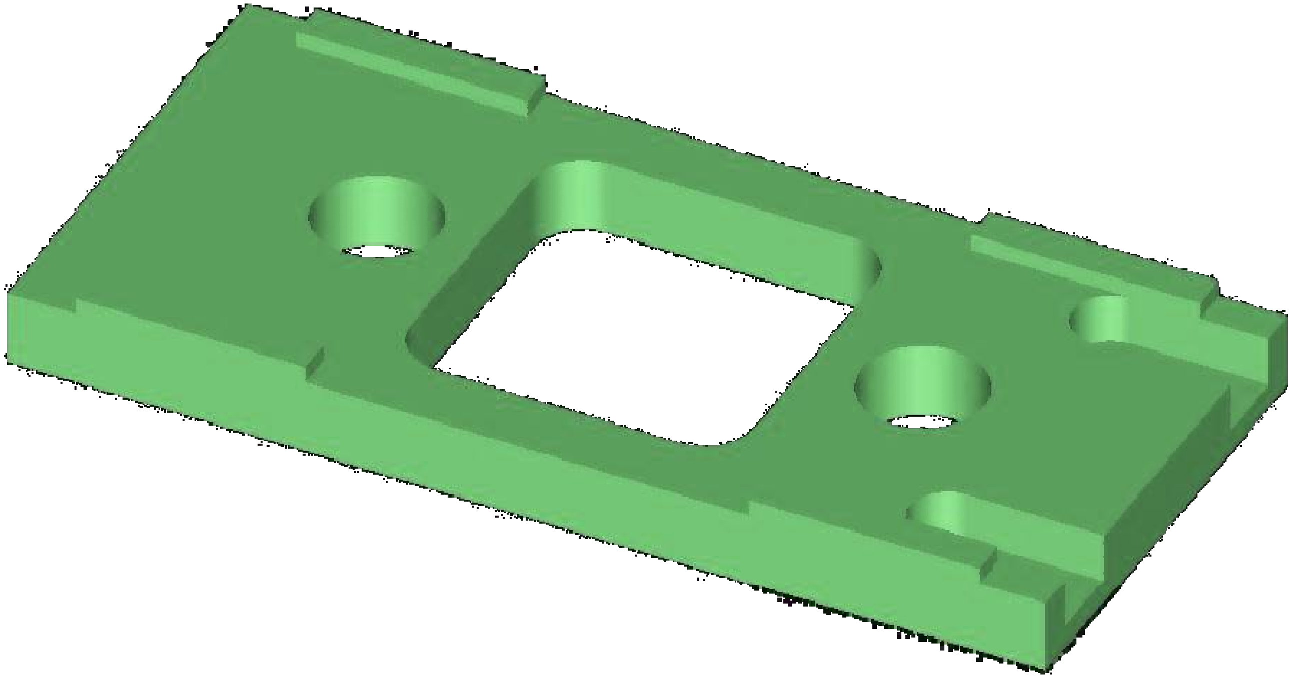
- Maximal Face Chains are defined in terms of knowing what is not a face chain.
  - this is not a Horn clause so FOL is needed.
- Face Chains are a recursive definition
  - We need to derive the proper mathematical induction law over this recursive structure, so SOL is needed.

# Some Deduced Features

- tuples specifying the number of features of each concept
  - ($cutout 2)
    ($thru-round-hole-0 4)
    ($closed-pocket 0)
    ($blind-round-hole-0 0)
    ($rectangular-cutout 0)
    ($closed-rectangular-pocket 0)
    ($boss 4)
    ($rectangular-boss 4)
    ($open-pocket 2)
- tuples specifying the what are the different features

# How many bosses are in the picture?

-



# How many bosses are in the picture?

- Actually zero.
- because the axiom says that a boss consists of a surface with a maximal face circular face chain of 4 connected rectangular faces.
- but each boss in the picture is at the edge of the part making that face be part of the side of the part's base.

# How do we fix this?

- General Nonmonotonic Actions (with appropriate frame laws) are used to modify the part by so as to result in better overall descriptions of the part.

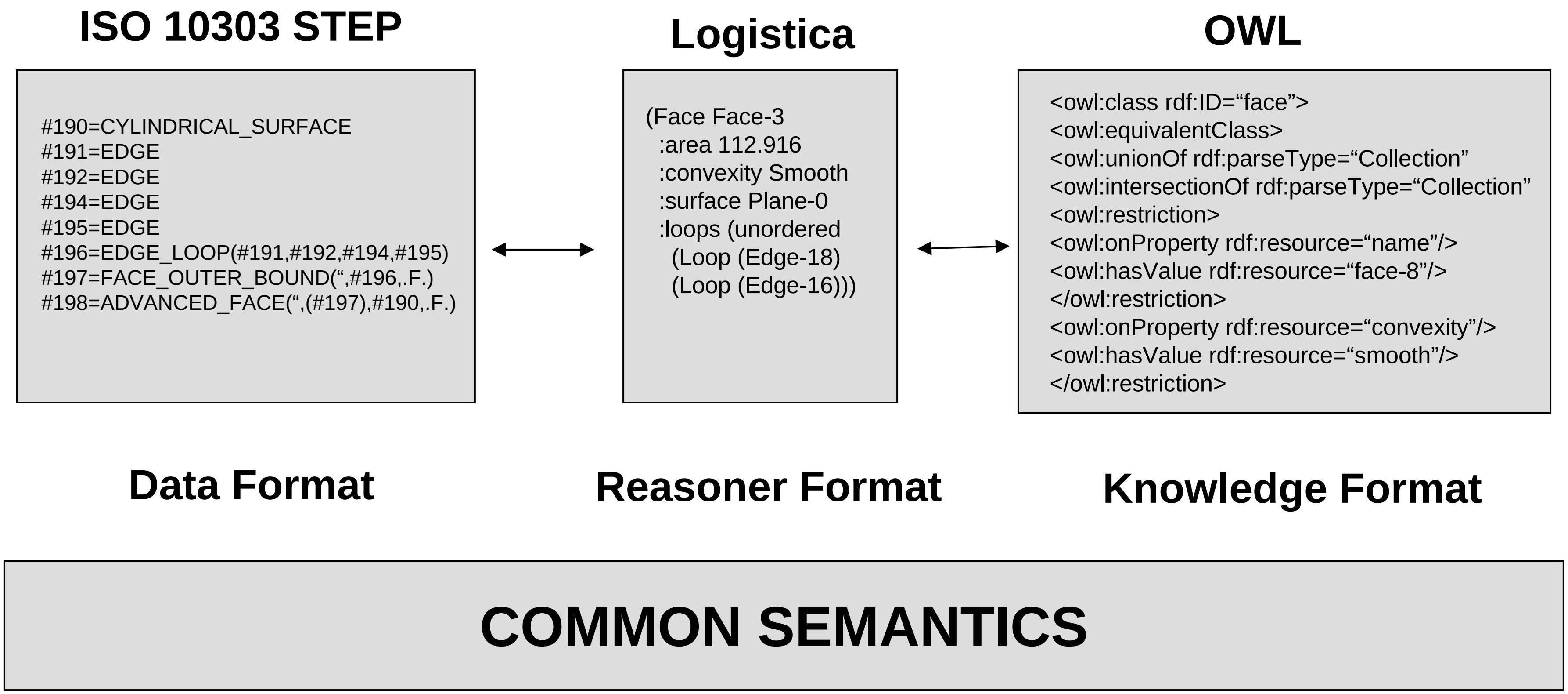
# Requirements for Knowledge Inferencing on Machined Parts

- Simple Sentences (STEP)
- Horn Clauses (Many simpler axioms)
- FOL  (Total Number of Faces)
- SOL  (Maximal Face Chains)
- Nonmonotonic Actions  (mutilations)
- Frame Laws (mutilations)

# Automatic Translation

- standards and other languages
  - STEP, DL, OWL, Logic(FOL facts, FOL axioms)
  - Logistica(SOL, Actions+FrameLaws)
- Automatic Translation systems
  - Step <--> Logic
  - Logic <--> DL
  - DL <--> OWL
- System for easily implementing translation systems: Logistica

# Same Meaning - Many Formats



**ISO 10303 STEP**

```
#190=CYLINDRICAL_SURFACE
#191=EDGE
#192=EDGE
#194=EDGE
#195=EDGE
#196=EDGE_LOOP(#191,#192,#194,#195)
#197=FACE_OUTER_BOUND(“,#196,.F.)
#198=ADVANCED_FACE(“,(#197),#190,.F.)
```

**Data Format**

**Logistica**

```
(Face Face-3
  :area 112.916
  :convexity Smooth
  :surface Plane-0
  :loops (unordered
    (Loop (Edge-18)
    (Loop (Edge-16)))
```

**Reasoner Format**

**OWL**

```
<owl:class rdf:ID=“face”>
<owl:equivalentClass>
<owl:unionOf rdf:parseType=“Collection”
<owl:intersectionOf rdf:parseType=“Collection”
<owl:restriction>
<owl:onProperty rdf:resource=“name”/>
<owl:hasValue rdf:resource=“face-8”/>
</owl:restriction>
<owl:onProperty rdf:resource=“convexity”/>
<owl:hasValue rdf:resource=“smooth”/>
</owl:restriction>
```

**Knowledge Format**

**COMMON SEMANTICS**

# Requirements for Knowledge Transmission of Machined Parts

- DL
  - OWL (existing at time of our project)
- FOL Horn Clauses
  - some current web work
- FOL
  - KIF ?
- SOL (restrictions on quantifier seq.?)
  - SKIF ?
- Actions with simple results
- Nonmonotonic systems of Frame Laws

# Logistica

- A Meta-programming System
- Allows efficient deduction and translation processes to be quickly specified and refined
- Synergistic Features

# Logistica Software Productivity on this project

- design and development time
  - 3 month project
  - about 6 man months
  - proved to NNSA that the approach led to rapid results
- quality of result
  - Low polynomial bounded deduction algorithms.
  - Deduce only what is needed (not everything)
  - Systematically avoid re-deducing things
  - proved to NNSA that the Deduction System worked in practical time

# Dr. Frank M. Brown

- email: brown@ku.edu
- telephone: (785) 864-7376