ADVANCED KNOWLEDGE AKT

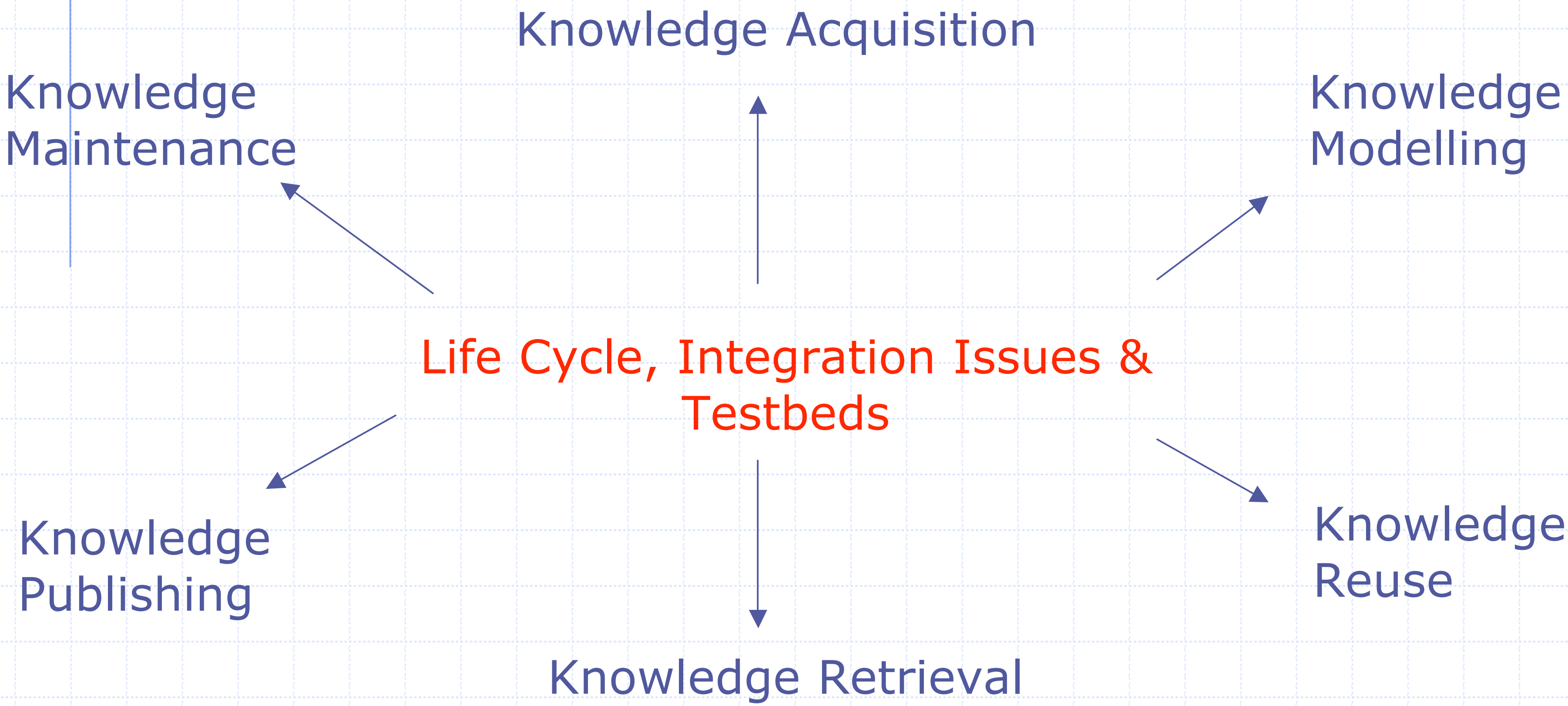
The role of ontologies in creating & maintaining corporate knowledge: a case study from the aero industry

Derek Sleeman (Abd)
Suraj Ajit (Abd)
David Fowler (Abd)
David Knott (R-R)

Jeff Pan (Abd)- associated

UNIVERSITY OF ABERDEEN

AKT's Challenges



The AKT Consortium

SOUTHAMPTON	**Prof Nigel Shadbolt** KA, Cognitive Psychology, Agents, Planning
	Prof Wendy Hall Multi-media & hyper-media systems
ABERDEEN	**Prof Derek Sleeman** Cooperative KA & Knowledge Refinement Systems, Discovery Systems
	Prof Peter Gray / Dr Alun Preece Knowledge Representation & Integration, V&V
EDINBURGH	**Prof Austin Tate (AIAI) & Dave Robertson** Planning, Enterprise Modelling, Semi-formal handling of Requirement Analysis
OU	**Dr Enrico Motta** Configuration of PSMs, Ontology-driven KA, Tools for the annotation of texts
SHEFFIELD	**Prof Yorick Wilks** Systems/Tools for eliciting Knowledge from Texts; Computational Linguistics

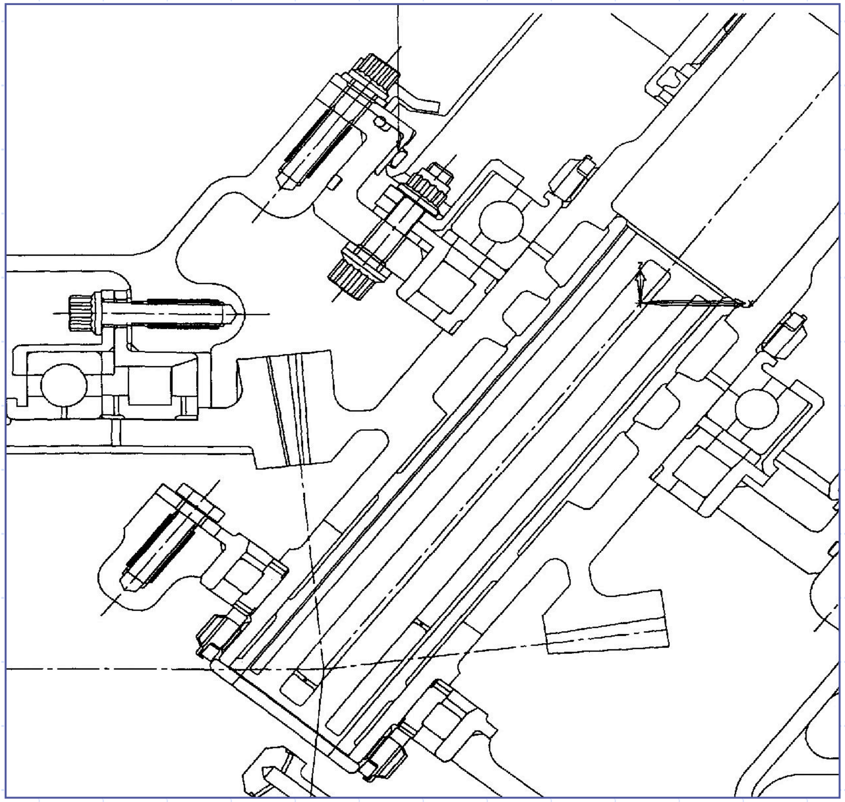
Rolls-Royce testbeds

- To enable Knowledge Technologies to be tested on real-life problems

- Initial meetings at RR led to two testbeds being identified:
  - Intelligent Document Retrieval (Gary Wills, Soton)
  - Designers' Workbench (which begat ConEditor)

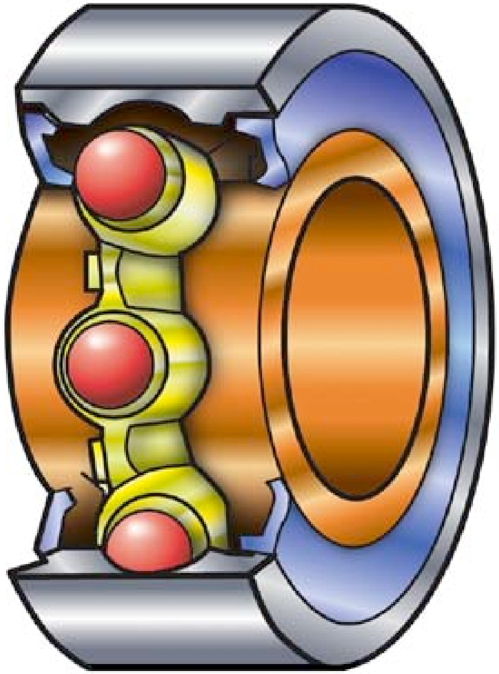
Designers’ Workbench

Scenario 1: Overlooking rules



- O-ring failure, causing in-flight shutdowns
- Cause: in certain conditions, o-rings can become twisted during assembly & disassembly

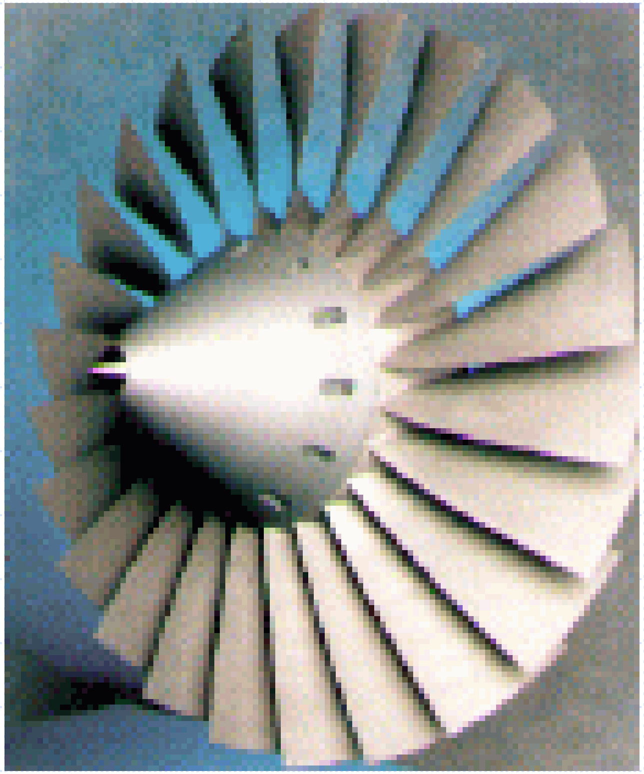
Scenario 2: Lost rationales



(Image from www.duratrax.com)

- A rule states that the total load on a bearing must be < 125 tonnes psi
- But the reason for this rule has been lost…

Scenario 3: Bending the rules



(Image from www.grc.nasa.gov)

- Parts that rotate relative to each other must be ≥ 5mm apart
- But experienced designers can bend this rule in some conditions

An example design rule

9.3.2 Internally trapped nuts (see Fig 4 Table 4)

TABLE 4

PCD ABOVE TO	2M
mm mm	mm
150 - 180	1,00
180 - 300	0,80
300	0,60

Ø N MIN. = PCD (NOM.) + 2M + MAX. NUT WIDTH
(SEE TABLE 5)

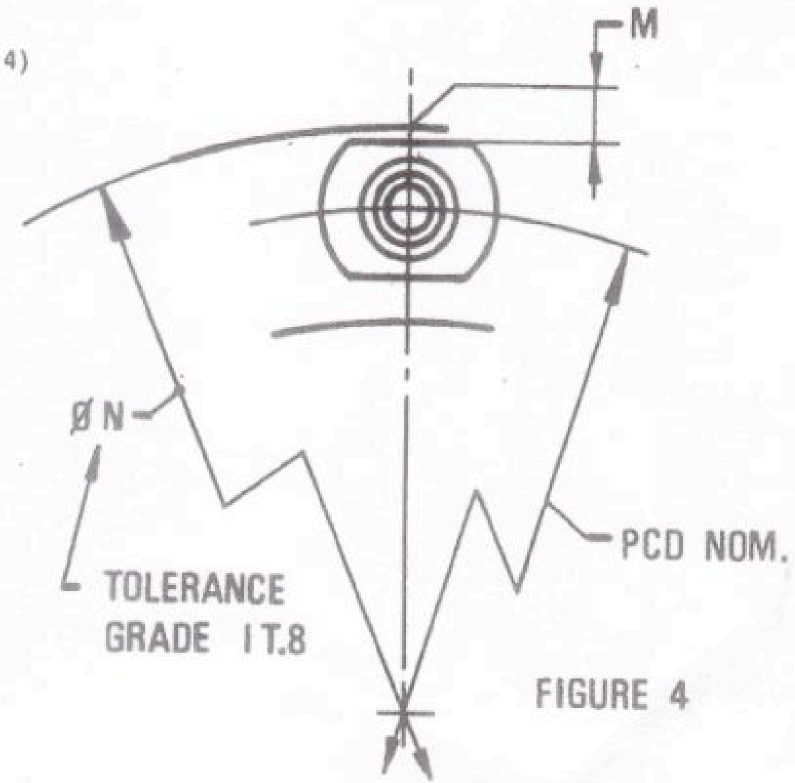
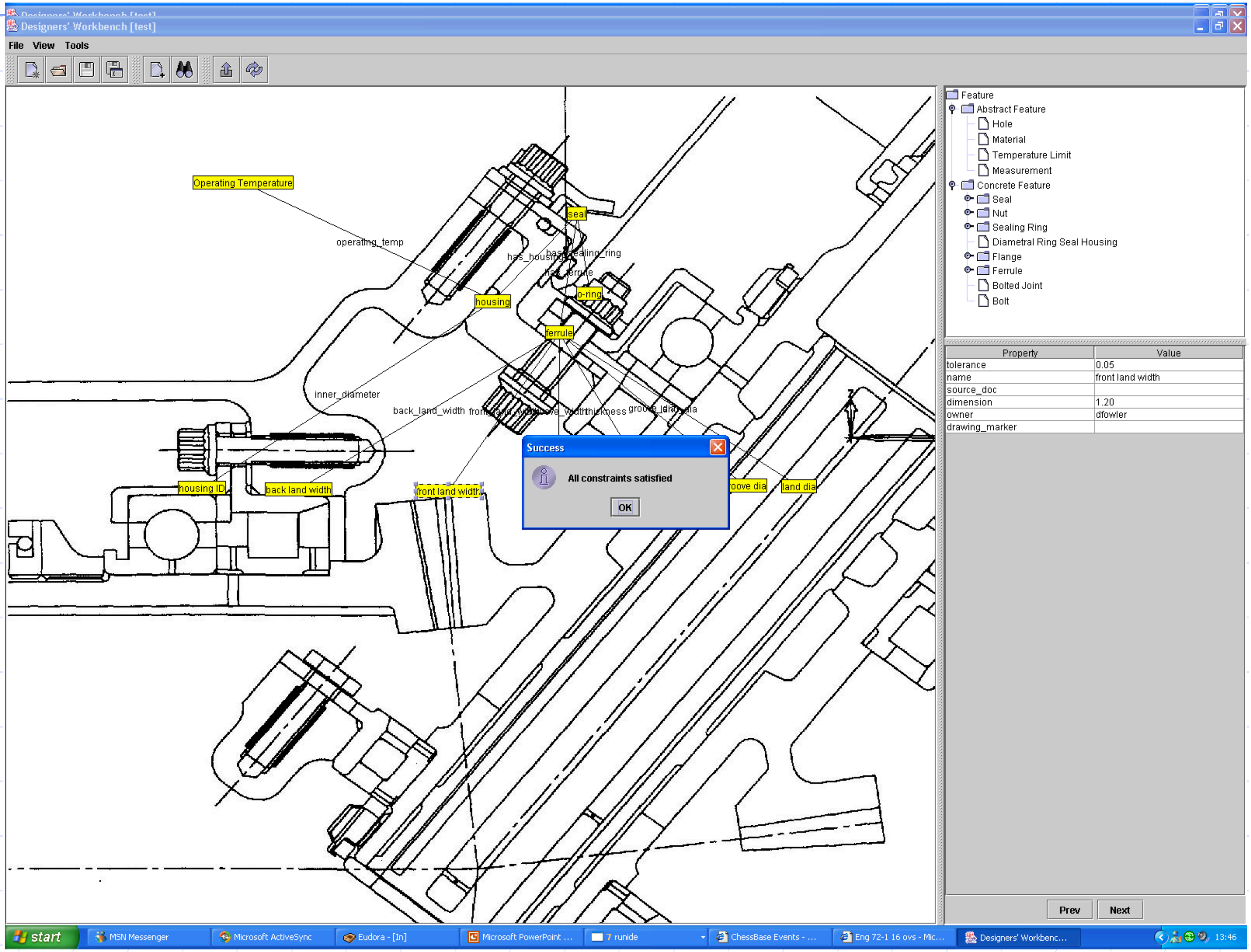


FIGURE 4

Initial aims

- To represent designs / parts against an ontology
- To implement design rules so that they can be checked automatically
- To give feedback to the designer about the violated rules

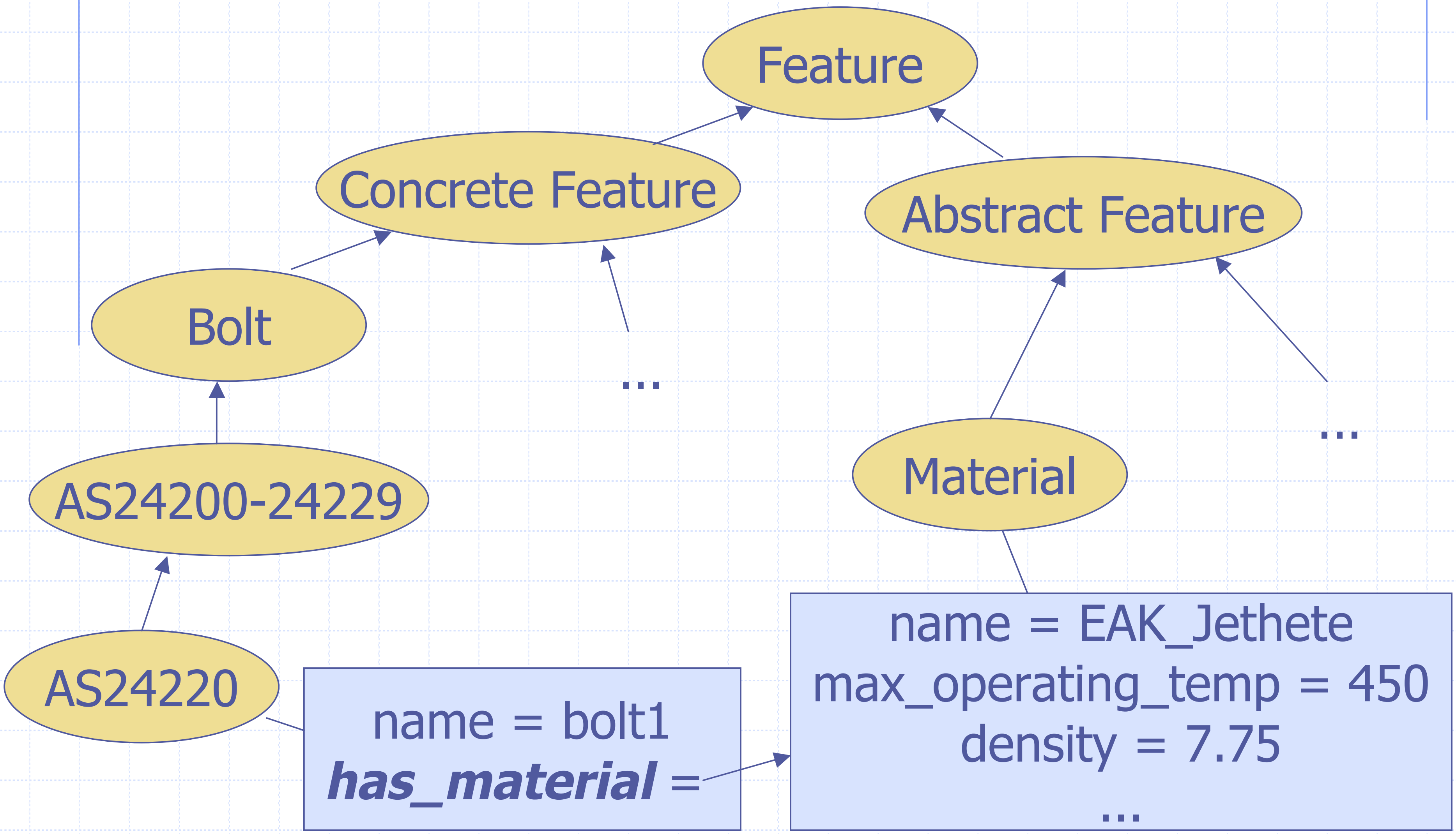
Designers' Workbench



The ontology (1)

- Classes represent types of “features”
  - physical items, e.g. nuts, bolts, assemblies
  - other items, e.g. holes, temperatures, materials
- Properties are defined on classes
  - e.g. **`Material`** class has a **`max_operating_temp`** property
- Instances represent specific features

The ontology (2): OWL-DL



Constraints (1)

- "Each concrete feature must have a material that can withstand the environmental temperature"

```
Constrain each f in ConcreteFeature
  to have
  max_operating_temp(has_material(f
  )) >= operating_temp(f)
```

CoLan version

Constraints (2)

- Constraints are handled in a two stage process:
  - a search is made using an RDQL query to find the features that are affected by a constraint
  - a call is made to a Sicstus Prolog predicate to check that the constraint holds

Constraints (3)

```
SELECT  ?arg1, ?arg2 WHERE         (RDQL)
(?feature, <dwOnto:has_material>, ?mat),
(?mat, <dwOnto:max_operating_temp>, ?arg1),
(?feature, <dwOnto:operating_temp>, ?optemp),
(?optemp, <dwOnto:temperature>, ?arg2)
USING dwOnto FOR [insert URI here]
```

```
operating_temp_limit(MaterialMaxTemp,
  EnvironTemp) :-                   (PROLOG)
EnvironTemp =< MaterialMaxTemp.
```

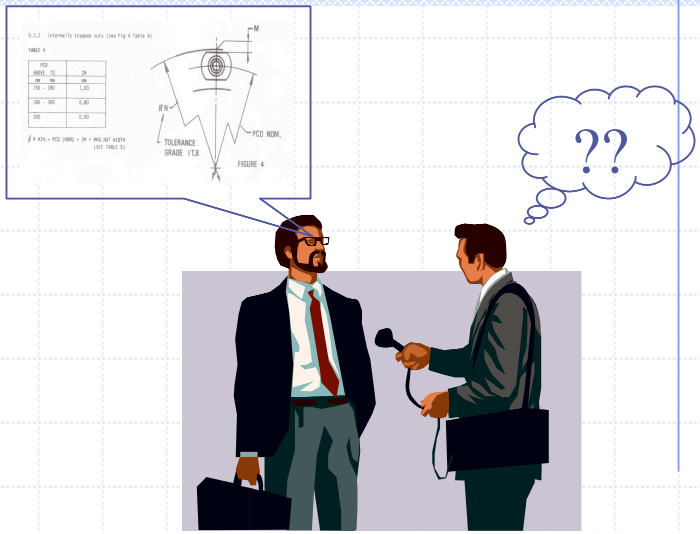
Issues/Wish list

- Allow constraint propagation/solving rather than just checking
- A better query language than RDQL?
- Integrate Designers' Workbench with a CAD/KBE system

- Facility to allow engineers to input & maintain the Knowledge (constraints)
- Better rationale / context management

ConEditor

- The Designers' Workbench needs constraints.
- Currently, a KE interviews designers...
- ...and studies documentation...
- ...and then implements the constraint using RDQL/Prolog.
- A tedious, error-prone task!

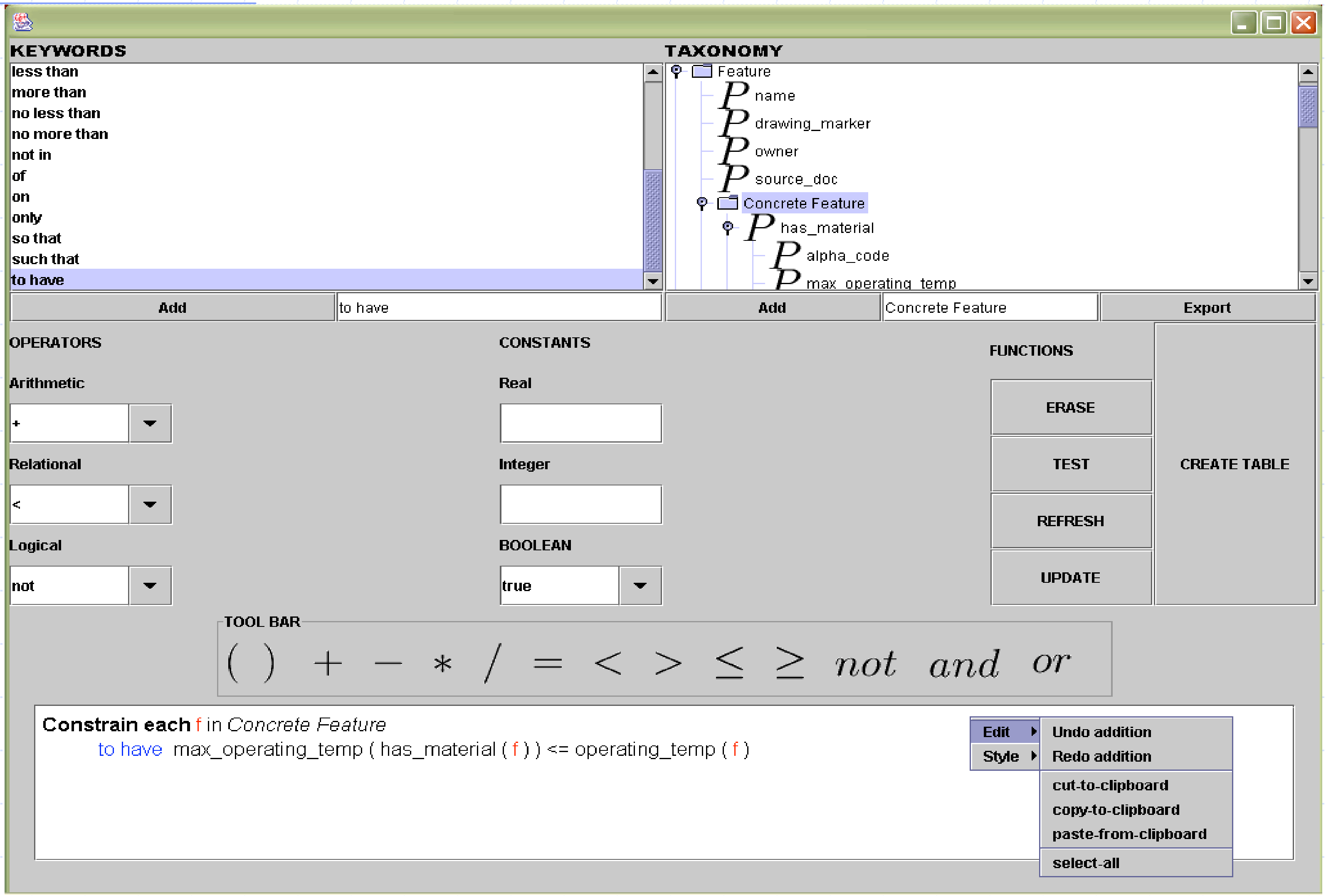


ConEditor

- Aim: to provide designers with an intuitive way to capture/input and maintain the constraints themselves.

- Designers will have control over the definition and refinement of constraints $\Rightarrow$ greater trust in the resulting constraint checks.

ConEditor



KEYWORDS
less than
more than
no less than
no more than
not in
of
on
only
so that
such that
to have
Add
to have
TAXONOMY
Feature
name
drawing_marker
owner
source_doc
Concrete Feature
has_material
alpha_code
max_operating_temp
Add
Concrete Feature
Export
OPERATORS
Arithmetic
+
Relational
<
Logical
not
CONSTANTS
Real
Integer
BOOLEAN
true
FUNCTIONS
ERASE
TEST
REFRESH
UPDATE
CREATE TABLE
TOOL BAR
( ) + − * / = < > ≤ ≥ not and or
Constrain each f in Concrete Feature
to have max_operating_temp (has_material (f)) <= operating_temp (f)
Edit
Style
Undo addition
Redo addition
cut-to-clipboard
copy-to-clipboard
paste-from-clipboard
select-all

ConEditor: Maintenance of Constraints

- Constraints might:
  - only apply in certain conditions
  - evolve
  - become redundant
  - require revision

- Add *application conditions* to constraints. Using constraints, application conditions & ontology: detect subsumption, contradiction, & redundancy

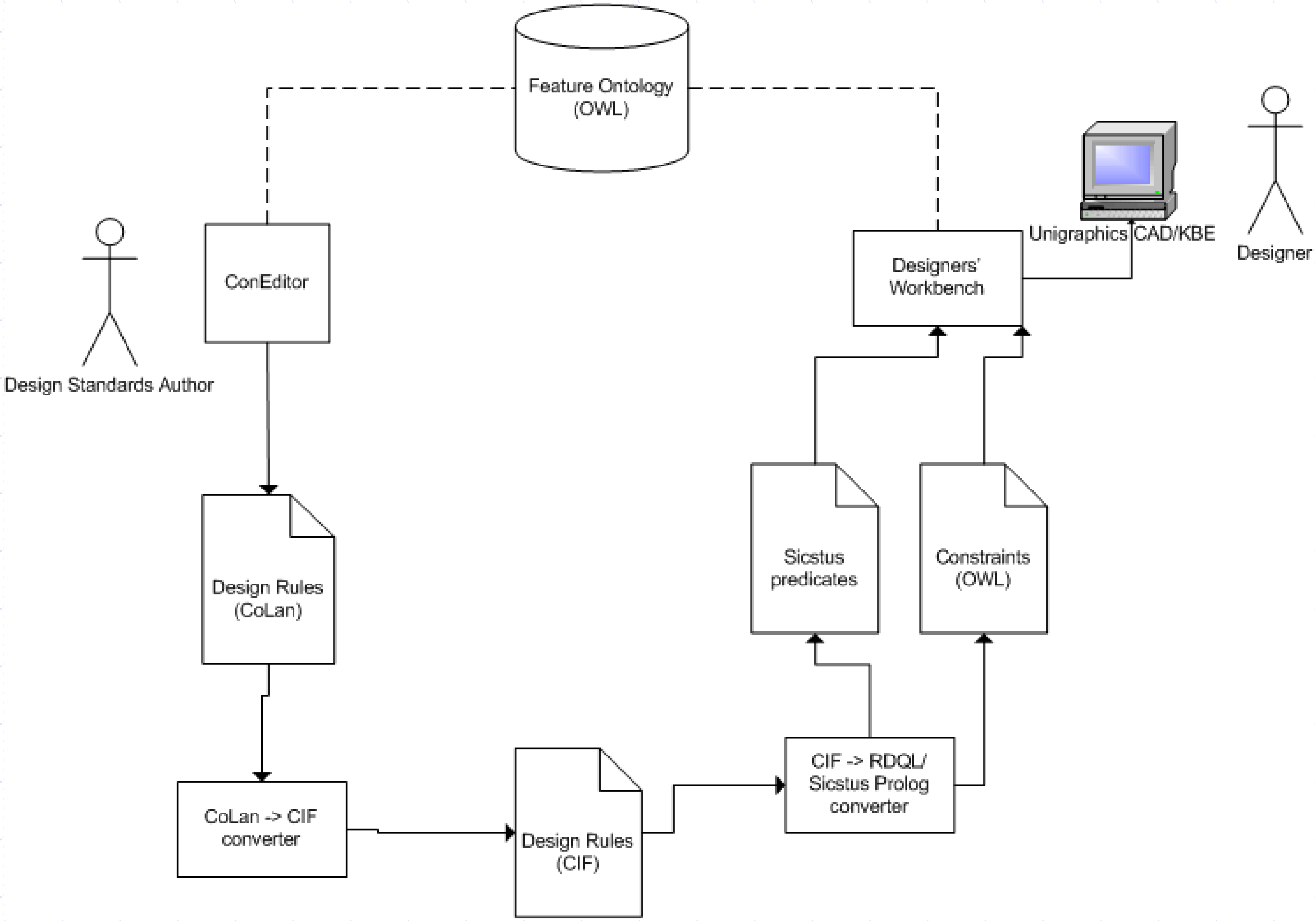
Application conditions example

```
Constrain each k in Kite

such that has_type(k) = "Flat" and
  has_shape(k) = "Diamond"

to have tail_length(has_tail(k)) =
  7 * spine_length(has_spine(k))
```

Application conditions example

```
Constrain each s in Sled_kite
such that has_size(s) = "standard"
to have kite_line_strength(has_kite_line(s))
  >= 15

Constrain each c in Conventional_sled_kite
such that has_size(c) = "standard"
to have kite_line_strength(has_kite_line(c))
  >= 15
```

Planned System Architecture



IPAS

Integrated Products and Services

IPAS: Information Life Cycle

- Providing extensive feedback from engineer maintenance & service facilities to designers
  - Likely that information will be described against (subtly) different ontologies
  - Identify important questions the designers wish to have answered

NB

- We are dealing with sophisticated items with life spans exceeding those of an individual's working life.
- Shift from designing/producing products to provision of services.

Designer's Knowledge Desktop: highlights

- Vision Demonstrator (DS&S)
- How will the Knowledge desktop fit in with a designer's current working environment? (Sheffield UTP)
- Interviews with designers and identification of questions they would like to ask about service data (Cam UTP)
- Multi-faceted document to describe role of ontologies in IPAS, their benefits, uses in other industrial settings, our methodology for ontology development, how to use IPAS ontologies in JAVA programs & in web services… (Abd)
- Implement Parts & Deterioration Mechanisms Ontologies (details on request) (Abd)
- Analysis of service event reports, and extraction of data *driven by the IPAS ontologies* (Sheffield AKT)
- Population of the Ontology from a variety of Sources (Epist)
- Knowledge Desktop (web services) Demo (Southampton AKT)

Revisiting the Requirements analysis

- Do we need to have ontologies to reflect both the design & the service perspectives? (Will a single composite ontology be sufficient?)
- Do we expect the ontologies to be revised / extended as the project progresses? (Ontology developed for the Trent 500 could be extended for another Trent engine; possibly Commercial to Military, … Ontology Evolution)
- Will concepts have to be represented in differing levels of details in different applications / services? (Basic ontology covers main concepts but perhaps not in sufficient depth for a particular service: ... Modularization of Ontologies)
- How complex do we expect the questions / services to become? (For some questions will we need to know about Engine Geometry? ... Hybrid Representations)

IPAS: Technologies to be used

- Strong focus on:
  - Integration of large-scale heterogeneous knowledge sources;
  - Meta-data, semantics, ontologies, vocabularies and lexicons;
  - Ontology Management environments (capture, evolve, modularization)
  - Text mining, search, analysis and knowledge representation.
  - Modelling and simulation;
  - And Web/Grid services and use of standards.

Questions/Comments?

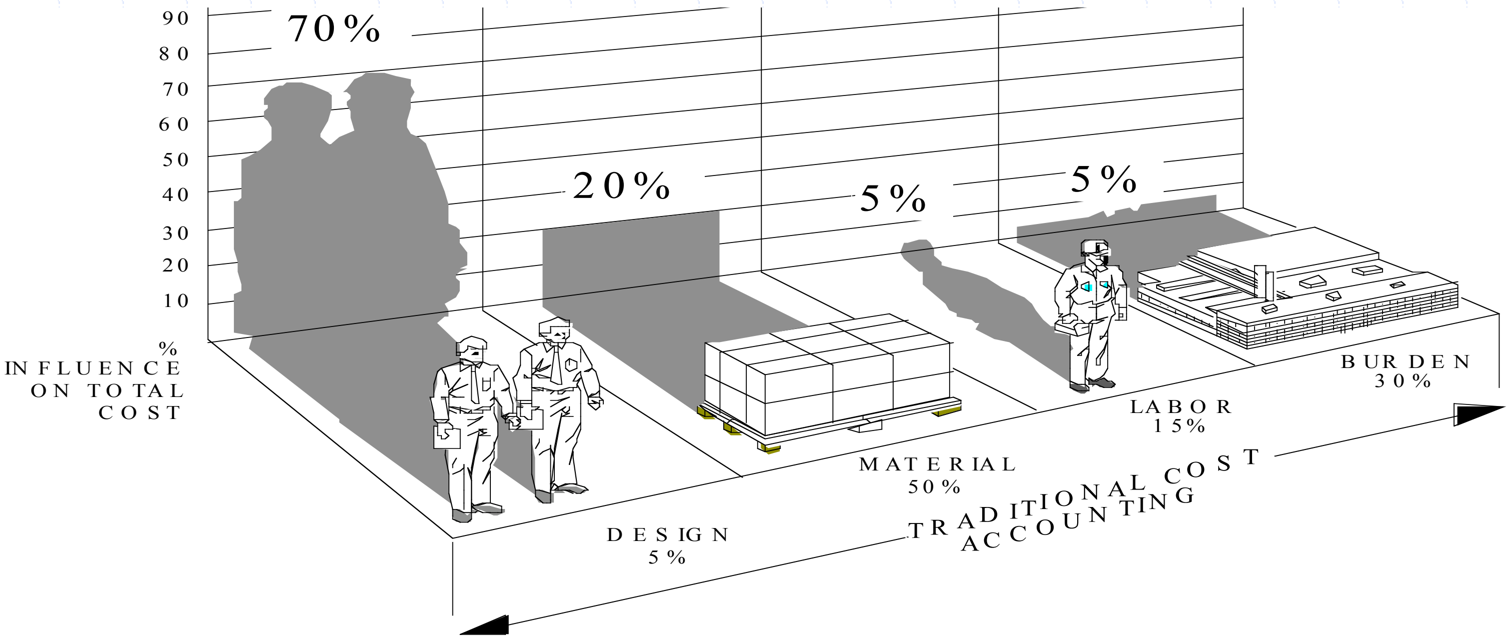
Summary of Aberdeen Work to date

- How is the servicing of aircraft engines organized?
  - Who are involved? What info do they use? Produce?
- Explore appropriate technologies to represent OWL ontologies
- Multi-faceted document to describe role of ontologies in IPAS, their benefits, uses in other industrial settings, our methodology for ontology development, how to use IPAS ontologies in JAVA programs & in web services…
- Discover various appropriate Knowledge Sources (eg Service event reports, Strip reports etc)
- Implement various Ontologies: Parts & Deterioration Mechanisms
- Developed Web Service according to specification of Southampton AKT (initial Designer's Knowledge Desktop)
- Developed CleanONTO (checks taxonomic structure of Ontology)

Integrated Products and Services (IPAS)

- Project involving:
  - Rolls-Royce (Lead)
  - DS&S
  - Epistemics
  - Aberdeen AKT
  - Cambridge UTP, Cambridge EDC
  - Sheffield AKT, Sheffield UTP
  - Southampton AKT, Southampton UTP
- Funding to the AKT partners from DTI & industrial partners: ~£200k (1RF + 1RS) over 3 years

Design has the greatest effect on total cost



Barnard Associates Ltd 1999

Source: Munro & Associates