Poseidon House
Castle Park
Cambridge CB3 0RD
United Kingdom

TELEPHONE: **Cambridge (01223) 515010**
INTERNATIONAL: **+44 1223 515010**
FAX: **+44 1223 359779**
E-MAIL: **apm@ansa.co.uk**

ANSA Phase III

Meta-information for Federation

(Presentation to ANSA Technical Committee 28/2/95)

Ben Crawford

Abstract

Work on federation can be divided into three aspects. One of these is the standardisation and modelling of the properties of different systems, and hence the differences between them. This aspect is fundamental to federation, because systems must have some common view through which to establish agreement and hence co-operation.

Previous work on modelling of properties, for example in the ANSAware Trader and MatchMaker, paves the way, but processes like interception and federated trading require more information, so further work is needed.

The meta-information approach gives us a way of analysing the problem which can help us to make the right decisions about what to standardise on. It can also promote reuse, by helping us to tackle the information management problems which standards introduce in a standard way.

In a fast-changing world, management of diversity will become an increasingly important aspect of distributed systems and their management engines. As the key to this problem, the meta-information approach seems likely to play an important role in systems architectures in the next few years.

APM.1421.01 **Approved** 28th February 1995

Project Management (confidential to ANSA consortium for 2 years)

Distribution:
Supersedes:
Superseded by:

Why do we need Federation?

- **Federation tackles connection of systems with diverse properties:**
  - **How to connect existing heterogeneous systems.**
  - **How to build new systems that will be easier to federate with in the future.**
- **This enables us to:**
  - **Support (unforeseen) changes in business practice.**
  - **Support alternative technologies.**
  - **Incorporate new technologies.**
  - **Protect investments and promote cheap reuse.**

A top-level view of Federation

- We can loosely divide the work area into three aspects.
- 1. Processes:
  - A set of processes for managing property differences.
  - These are interception, trading, binding, and resolving.
- 2. Information:
  - Modelling and managing property repositories to support gateways.
- 3. Mechanisms:
  - To manage property repositories.
  - To implement the processes using information from repositories.
- We cannot make full use of recent work on processes without more progress in property modelling.

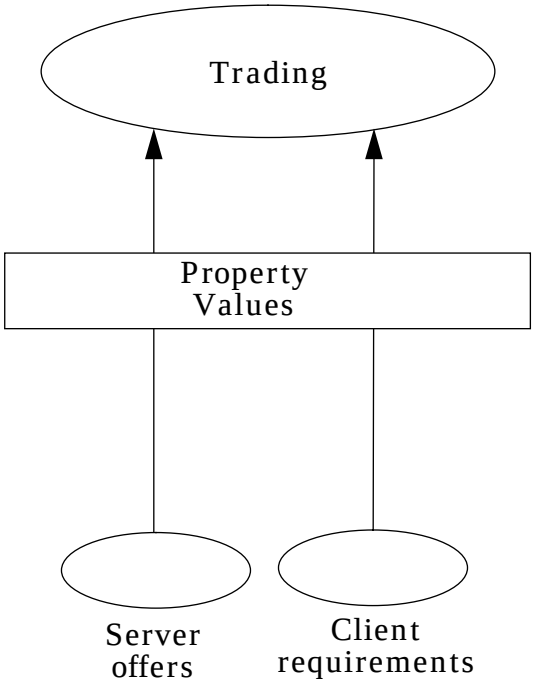
Property Modelling Review

- **Analyse and classify properties:**
  - **What kinds of properties do we need in practice?**
  - **Information model for federation [APM.1229].**
  - **Micro-scenarios for federation.**
  - **This aspect is not a priority.**

- **Modelling information about properties:**
  - **ANSAware Trader [APM.1005].**
  - **MatchMaker [APM.1384].**

Properties in existing Traders

- Trading uses interface types and other properties to match offers and requirements. E.g:
  - "Interface type = (INOUT String)",
  - "Resilience = HIGH".
- This works fine as long as clients and servers use the same properties.
- But effectively this means they are inside a single domain.
- The matchmaker provides for different models of properties, but has no support for relating them.



Federation requires extra information

- A trader in a federation might face this:
- Client requirement:

```
Interface =“(IN Cursor& resultsBuffer,IN String queryString,
             OUT Cardinal recordsFound)”
Resilience >= “HIGH”
Time Limit <= 8
```

- Server Offer:

```
Interface = “(IN querySpec query, IN Buffer& results,
             OUT Integer bufferSize)”
Failure rate = 0.001
Mean response time = 1.2
```

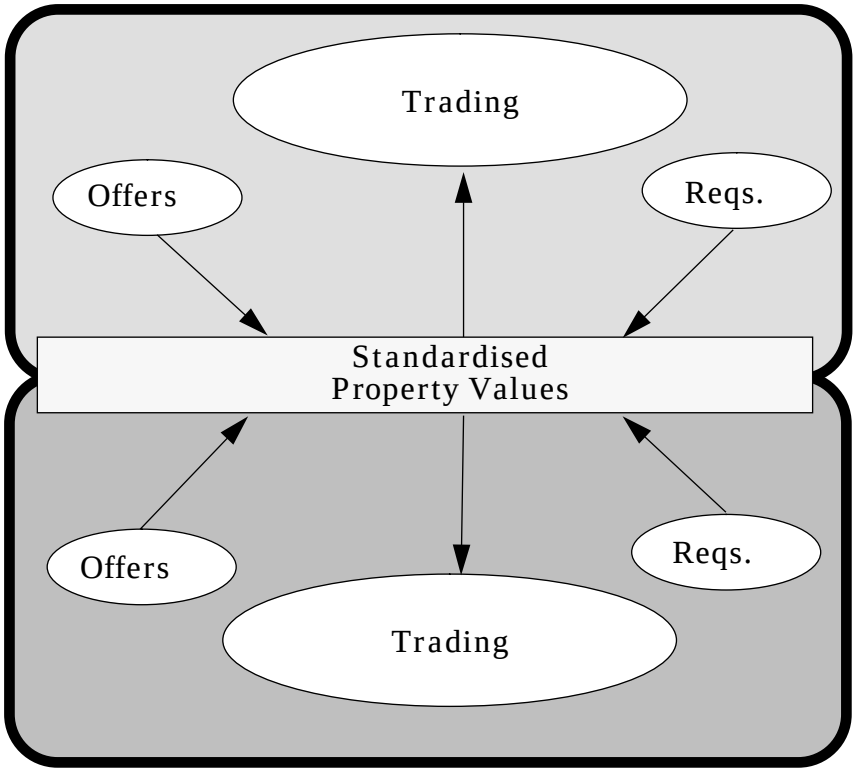
- Failure to bridge this gap could be costly.
- But to succeed, standardisation is needed.

How do we choose the right standard?

- We cannot interwork without standards.
  - What should we choose to standardise?
  - What are the implications of our choices?
- What do we want from a standard?
  - Stability.
  - Minimal constraints.
  - As widely applicable as possible.
  - Easy to understand and apply.
- Example: federating interface types.
  - Standardise property values, e.g. the set of all interface types.
  - Standardise on a property type, e.g. (e.g. IDL).
  - Standardise a way of describing property types (e.g. an IDLDL / AST).

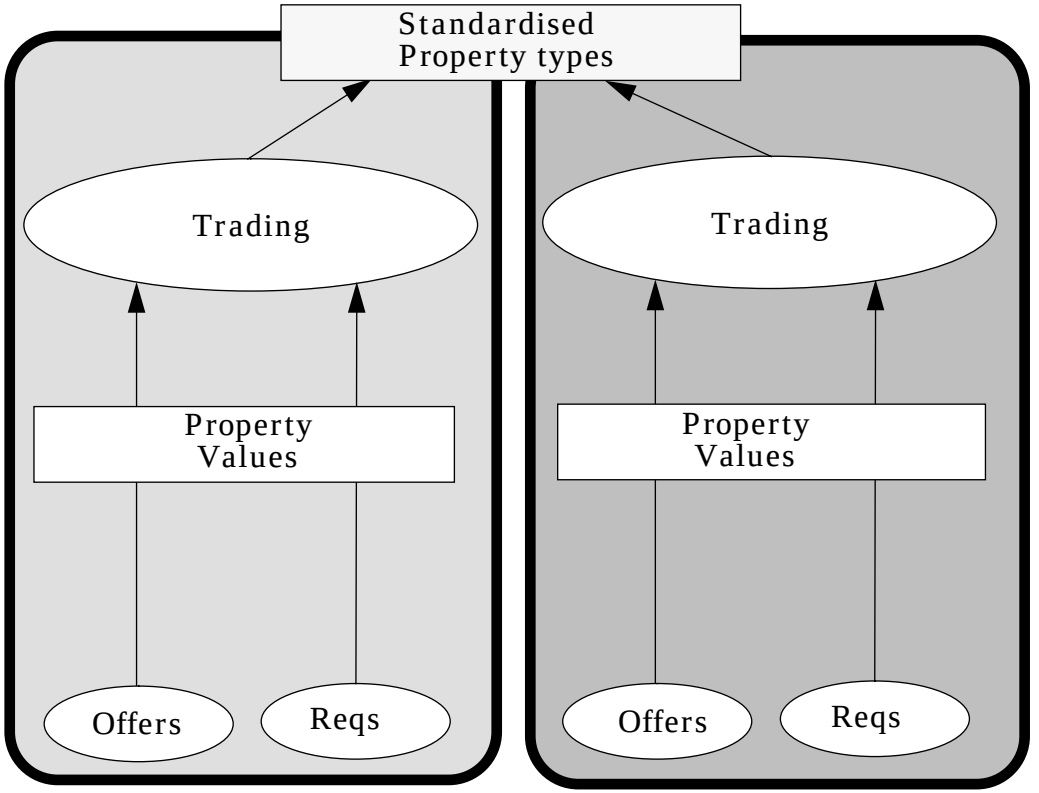
Standardising property values

- We could achieve cooperation by agreeing the form and meaning of property values.
- For example, interface types could be agreed globally. So whenever you produce a new service, you must include its IDL in a global standard!
- Then you pass the interface type as a property of an offer.



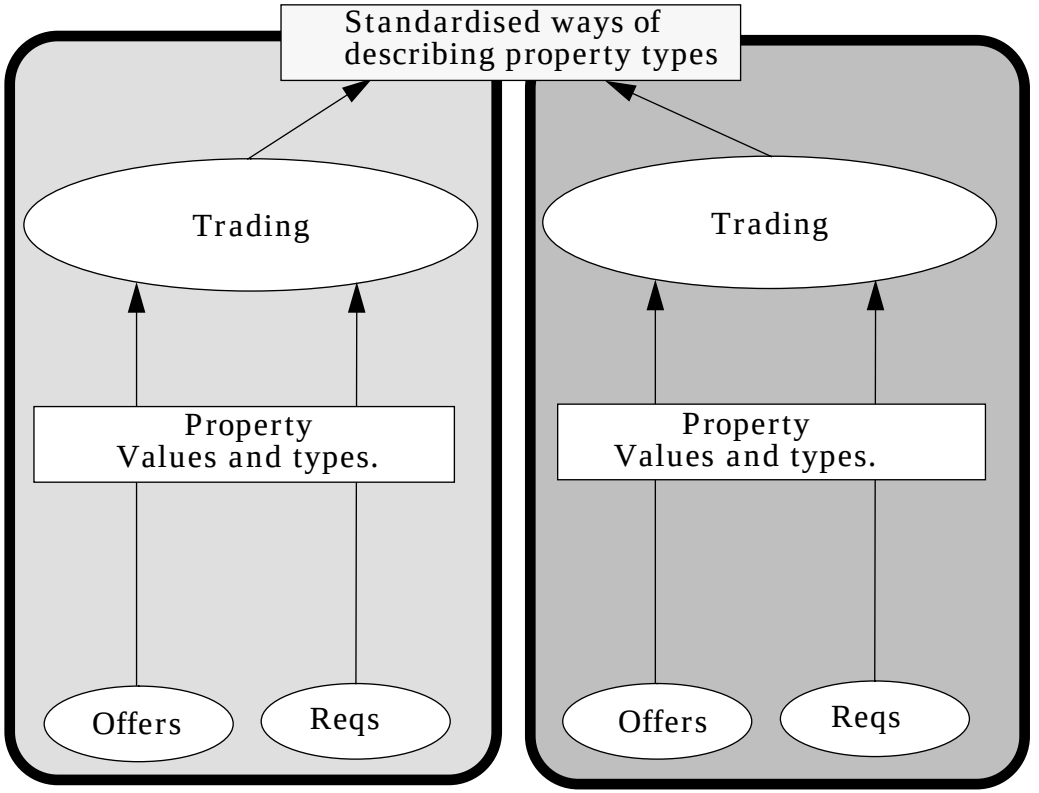
Standardising property types

- Alternatively we could agree types, i.e. ways of describing values.
- So you define IDL as part of a global standard, and you can then use new interface types ad-hoc.
- Again we pass the interface type as a property of an offer.



Standardising descriptions of property types

- **Alternatively we could agree ways of describing types.**
- **So you define an AST as a global standard, and then you are free to define your own IDLs in terms of it.**
- **Now we pass the interface type and a description of the IDL which it is described in as a property of an offer.**

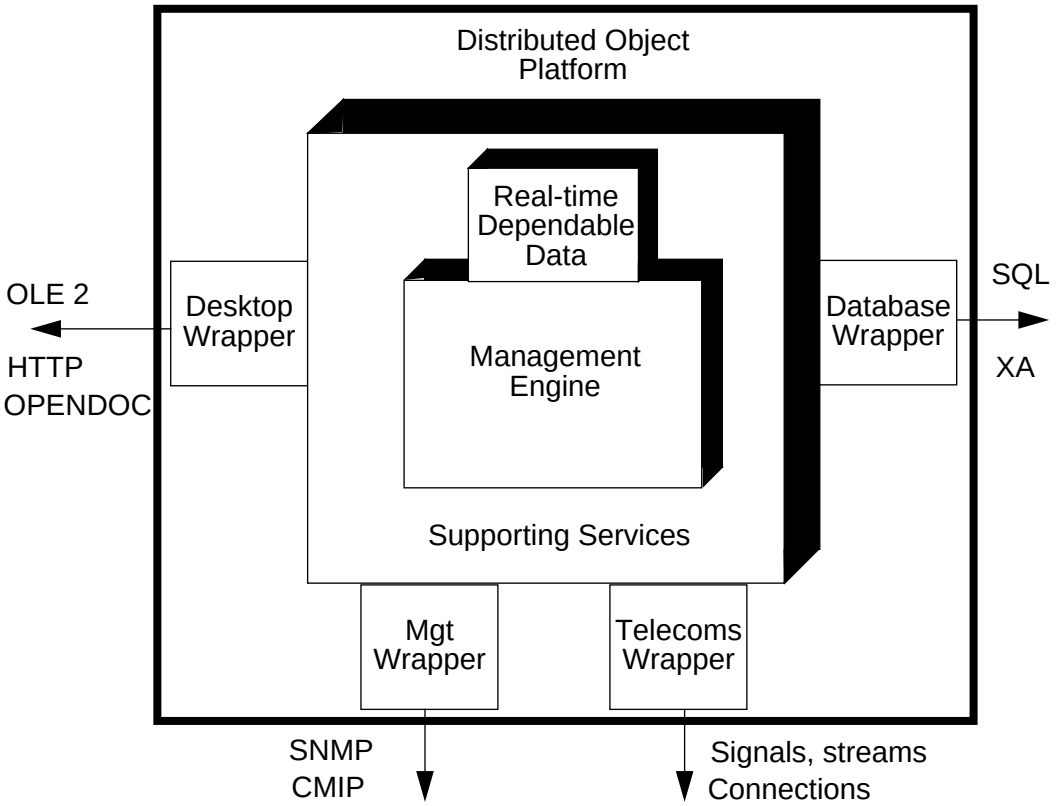


Abstract is best

- The best way forward is to use more abstraction in standards.
- This will enable us to:
  - Rely on standards which are stable enough to be usable.
  - Build mechanisms to handle the concrete in a general way.
  - Federate and manage diversity which we cannot control.
- Problems with abstract standards:
  - Harder to understand and apply.
  - They introduce a maintenance problem.
- The telling factor:
  - How do we define a more concrete standard?
  - Only in terms of a more abstract one!

A Management Engine

- **This view of a management engine, taken from [APM.1275] depicts technological diversity.**
- **Abstraction offers a good method for isolating management logic from this diversity.**



How do we tackle abstract information?

- We need:
- A conceptual framework:
  - For thinking about the uses of standards.
  - For deciding what standards to apply.
- A common approach to using and maintaining standards:
  - We use different standards for similar purposes.
- A common approach to building mechanisms to manage diversity:
  - information management and change control
  - mapping software
  - code generation tools
- The meta-information approach can give us these things.

So what is meta-information?

- Information about information.
  - A simple concept, with complex implications.
- Properties are meta-information about services:
  - E.g. interface types, protocols, OMNIPoint objects, QoS parameters.
- Descriptions of properties are meta-meta-information
  - E.g. IDL describes interface types, GDMO describes OMNIPoint objects, OSI standards describe protocols.
- Descriptions of descriptions are meta-meta-meta-information
  - E.g. AST for describing IDLs, XOJIDM mapping IDL and GDMO.
- Clearly, meta-information will not solve all problems. We need to investigate where it should be applied and identify its limits.

The way forward

- As work on management processes stabilises, more work on property modelling will be needed to apply results in practice.
- 1. Enhance the modelling of properties in the MatchMaker:
  - Include descriptive meta-information about property types and the mappings between them.
  - Investigate the maintenance problem caused by this redundancy.
- 2. Use this enhanced information in gateway prototyping:
  - Federate traders, clients and servers which use different property models.
  - Apply standardisation at different levels to investigate the trade-offs.
  - Attempt to generalise mapping software.
  - Investigate auto-generation of mapping software.
- 3. Identify other applications for meta-information.