

A Framework for Understanding and Classifying Ontology Applications



Michael Uschold

Adapted from talk co-authored by Rob Jasper
given at IJCAI99 Workshop

"Ontologies and Problem-Solving Methods:
Lessons Learned and Future Trends"

Describing Ontology Applications

Purpose of the ontology – intended benefits

Principal actors: who does what?

Where does ontology fit in architecture?

What exactly does it do and when?

Technology stack

Intended Benefits

Communication between humans

Interoperability among computer systems

Search

System engineering benefits:

- Flexibility, Flexibility and Flexibility
- Lean/Agile application development
- Enhanced reusability
- Reliability / consistency
- Easier maintenance

New things become possible

Actors / Roles

Ontology Author (OA)

Data Author (DA)

Application Developer (AD)

Application User (AU)

Knowledge Worker (KW)

Technology Stack

Language: OWL? Flogic? etc

Inference and Data Stores

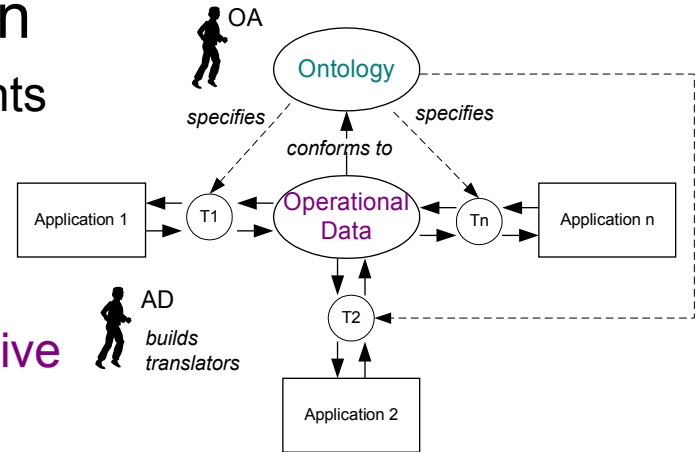
Services / Cloud

NEXT: generic architectures and variations

Ontology Application Scenarios

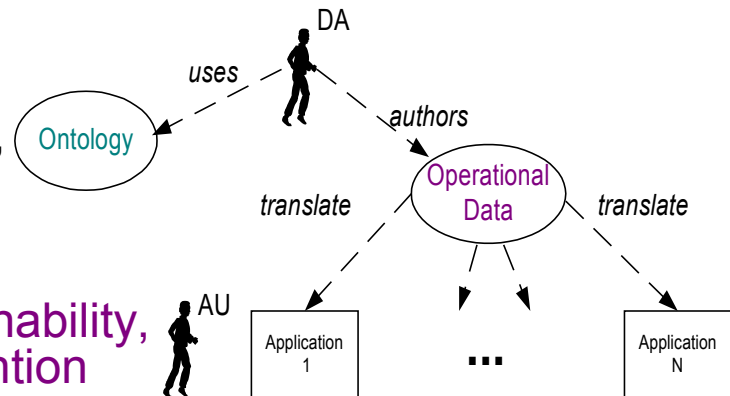
Common Access to Information

- information required by multiple agents
- expressed in wrong terms/format
- ontology used as agreed standard, basis for converting/mapping
- *Benefits: interoperability, more effective use/reuse of knowledge*



Neutral Authoring

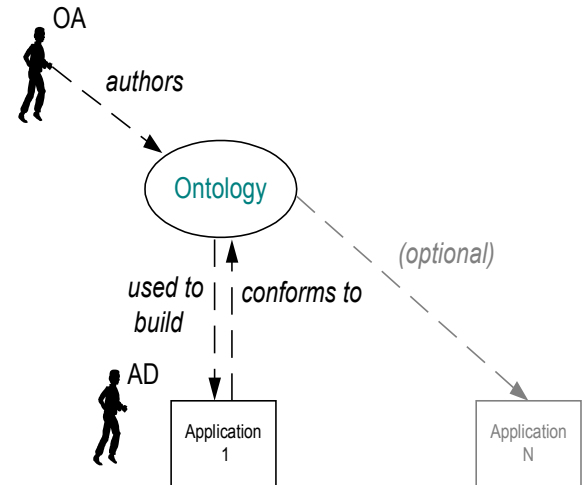
- artifact authored in single language, based on ontology
- converted to multiple target formats
- *Benefits: knowledge reuse, maintainability, long term knowledge retention*



More Application Scenarios

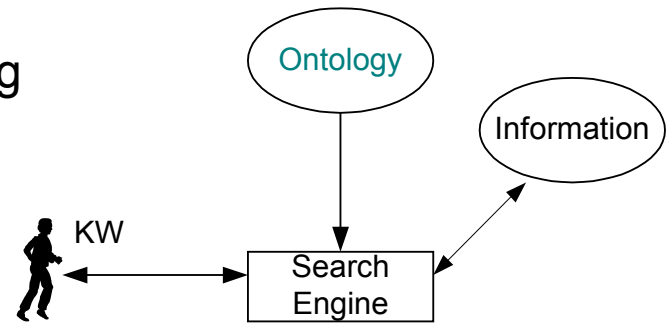
Specification for Software

- build ontology for required domain
- produce software consistent with ontology
 - manual or partially automated
- **Benefits:** documentation, maintenance, reliability, knowledge (re)use



Search

- Ontology used for semantic structuring of information in a repository
- **Benefits:** better information access



I: Common Access to Information

Human Communication

Data Access via Shared Ontology

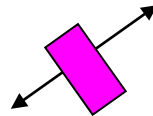
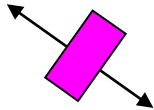
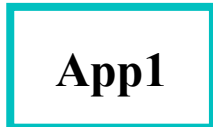
Data Access via Mapped Ontologies

Shared Services

Data Access via Shared Ontology



*Language &
vocabulary*



Reduce cost of access by
multiple applications.

Interoperability

Example: PIF / PSL

PIF / PSL *Time-consuming to create*

*Language &
vocabulary*

ILOG

App N

**Process
model**

IDEF3

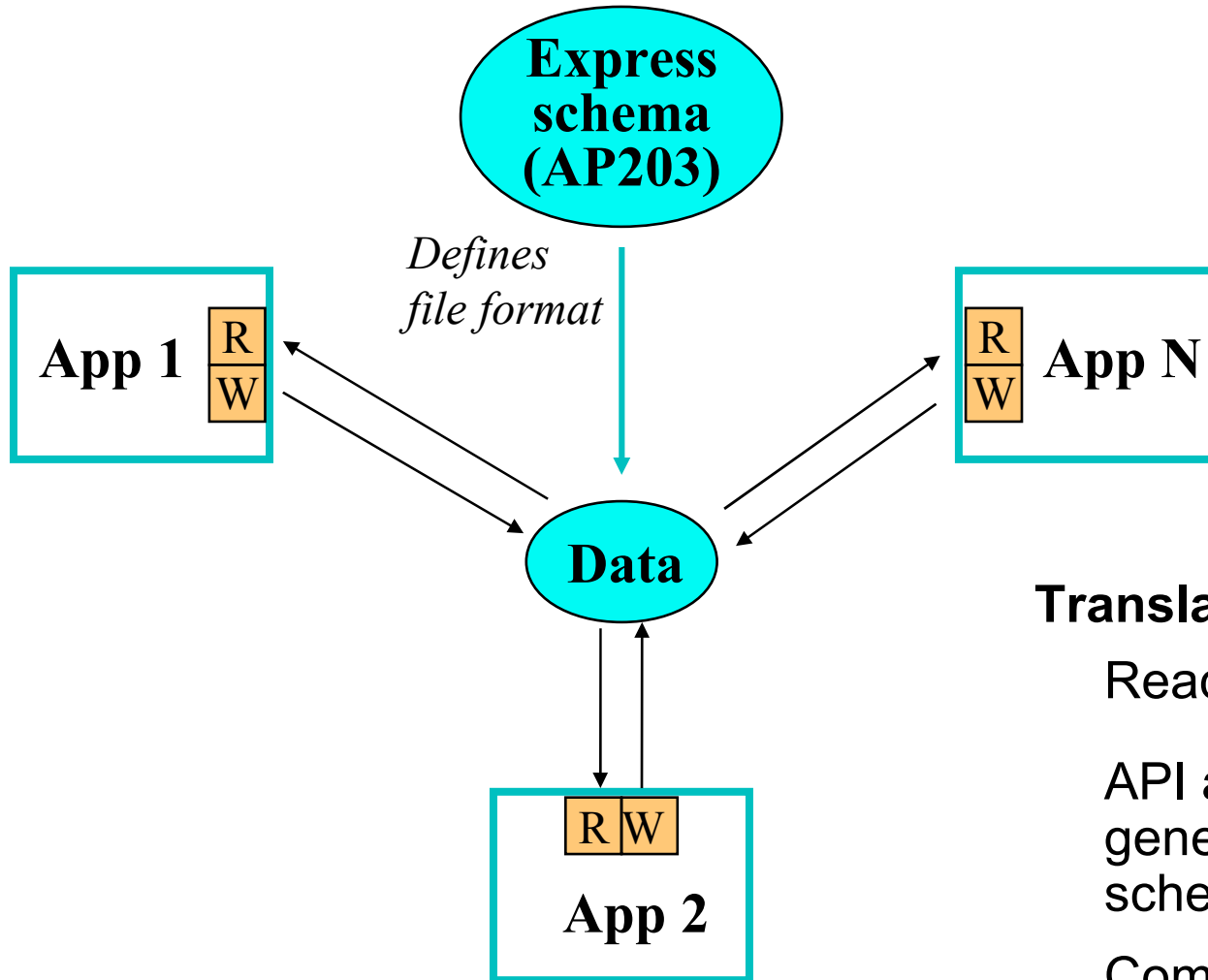
Funded by NIST

Rigorous logical formalism

2 Translators + Demonstrator

Research prototype, ongoing

Example: STEP / Express



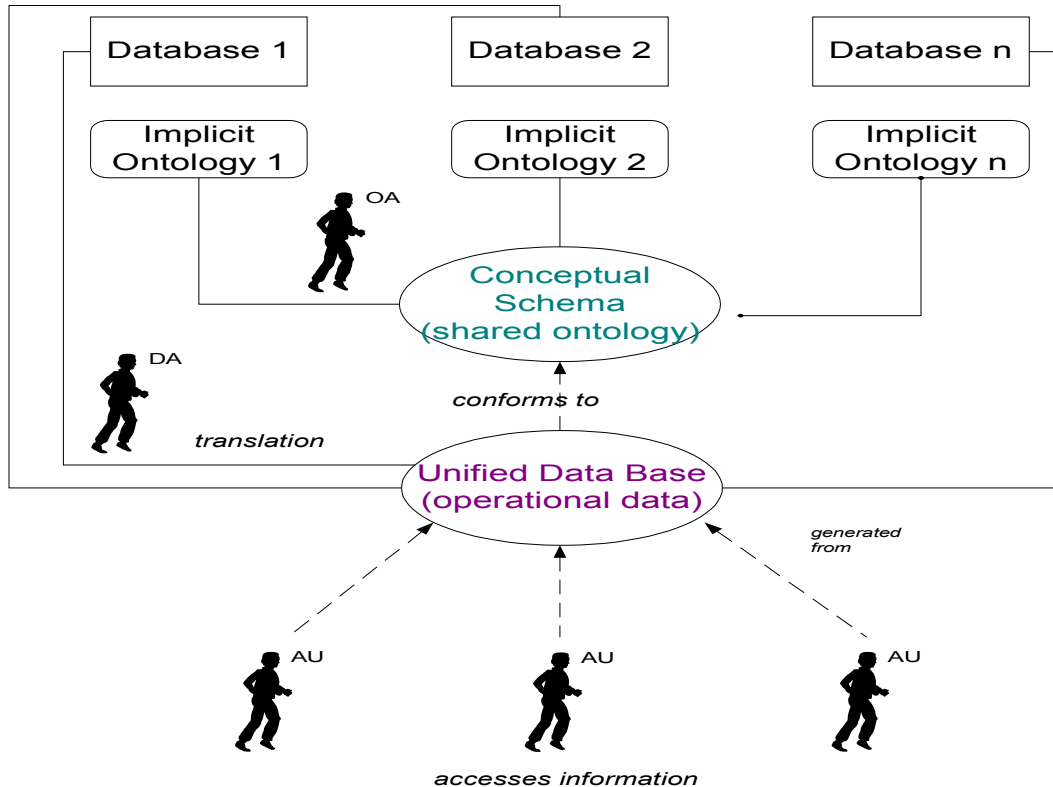
Translation:

Readers/Writers

API automatically generated from schema.

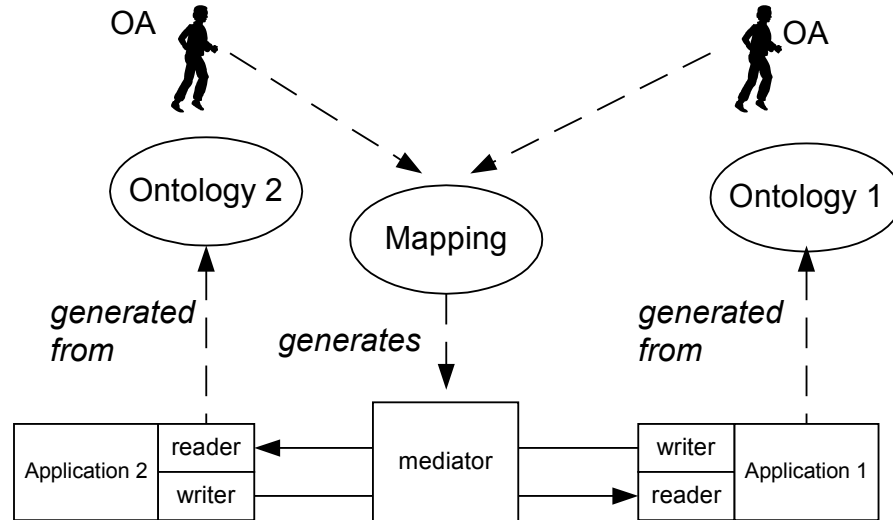
Commercial tools

Heterogeneous DBs: EcoCyc



- Commercial Application
- Uses Generic Frame Protocol with API

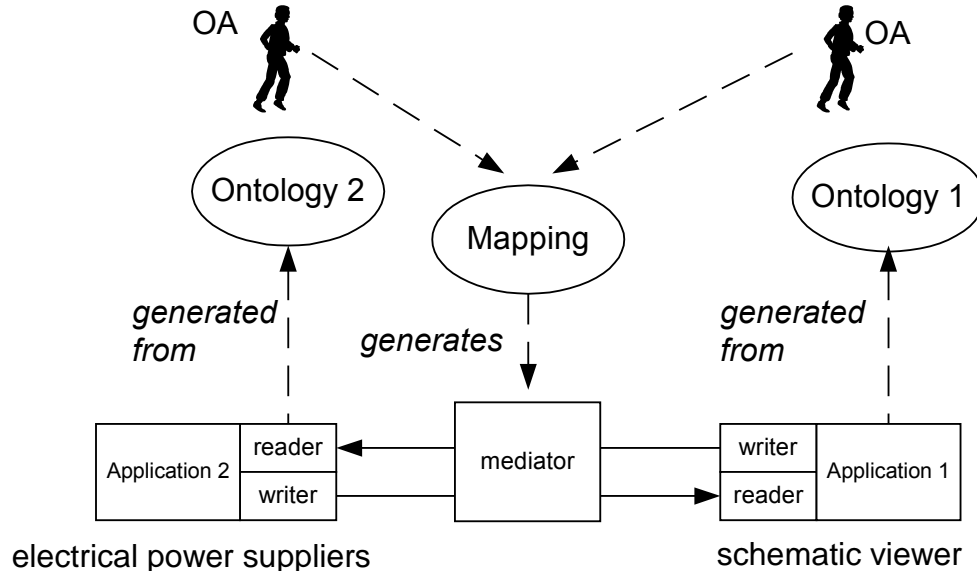
Data Access via Mapped Ontologies



Differences:

- no explicit shared ontology (no need to reach agreement)
- terms mapped from one ontology to another
- Supporting technologies:
 - parser generators
 - printers
 - mediators

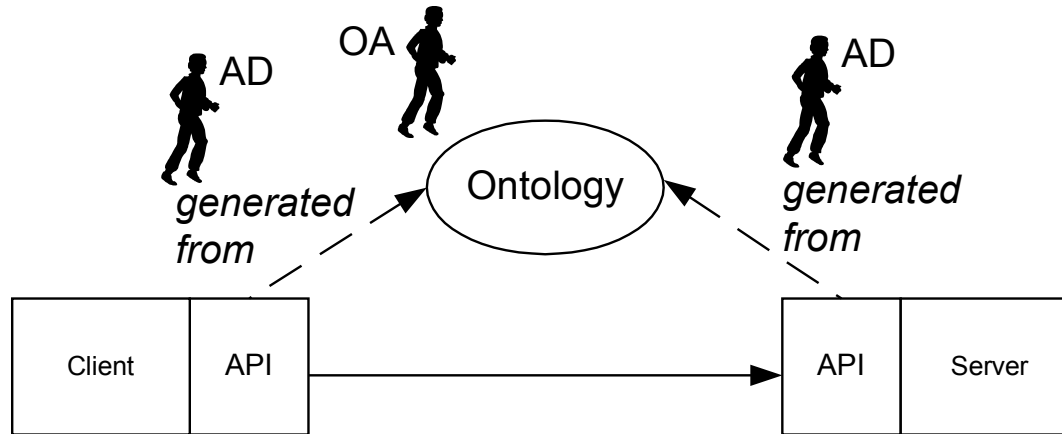
Data Access via Mapped Ontologies



Example:

- Ontologies represented using EXPRESS
- Developers agree on a mapping
- Represented using EXPRESS-X
- Mediator automatically generated
- Also for integrating heterogeneous databases?

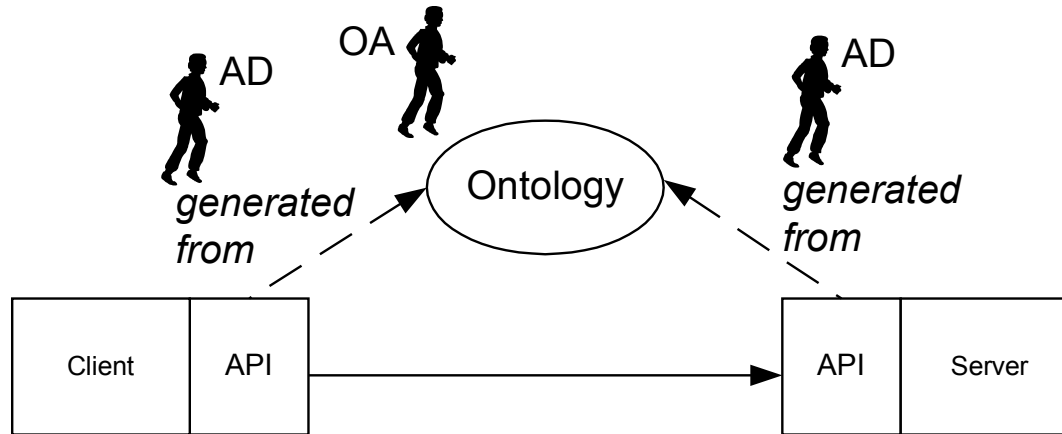
Shared Services



Motivation: neutrality

- language
- machine
- operating system
- location

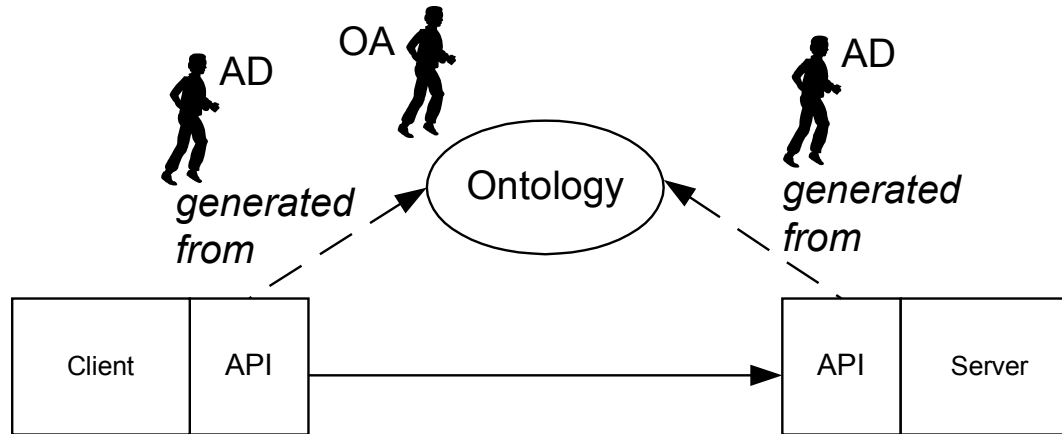
Shared Services



Example: CORBA

- OA creates ontology in some domain (e.g. product data)
- Ontology used to generate interface code for client & server
- Client applications use services independent of location, operating system, language or machine

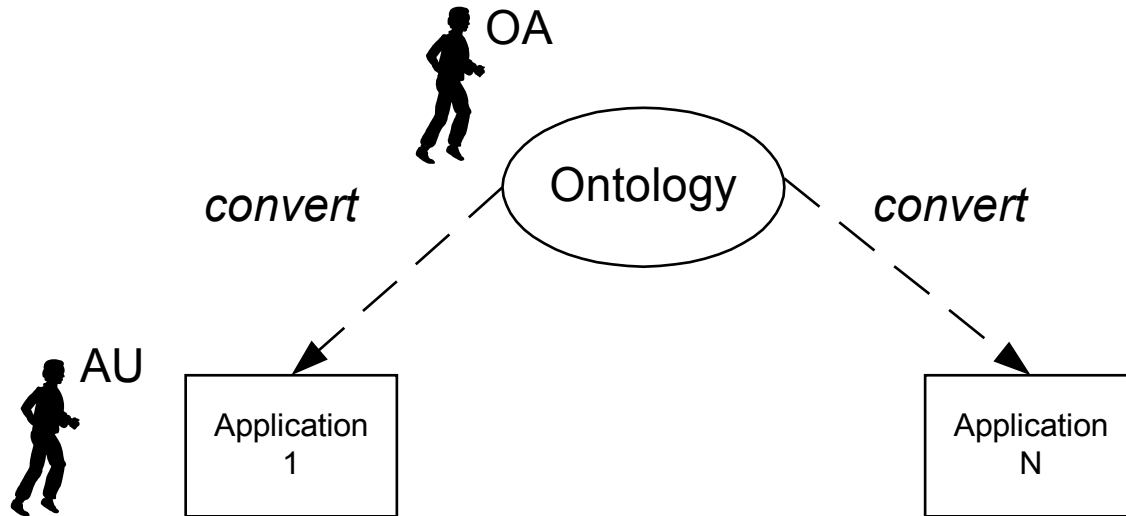
Shared Services



Supporting Technology

- application interface generators
 - parser generators & printers
- marshalling routines
- **Very Mature**

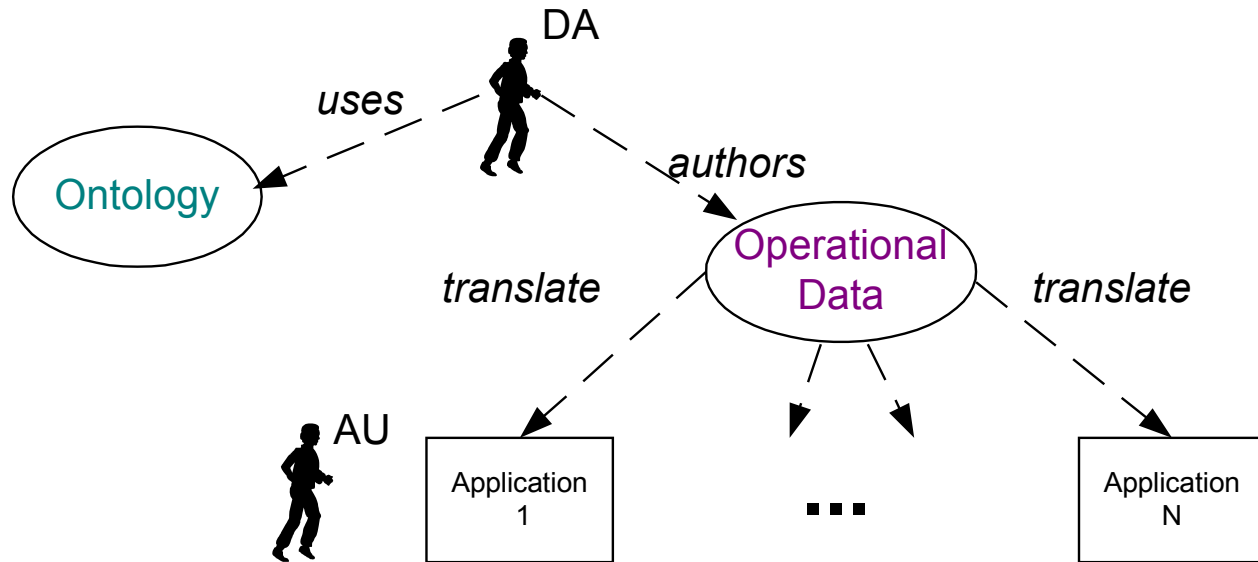
II: Neutral Authoring: Ontologies



Example:

- Ontolingua ontology, convert to Loom and Prolog

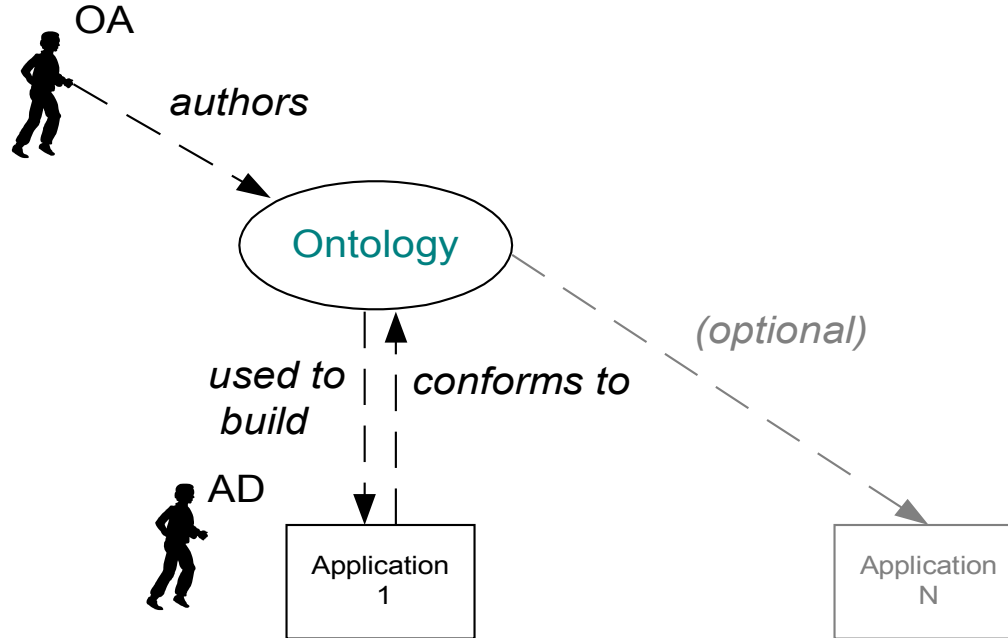
Neutral Authoring: Opnl. Data



Example:

- DA uses a *workflow ontology* to author a *workflow model*; tools translate data into format required by target applications
- DA uses an Express schema as a format for authoring product data, which can be read by many applications.

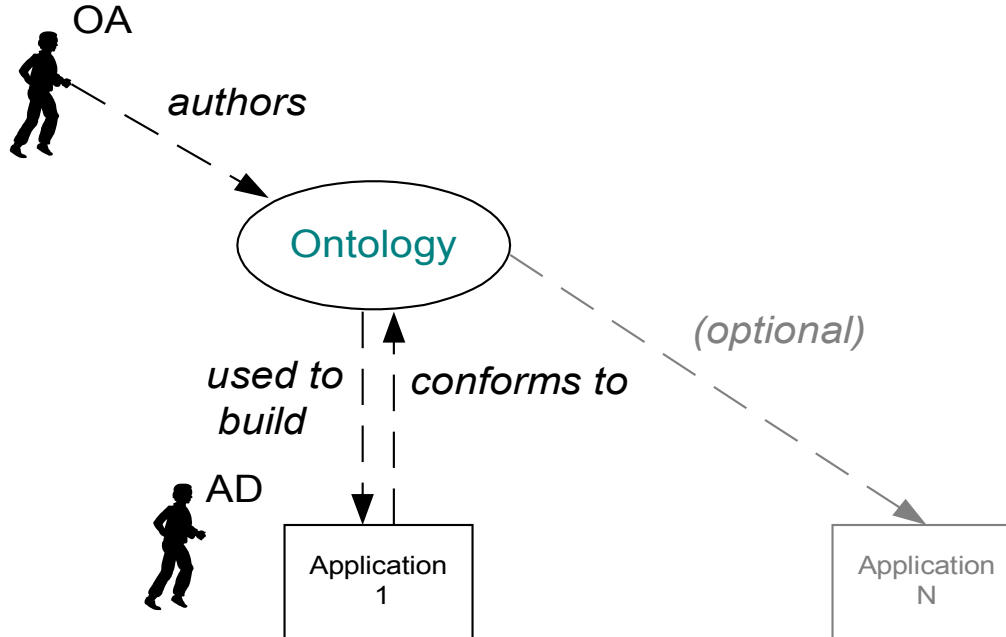
III: Ontology as Specification



Similar to Neutral Authoring, *but*:

- useful even if single application
- not translated per se, but guides development of and is manifest in the target application(s)

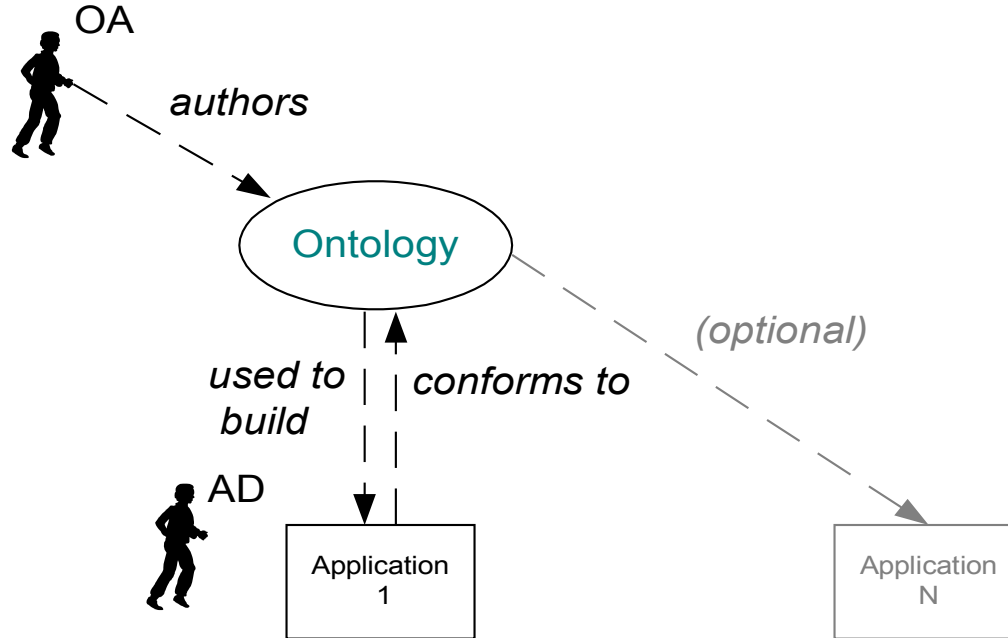
III: Ontology as Specification



Benefits:

- documentation & maintenance,
- reliability,
- knowledge (re)use

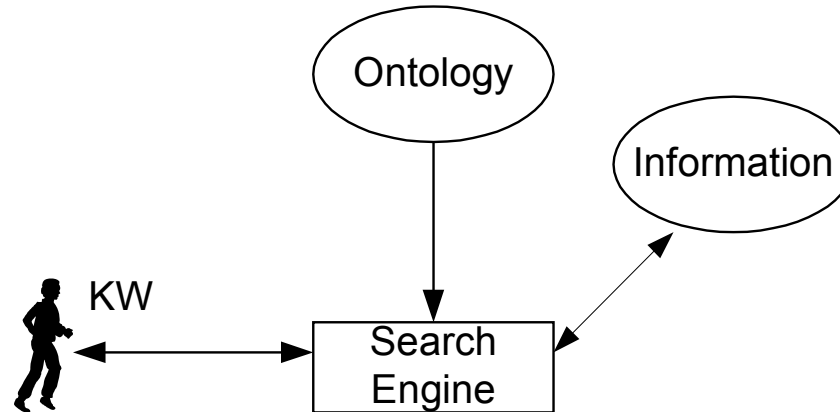
III: Ontology as Specification



Examples:

- CML as part of KADS methodology
- Data modeling tools (e.g. IDEF1X -> DB schemas)
- Protege: ontology is basis for KBS; guides knowledge acquisition

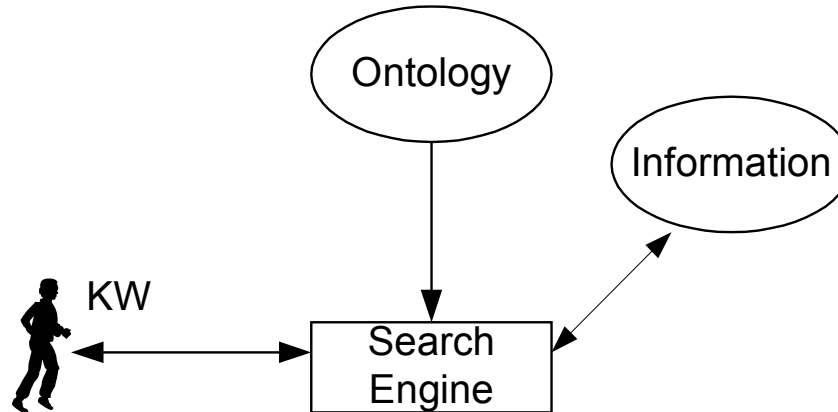
IV: Search



Motivation: locating artifacts

- Ontology used to semantically structure and/or index the repository
- Typically, simple taxonomies used, minimal semantics

IV: Search



Many Many Examples

- Yahoo!, Yellow pages, E-Commerce..., XML DTDs
- Needs whole (sub)framework to clarify work in this area.