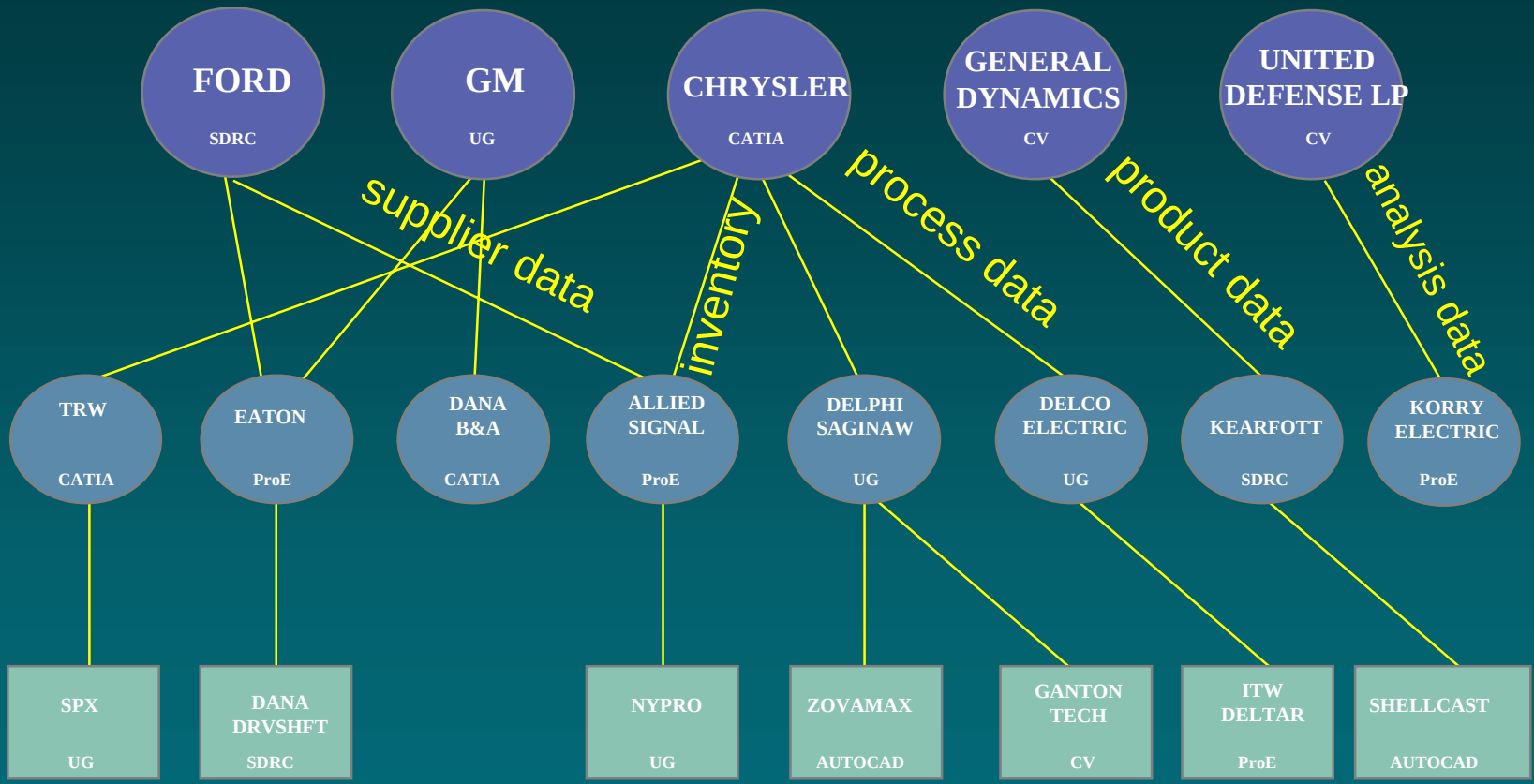
Interoperability Standards in the Future

A discussion of some issues in systems integration

Steven Ray, Ph.D.
Chief, Manufacturing Systems Integration Division
Manufacturing Engineering Laboratory

May 23, 2003

The Integration Nightmare



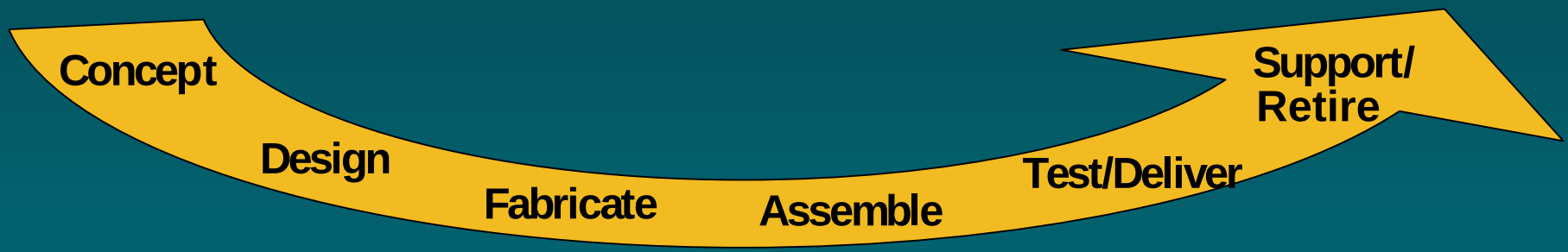
What is the solution?

- ✗ Point-to-point customized integration among the software systems supporting product and process development
  - Expensive to build & maintain
- ✗ Mandating specific vendor software systems among supply chain partners
  - Pushes interoperability problems lower in the supply chain - it doesn't solve them
- ✓ Using neutral standards
  - Standards for information technology are technical rules providing the foundation that enable interconnected systems to work across activities, organizations, and geographic locations.

IT Standards Enable Interoperability

STEP - STandard for the Exchange of Product Model Data - ISO 10303

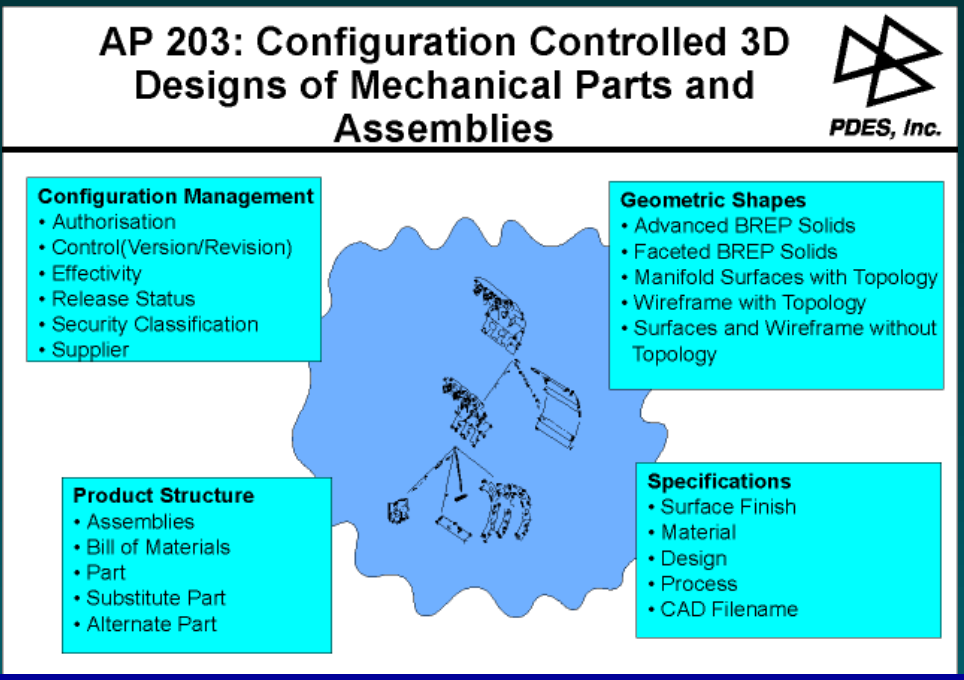
The international standard which provides an unambiguous, computer-interpretable definition of the physical and functional characteristics of a product throughout its life cycle

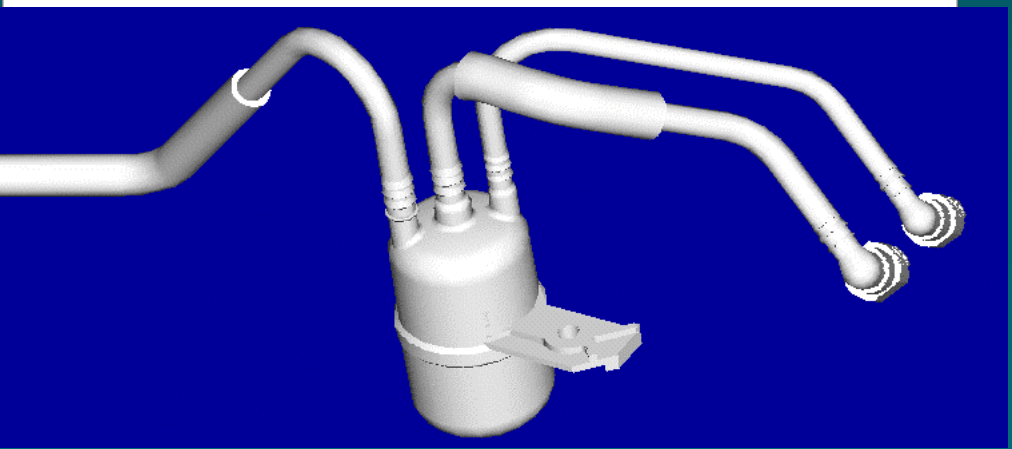


Product Life Cycle

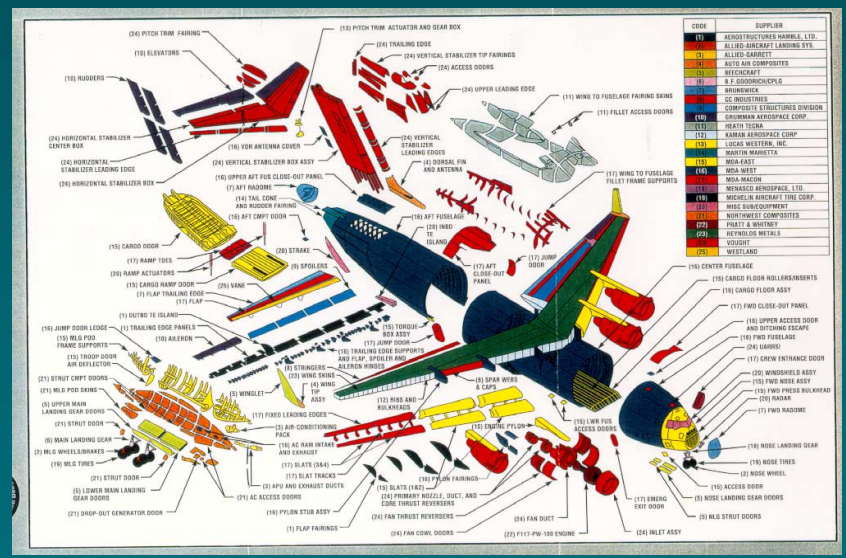
As Defined Configurations	As Planned Configurations	As Built Configurations	As Maintained Configurations

STEP in Production





- Boeing Commercial Aircraft
- Boeing CSTAR
- Delphi Automotive Systems
- Lockheed Martin
- NASA
- Motorola
- ….



Beyond Product Data...

- Manufacturing plans
- Material specifications
- Process specifications
- Analysis data
- Cost data
- Supplier information
- Inventory management data
- Supplier delivery data
- Demand forecasts
- Production status
- Pricing information
- Warranty information
- Quality information
- Product genealogy
- Simulation data
- ...

The Many Dimensions of Systems Integration

Integration Problem Categories (1)

Technical

- connection conflicts
  - A software component must provide data to an application whose only data entry interface is a graphical user interface (GUI) intended for human input.
- syntactic conflicts
  - One system uses ASN.1 to represent the data schema and the Basic Encoding Rules to represent the corresponding data; the other component uses XML Schema to represent the data schema and corresponding XML representations of the data.
- control conflicts
  - "too many leaders" – Both components expect to be the "client" component, invoking a service (operation, procedure) provided by the other component; neither expects to be a "server" and respond to the other's requests.

Integration Problem Categories (1)

Technical (continued)

- quality-of-service conflicts
  - A component is expected to operate in a real-time system and respond within certain time constraints.
- data consistency conflicts
  - The manufacturing scheduler asks the database system for the location of the materials container for lot 604, finds that it is still in an in-buffer for an automatic storage and retrieval system (ASRS), and sends a command to the ASRS to cancel the "store" command for that container, but the ASRS controller reports that the command has already completed — the ASRS database transaction to update the location occurred between the two actions of the scheduler.

Integration Problem Categories (2)

Semantic

- conceptual factorization conflicts
  - Continuous state-based decision making vs. discrete event-based decision making.
- conceptual scope conflicts
  - One component manages versions of documents while the other does not have a "version" concept in its document identification model.

Integration Problem Categories (2)

Semantic (continued)

- interpretation conflicts
  - Components assume different units (e.g., metric vs. English measure) for measurement values that don't specify the unit.
- reference conflicts
  - One component identifies the Part by item number on the order form; the other identifies it by stockroom and bin number. (different relationships, extended properties)

Integration Problem Categories (3)

- Functional
  - functional model conflicts
    - Nobody's job: An email exploder expects messages to be assigned Message IDs by the mail relay component. However, the targeted mail relay treats messages lacking Message IDs as invalid and ignores them. It is nobody's job to assign the Message IDs, so these components cannot interact to distribute email.
  - functional scope conflicts
    - A relational database system interprets a DELETE operation to delete only the row that represents the object named, but an object-oriented database system interprets a DELETE operation to delete the object named and all the objects that are dependent on it. If the requester was expecting the object-oriented behavior, and the performer is a relational database, objects which should have been deleted will still appear in the database. If the requester was expecting the relational behavior, it may subsequently attempt to make new associations for objects which have been deleted.

Integration Problem Categories (3)

- Functional (continued)
  - intention (application scope) conflicts
    - A PDM system loses some information some of the time when exchanging information with suppliers. The integration engineer used the "Note" feature for all text extracted from some standard field of an exchange file. The PDM designer expected the "Note" feature to be used for "annotations" to CAD models.
  - embedding conflicts: configuration, conditioning
    - When the behavior of the component is affected by the integrated system in such a way as to produce unexpected and undesirable results.

Integration Problem Categories (4)

- Qualitative
  - security concerns
  - correctness, credibility and optimality concerns (data quality)
  - timeliness concerns
  - reliability concerns
  - version conflicts

Integration Problem Categories (5)

- Logistical
  - Trust (third party authentication, credibility, disclosure, abuse)
  - Competition (auctions, dispatchers, brokers)
  - reliability and failure recovery
  - flexibility, ability to change
  - cost

So what's next?

Some Overarching Issues

- Need for more rigor (less ambiguity) in exchange standards
- Rapid growth in the number of standards needed

The pursuit of rigor in data standards

Old-style (most common) standards specifications: (e.g. ISO 14258, Requirements for enterprise-reference architectures and methodologies)

"3.6.1.1 Time representation

If an individual element of the enterprise system has to be traced then properties of time need to be modeled to describe short-term changes. If the property time is introduced in terms of duration, it provides the base to do further analyses (e.g., process time). There are two kinds of behavior description relative to time: static and dynamic."

Data-model standards (e.g. ISO 10303-41, Product Description and Support)

```
ENTITY product_context
  SUBTYPE OF (application_context_element);
  discipline_type : label;
END_ENTITY;
```

Semantic-model standards (e.g. ISO 18629-11, PSL Core)

```
(forall (?t1 ?t2 ?t3)
      (=>    (and    (before ?t1 ?t2)
                     (before ?t2 ?t3))
             (before ?t1 ?t3)))
```

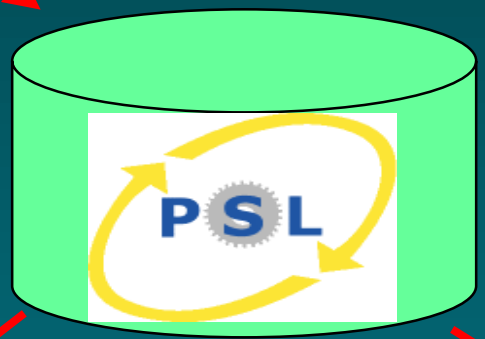
The Process Specification Language (PSL)

Process Modeler

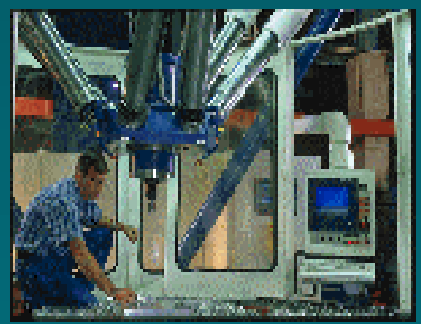
(ProCAP / KBSI)

Process Planner

(MetCAPP/Agiltech)



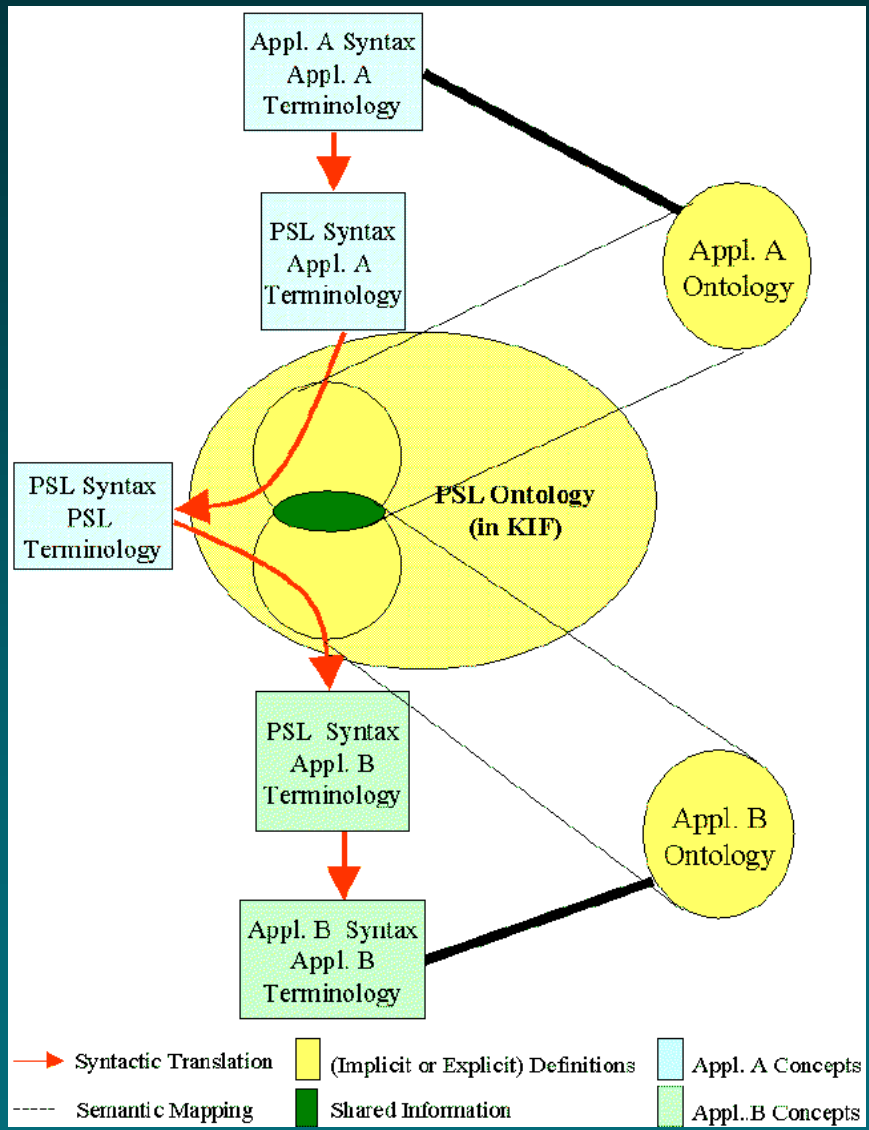
Simulator (Quest / Dessault)



Scheduler

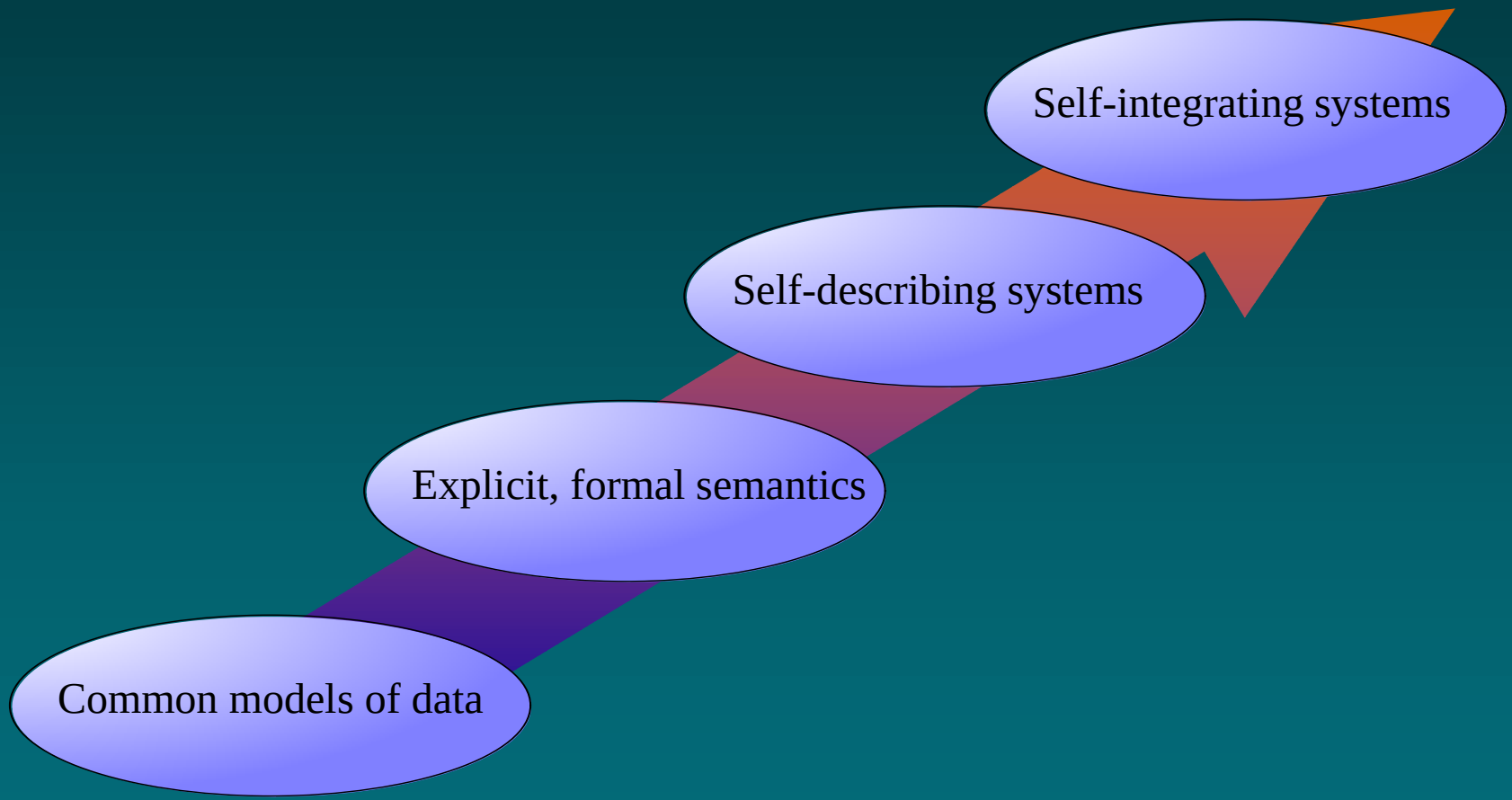
(ILOG Scheduler)

How Does PSL Work?

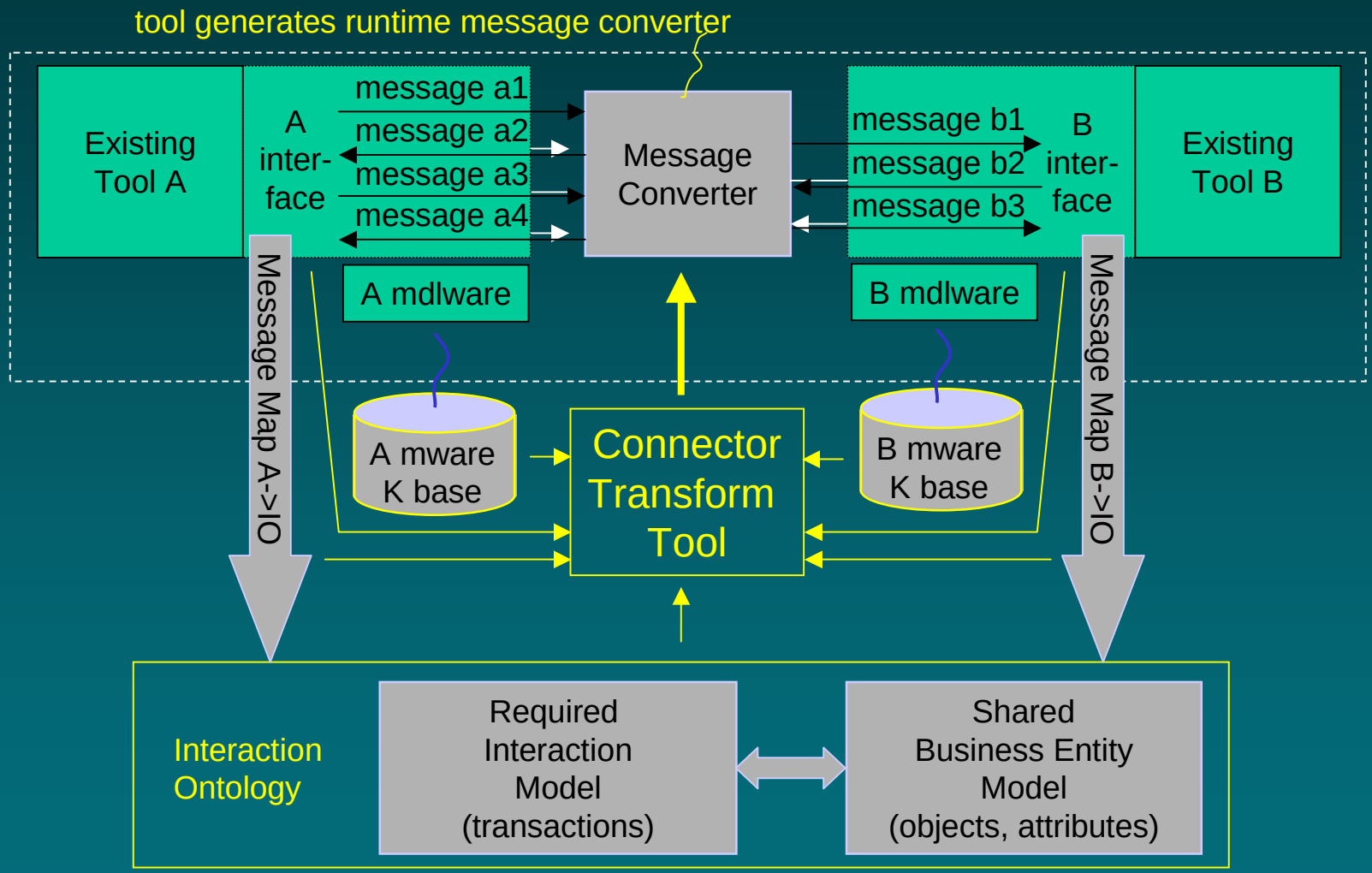


See http://www.nist.gov/psl

Evolution of Integrated Data Exchange



Automated Methods for Integrating Systems project



Technical Picture

- Systems integration is hard
- Interoperability continues to grow as a problem among increasingly IT-dependent systems
- Rigorous information exchange standards are becoming even more important
- A semantic approach offers a rigorous solution to the next generation of interoperability problems
- A semantic foundation also offers a way out of a race we can't win - trying to keep up with the pace of standards needs