Data Exchange Standards and Ontologies for Engineering

How to make the best use of both

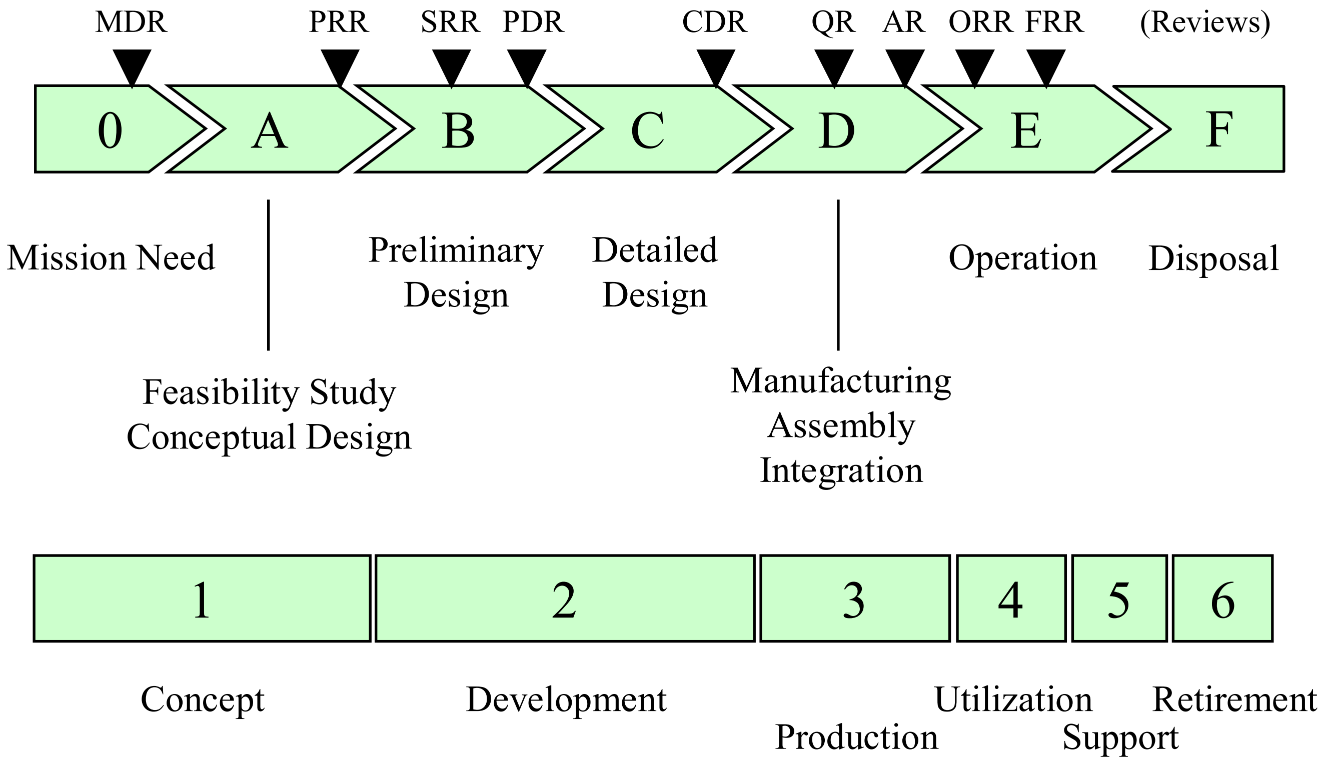
Hans Peter de Koning

(ESA/ESTEC, Noordwijk, The Netherlands)

Why do we need open standards based data exchange and sharing

- Essential for efficient and cost-effective development and operation
- Design, analysis and simulation tools for each of the individual engineering disciplines are quite mature today
- A next major efficiency improvement in the development of complex (space) systems needs to come from much better, easy-to-use and reliable integration of computer aided engineering tools and methods across disciplines, system breakdown, supply chain
- All analysis and simulation models need to be linked into the core system requirements database, functional breakdown, architectural design, product structure
- Support for multi-disciplinary design optimization and multi-physics
- Support for distributed project teams

System Life Cycle

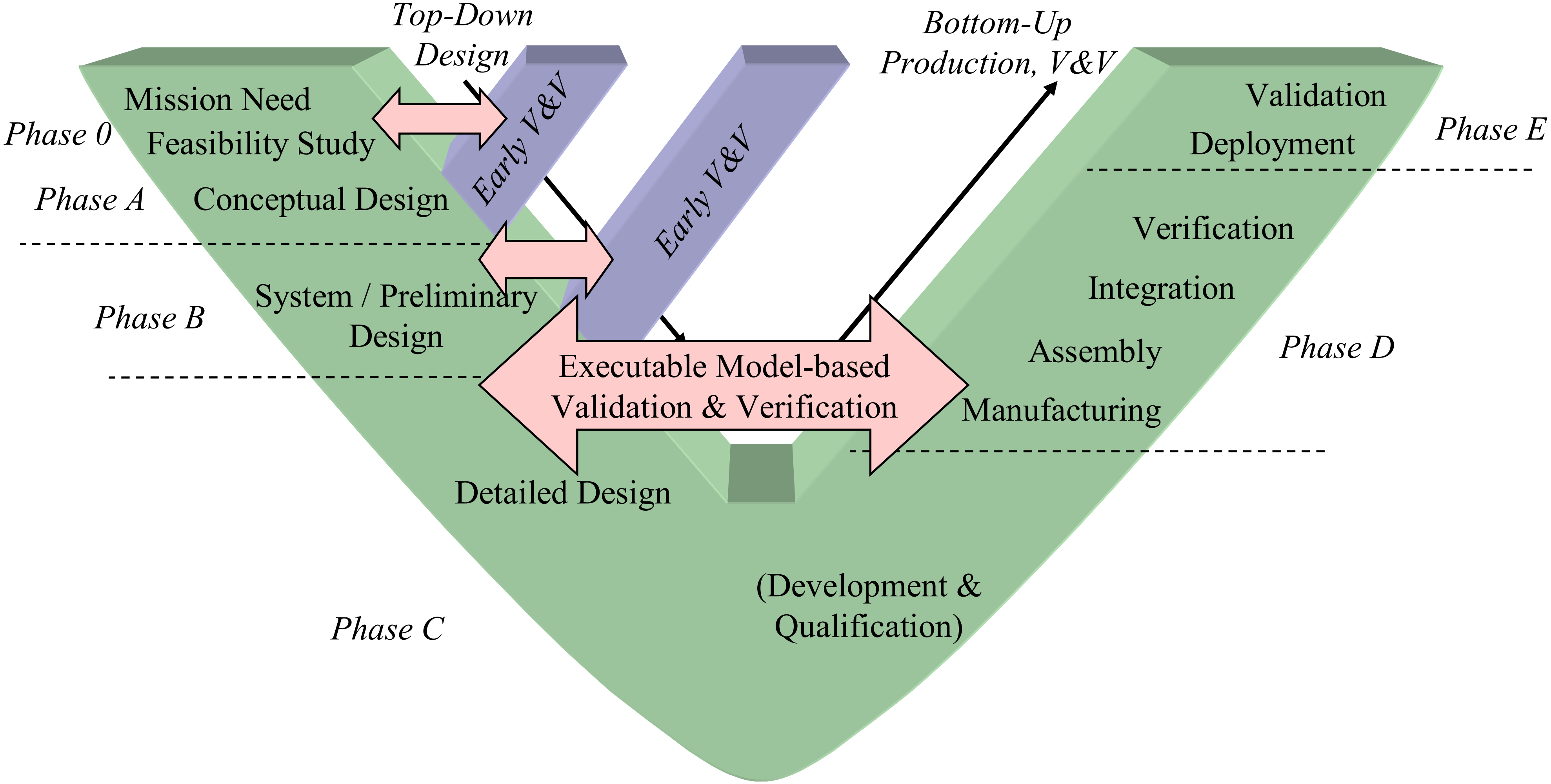


ESA Space System
Life Cycle Phases
(Similar to NASA's)

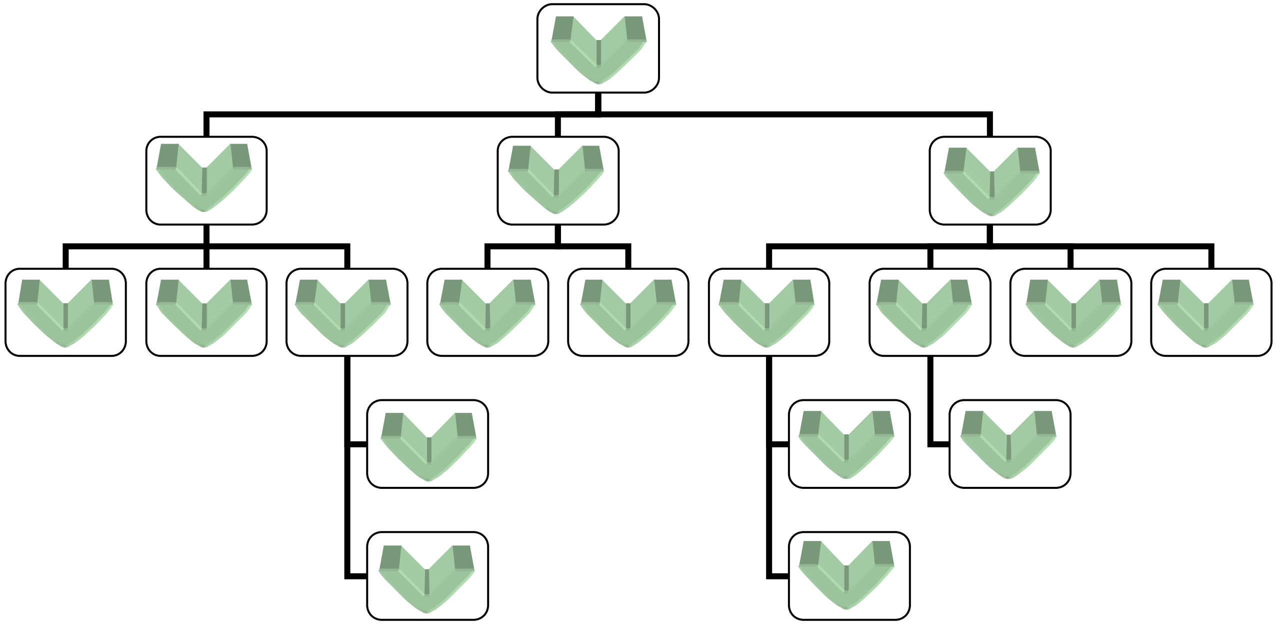
ISO/IEC 15288
System Life Cycle Stages

Source: ISO/IEC 15288:2002 Systems engineering — System life cycle processes (Annex B)

Trend: System Engineering "V" with Model-based Validation & Verification

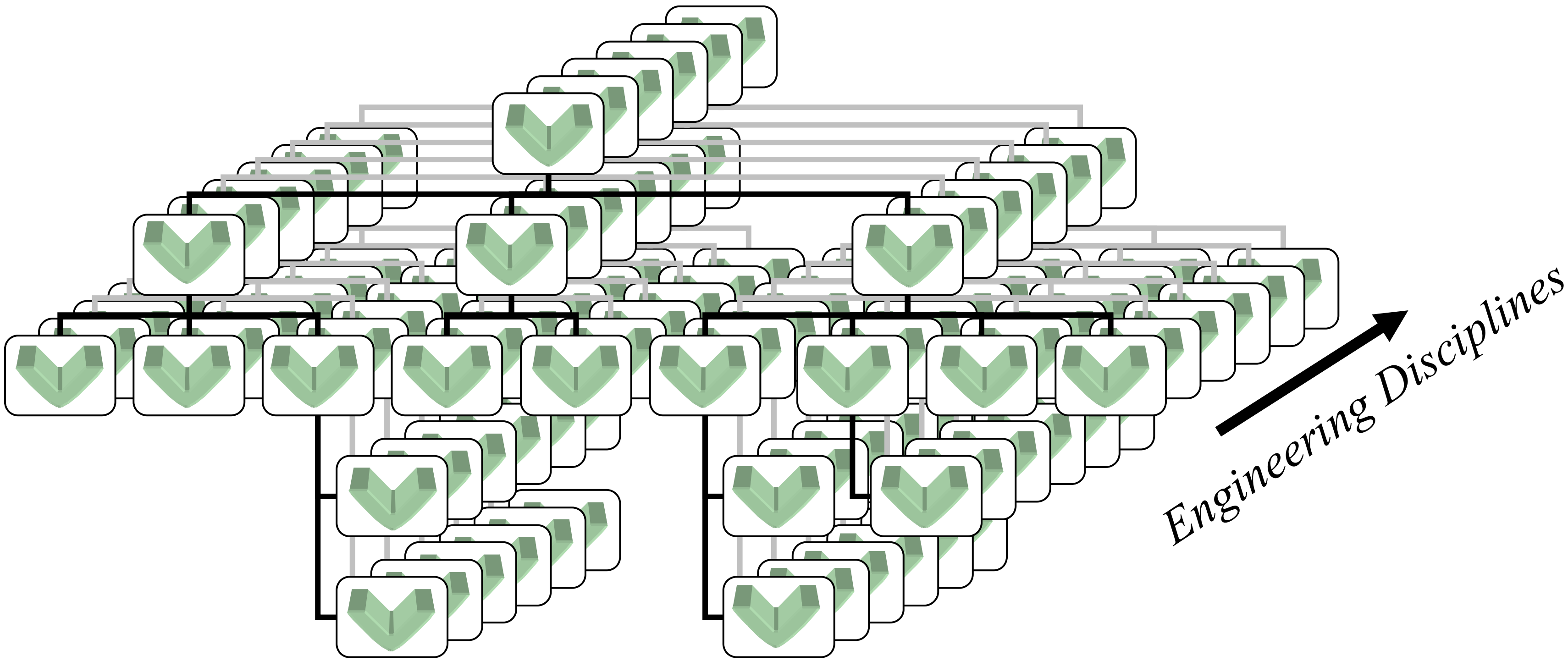


"V" applied recursively at all subsystem/subcontractor levels throughout the supply chain



System of Systems pattern: A *subsystem* at one level is a *system* at the next lower level

"V" applied recursively at all subsystem/subcontractor levels and for all engineering disciplines



Complex system & Large (international) supply chain & Multiple disciplines =

Very large number of (process-) interfaces

Holy grail of exchanging & sharing data (1)

- Seamless and total interoperability
  - Across disciplines, organizations, system levels, modelling methodologies, tools
- Flexible and precise formal open standards
- Reliable and affordable
- Affordable timely implementation and rigorously verified interfaces
  - Requires high quality public testsuites with adequate coverage
- Future proof and stable for long term archiving
  - May include open source middleware

Holy grail of exchanging & sharing data (2)

- Scaleability from small messages to full design/analysis/test/operation datasets
- Pervasive <u>and</u> standardized configuration control / versioning
- Minimal loss of information <u>and</u> common denominator between different (classes of) tools
- Easy-to-use/easy-to-implement and spanning many disciplines
- Support for white-box and black-box (degenerated/encrypted) data
  - Coming from genuine business needs and IPR protection

"Classical" Engineering Data Standards and Ontologies

- "Classical" engineering data exchange/sharing standards
  - Scope is usually a relatively specific and confined end-user problem
  - Terminology from software engineering (OO and/or ER)
  - Explicit, often detailed formal data model or just a file format
  - Reflection capabilities depend on implementation programming language
- Ontologies
  - Scope is often a "grand" data sharing problem for a complete industrial/scientific sector
  - Terminology shows scientific background, coming from philosophy, linguistics, artificial intelligence
  - Simple core data model allowing to state a large number of 'facts'
  - Built-in extensibility
  - Allows automated reasoning (inferencing) and has built-in reflection

Ontology spectrum

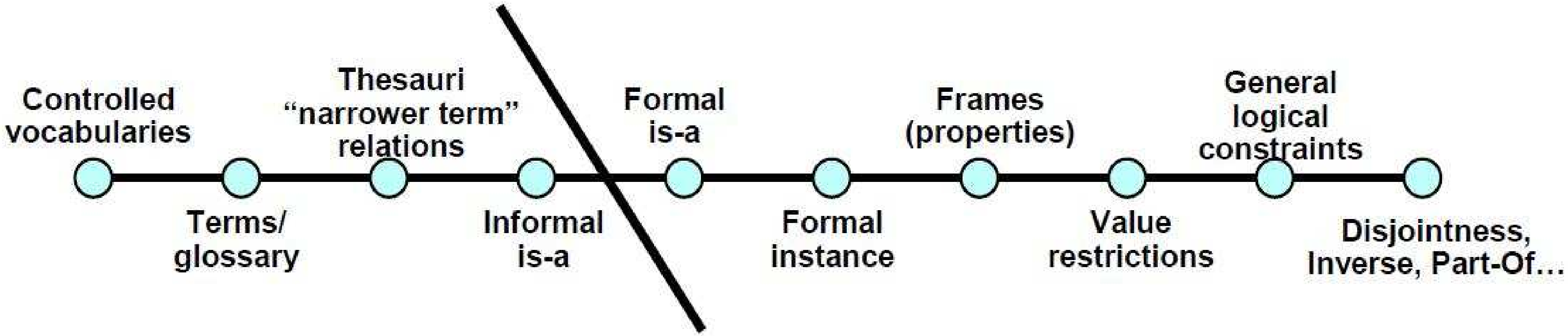


Figure 15, Categorization of ontologies according to Lassila and McGuinness

Source: PhD thesis Andries van Renssen

Terminology

Entity-Relationship	Object-Oriented Software Engineering	RDF/OWL (Semantic Web)	Description Logics	Frame Systems
entity, datatype	class, datatype	class, datatype	class, concept	frame, schema
object	instance, object	resource, individual	instance, individual	frame, instance, individual
attribute, relationship	attribute, instance variable, data member	property	role	slot
value	value	property value	filler	filler

Layers in different standard families

Standard family	**ISO 10303 (STEP)**	**W3C XML**	**W3C Semantic Web**	**W3C Ontology**	**OMG UML/MDA**
Origin	Mechanical engineering	Structured web data	Structured web data with meaning	Structured web data capturing knowledge	Software engineering
Data structure definition	ISO 10303-11 EXPRESS	DTD XML Schema	RDF Schema (uses XML Schema datatypes)	OWL (Lite/DL/Full) (builds on top of RDF Schema)	MOF UML/OCL XMI
File exchange	ISO 10303-21 clear text encoding ("STEP file") ISO 10303-28 XML encoding ISO 10303 Binary (in progress, possibly HDF5)	XML Unicode encoding (e.g. UTF8) XML/Binary (progress unkown)	RDF-XML	OWL-XML	-
Data access API	ISO 10303-22 SDAI ISO 10303-23 C++ ISO 10303-24 C ISO 10303-27 Java	DOM SAX (many open source)	RDF library (various open source)	OWL library (various open source e.g. Jena)	Generated from UML model QVT

Pros and Cons

	Advantages	Disadvantages
"Classical" engineering data exchange/sharing standards	Terminology closer to engineering domains / database technology Explicit scope, therefore implementations are in principle exhaustively verifiable	Rigid relatively inflexible formal structure, costly to extend Complex data model, often steep learning curve for implementers
Ontologies	Flexible, extendible Simple core data model Supports automated reasoning Lowers integration barriers Strong support from W3C / semantic web / open source software	Terminology "foreign" to engineering domains, therefore learning curve Implicit, extendible scope, therefore implementations are in principle not exhaustively verifiable

Questions

- How to reap benefit of ontology based data standards, while keeping data exchange/sharing implementations affordable?
- How to test correctness and completeness of implementations?
  - Complete mapping of native (source or target) data structure to ontology
  - Exhaustive testing possible? Sufficient test coverage possible?
- How to overcome terminology issues and differing capabilities within data standardization technologies itself?
- How to manage public (or community) reference data libraries / dictionaries / upper ontologies?
  - Public funding? Management authority? Peer reviews? Conflict resolution?
  - Long term archiving? Backwards compatibility?
- How to address IPR/confidentiality issues? How fine-grained access control?