Course 7

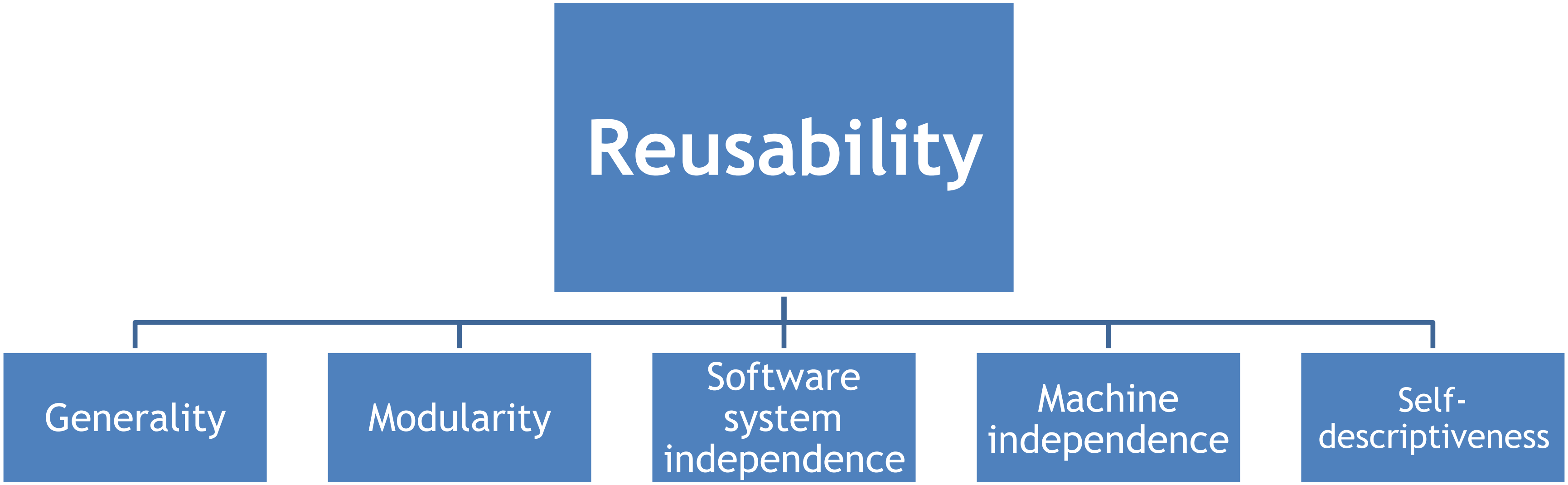
Reusability, interoperability

Reusability

Reusability

- Definition: *Extent to which a program can be used in other applications* ***- related to the packaging and scope of the*** *functions that programs perform*.

- Impact:
  - Measured: design + implementation
  - Realized: transition



Reusability

Generality

Modularity

Software system independence

Machine independence

Self-descriptiveness

Reuse what?

- Architecture
- Source code
- Data
- Design
- Documentation
- Templates
- GUI
- Requirements
- Test cases

language

paradigm

methodology

Reusability

- Paradigms:
  - OOP
  - Component based programming
- Languages: OOL
  - C++
  - Java

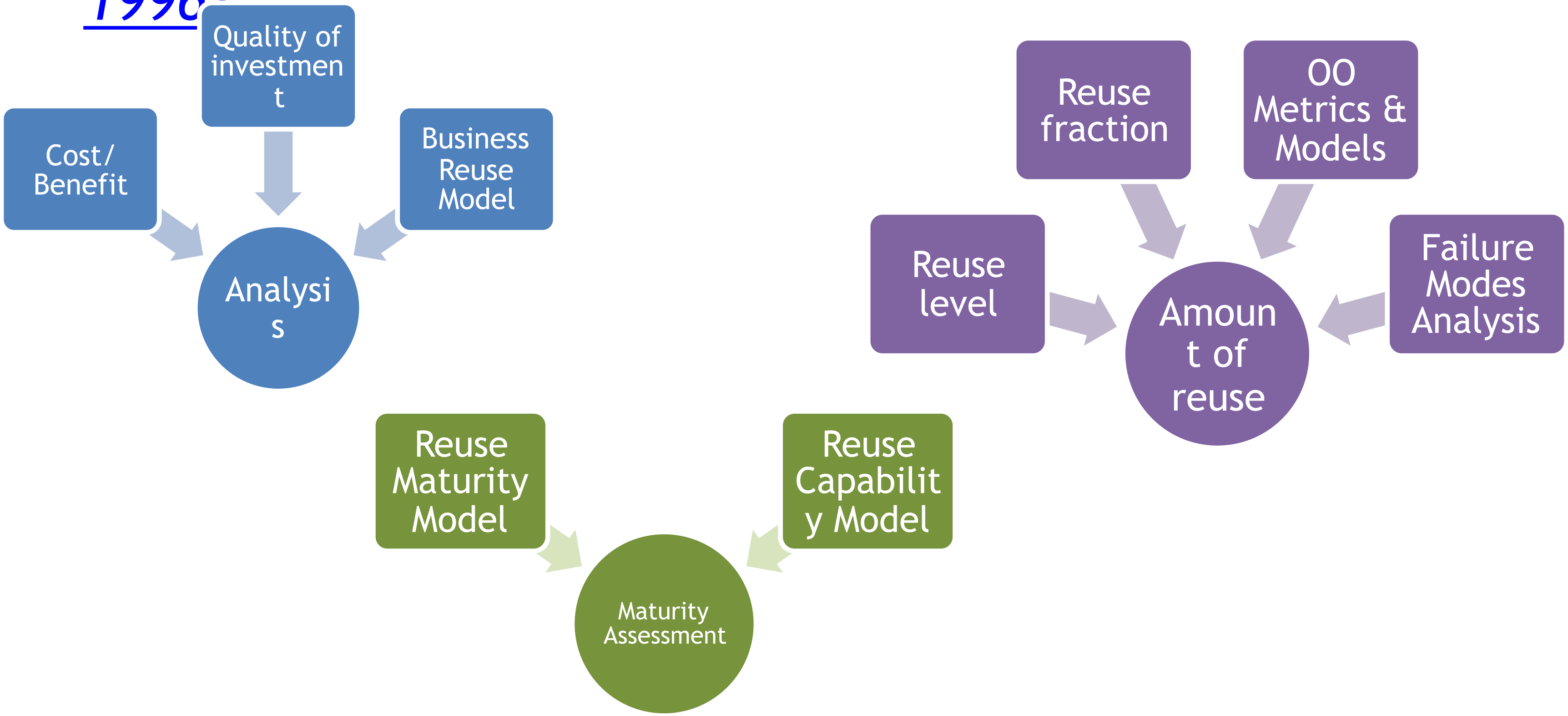
STL, templates,
multiple inheritance

interfaces,
RPC

- Methodology
  - Component Based Software Development
  - MDD

Reuse: metrics & models

[W. Frakes, C. Terry – Software Reuse: Metrics and Models, ACM Computing Surveys 28(2), pp. 415-435, 1996]



Cost/ Benefit

Quality of investment

Business Reuse Model

Analysis

Reuse fraction

OO Metrics & Models

Reuse level

Failure Modes Analysis

Amount of reuse

Reuse Maturity Model

Reuse Capability Model

Maturity Assessment

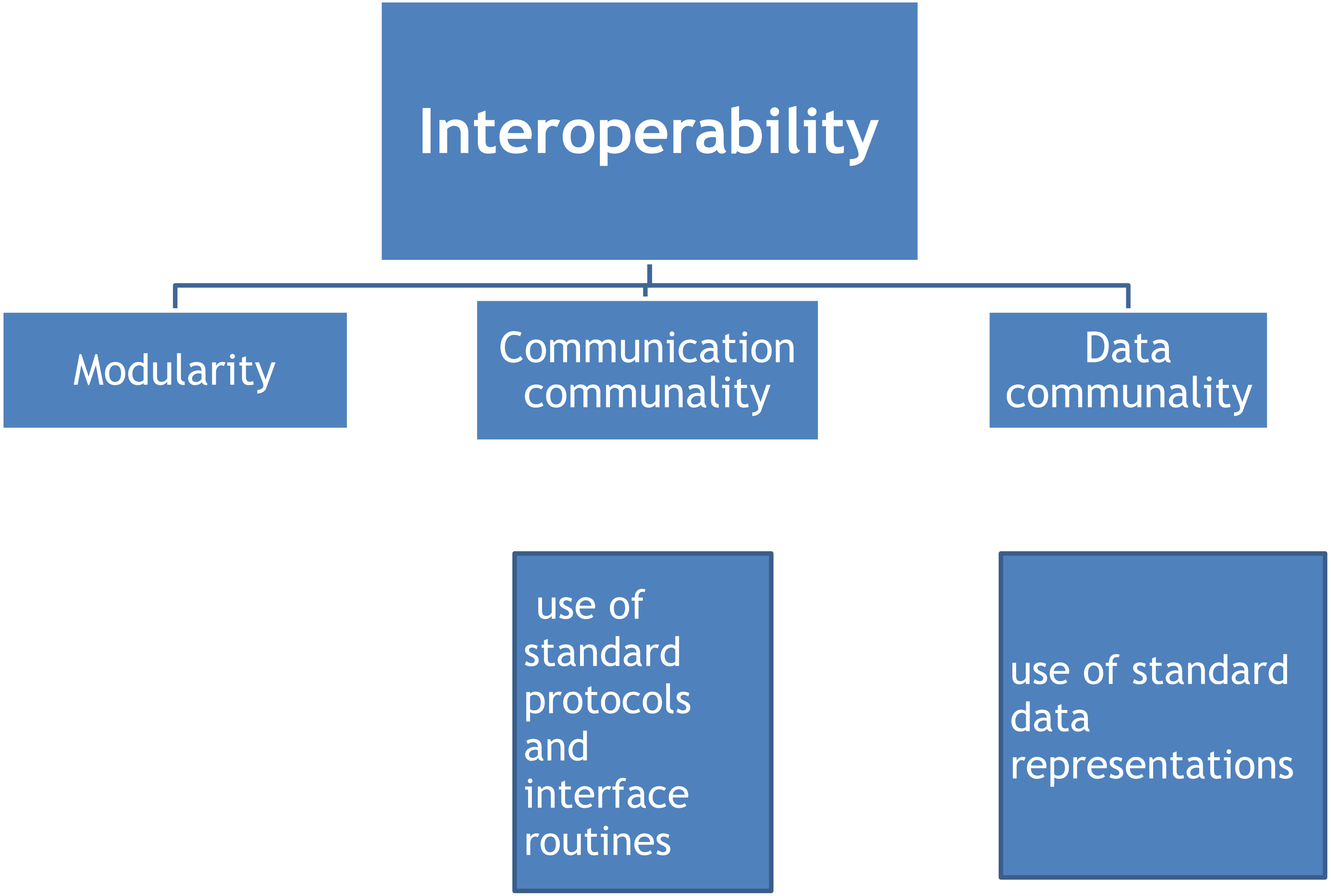
Further reading:

- https://www.researchgate.net/publication/282867676_Reusability_Metrics_of_Software_Components_Survey
- http://jeffreypoulin.info/Papers/ICSR94/icsr94.pdf

Interoperability

Interoperability

- Definition: *Effort required to couple one system with another*
- Impact:
  - Measured: design
  - Realized:
    - operation
    - transition



Interoperability

Modularity

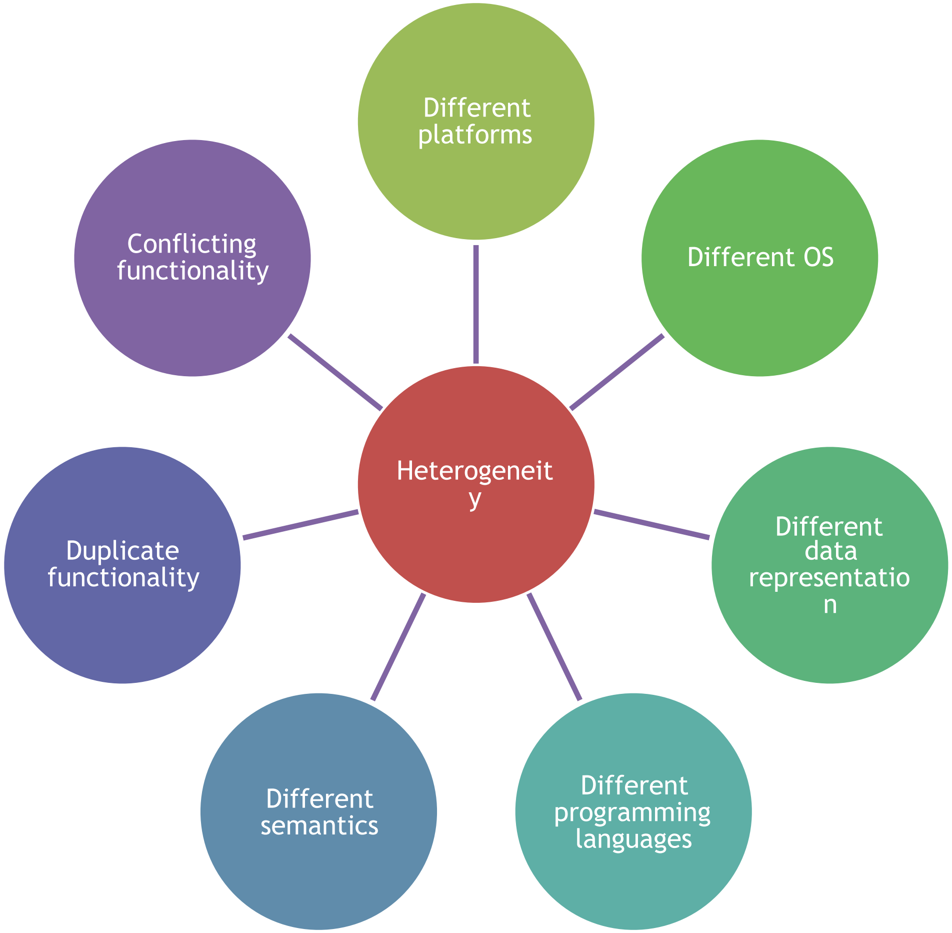
Communication communality

Data communality

use of standard protocols and interface routines

use of standard data representations

Interoperability issues



Interoperability issues

- Interoperability => interaction:
  - Communicate correct
  - Exchange data and services
- Difficulty: heterogeneity

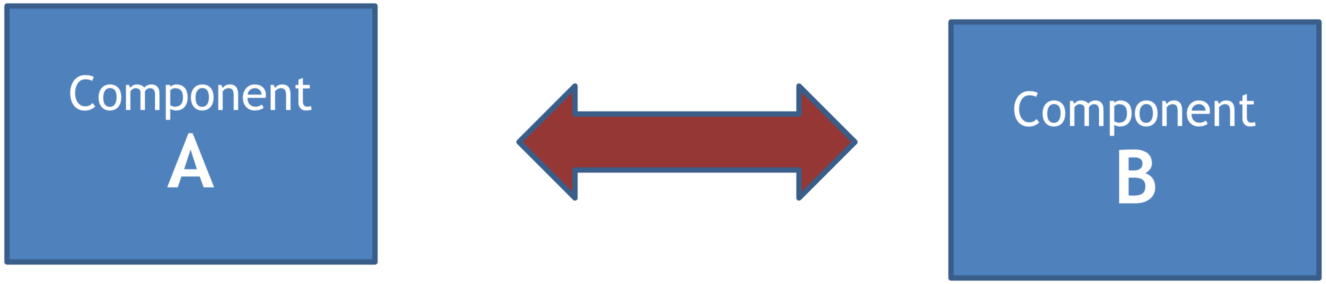
Interoperability

Syntactic

- guarantees only that data will pass through a connector properly
- Specified data formats, communication protocols
- Ex. XML, SQL standards

Semantic

- achieved only when components agree on the *meaning* of the data they exchange
- automatically interpret the information exchanged meaningfully and accurately



- Middleware:
  - CORBA
  - COM
  - RMI
  - Microsoft Message Queue

Interoperability

- Data interoperabiliy:
  - XML
  - JSON
- Web interop:
  - SOAP
  - Web services
  - microservices

Enterprise interoperability frameworks

- 2003: IDEAS: Interoperability Developments for Enterprise Application and Software.
- 2004: EIF:The European Interoperability Framework
- 2004: e-GIF: e-Government Interoperability Framework
- 2006: FEI:The Framework for Enterprise Interoperability
- 2006: C4IF: Connection, Communication, Consolidation, Collaboration Interoperability Framework
- 2007: AIF: Athena Interoperability Framework
- 2007: Enterprise Architecture Framework for Agile and Interoperable Virtual Enterprises

Interoperability metrics

1. Carnegie Melon:

[Carnegie Melon University – Measuring System Interoperability, version 1.0]

- Technical compliance measures
- Systems interoperability measures
- Operational interoperability measures
- Organizational and cultural measures

Scoreboard:
System per system - score

Information flow - systems- score

Scoreboard:
System – score (pass/margin/fail)

Interoperability metrics

2. LISI (Levels of Information Systems Interoperability) [MITRE, C4ISR]
   - ➢ Level 0: isolated – no connection (manual gateway)
   - ➢ Level 1: connected – electronic connection; separate data & application (email, FM voice, tactical data links, text files)
   - ➢ Level 2: functional – minimal common functions; separate data & application (annotated images, maps)
   - ➢ Level 3: domain – shared data & separate applications (common operational picture)
   - ➢ Level 4: enterprise – cross-domain information, interactive manipulation, shared data & applications (event-triggered global database update)

LCIM - Levels of Conceptual Interoperability Model

- https://www.researchgate.net/publication/224178883_Architecture_constraints_for_Interoperability_and_composability_in_a_smart_grid/figures?lo=1

Interoperability metrics

3. **ATL Use Case - Software Quality Control Tools Interoperability (Bugzilla, Mantis, Excel)**

[https://www.eclipse.org/atl/usecases/SoftwareQualityControlToolsInteroperability/]