



Future Lab Interoperability

ProLearn Summer School 2006, Bled, Slovenia



▪ **Fridolin Wild**
Vienna University of Economics
and Business Administration

Structure

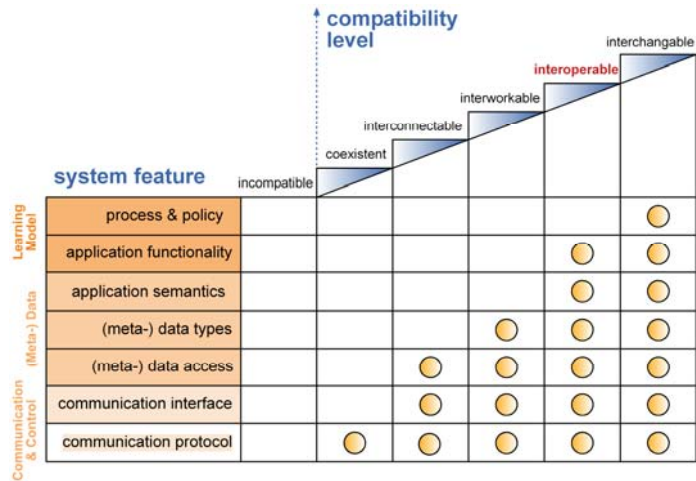
- Concept of Interoperability
- Methodology: Guiding Principles
- Existing Frameworks
- Theoretical Approaches
 - Information Integration & Dissemination
 - Remoting: Service Orientation
 - Presentation
- Concluding Remarks

DEFINING INTEROPERABILITY

Interoperability

is a **property** that emerges, when
distinctive information systems (subsystems)
cooperatively **exchange data**
in such a way that
they **facilitate** the
successful accomplishment
of an overarching **task**.

Concept of Interoperability



(modified from Kosanke, 2005; IEC, 2005)

<5>

METHODOLOGY



<6>

Methodology: Guiding Principles



- Layering
- Viewpoints
- Quality Attributes
- Patterns

<7>

GP 1: Layering



- A layer: dissect hierarchical system into a stack of dependant layers (specifying services and service interfaces)
- Two types relevant:
 - **Responsibility-based layering** (functional)
 - **Reuse-based layering** (system characteristics, organisations)

<8>

GP 2: Viewpoints

- Build set of partial models for single participants: isolated perspectives
- Hide complexity, explore hidden relationships
- Various Types:
 - Domains (Physical, Organisational, ...),
Classes of Stakeholders (Analyst, Designer, ...),
Roles of Users (Learner, Tutor, ...), ...
 - Informal, semi-formal, formal
- Many views make a viewpoint.

<9>

GP 3: Quality Attributes

- ... are non-functional requirements
- ... can be conflicting among each other (therefore: NFR framework!)
- System-specific Attributes:
 - Run-time: Performance, security, availability, usability
 - Non run-time: modifiability, portability, reusability, integrateability, testability
- Domain-specific Attributes
 - Take from learning theory, collaboration research (esp. CSCW & CMCW)
 - e.g. awareness, adaptability, seamlessness, collaborative filtering, dynamism

<10>

GP 4: Patterns

= recurring problem + solution + forces + ...

- Pattern language:
aligned patterns serving an overarching goal
- Relevant pattern traditions:
 - Pedagogical patterns (e.g. 'invisible teacher')
 - Patterns of Cooperative Interaction
(e.g. 'overlapping responsibilities')
 - Architectural Patterns
 - Integration Patterns
(e.g. remoting patterns like 'broker')
- Current shortcoming: learner-centred patterns

<11>

EXISTING FRAMEWORKS

<12>

Existing Frameworks (I)

	Layering	Viewpoints	Roles	Quality Attributes	Patterns
ELF	functional / opaque	Single (Developer, semi-formal)	Application Services	Reusability, affordability, interoperability, multi-linguality, multi-tenancy	
CORDRA	re-usability / opaque			Scalability, availability, security, enterprise readiness	
POOL / ECL	functional			Flexibility, adaptability, portability, interoperability, flexibility	
OKI OSID	functional			Availability, security, enterprise readiness	
IBM Learning Infrastructure / IBM SIS	functional / opaque	Single (Developer, semi-formal)	Application Services	Reusability, affordability, interoperability, multi-linguality, multi-tenancy	
LearnServe	functional / opaque	Single (Developer, semi-formal)	Presenting Services	Scalability, availability, security, enterprise readiness	
Sun e-Learning Framework	functional		Application Services	Reusability, affordability, interoperability, multi-linguality, multi-tenancy	
IMS TI	functional / indirect			Investment protection, cost-effectiveness, enterprise readiness	
WS Framework for Dynamic E-Learning Systems	functional			Security, portability, interoperability, flexibility	
LORI	functional / indirect / opaque	Single (Developer, semi-formal)	Application Services	Reusability, affordability, interoperability, multi-linguality, multi-tenancy	

Existing Frameworks (II)



- **GP 1: Layering**
 - Widely adopted
 - Mainly functional layering
 - CORDRA: community based re-usability
 - LORI/IMS-TI: integration of legacy systems
- **GP 2: Viewpoints**
 - Small number of viewpoints (usually: developers)
 - IBM LI/SIS: + organisation
 - IMS-TI: + teacher + learner

Existing Frameworks (III)

- **GP 2: Roles**
 - Mainly: application services or information integration
 - IMS-TI / LearnServe: presentation integration
- **GP 3: Quality Attributes**
 - Only generic attributes,
 - Domain-specific attributes are missing
- **GP 4: Patterns**
 - No considerations in the frameworks
 - No patterns for distributed systems

<15>

THEORETICAL APPROACHES

<16>

Theoretical Approaches



- Three big streams:
 - Information Integration and Dissemination
 - Remoting: Service Orientation
 - Presentation Integration

<17>

Information Integration & Dissemination



- Basic Scenarios
 - Mapping-based approaches
 - Intermediary-based approaches
 - Query-based approaches
- Scope of Integration
 - Persistent Integration (through (partial) replication)
 - Virtual Integration (on demand)
- Composition
 - Data Integration (heterogeneous)
 - Data Exchange (idempotent systems)

<18>



I.I. & D.

- Information Retrieval / Querying
 - (Ad hoc) queries are matched against data
 - e.g. query – results
- Information Filtering
 - Data streams matched against standing queries
 - e.g. publish-subscribe
 - Stateful or stateless

<19>



Remoting: Service Orientation

- Idea: dissolve functionality from application, provide standard interface
- e.g. web-services: are “a software system designed to support interoperable machine-to-machine interaction over a network” (w3c, 2004)
- Choreography and Orchestration: (BPEL, BPEL4people, SLeD)

<20>



Presentation Integration

- Portlets:
 - Encapsulated applications
 - Assembled into pages by a container
 - Return renderable mark-up (e.g. XHTML)
- ... can be made interoperable via WSRP
- Java Servlets
- JSP (extend Servlets)

<21>



CONCLUDING REMARKS

<22>



Concluding Remarks

- Guide the design process:
 - Quality attributes
 - (from social-constructivist learning theory)
 - (through usage and interaction analysis)
 - (from documented pedagogical patterns)
 - Patterns
- Draw upon prior experiences
 - Viewpoints: learners, facilitators, organisations, developers
 - Layers: have to be specified
- Integration Architecture: pursue information querying and information filtering, add bridging elements

<23>



#eof.

beware, the end is near.

<24>