XML for New Generation OSS

David Lewis

UH Communications A/S

Overview

- Challenges of NGOSS
- Relevant Feature of XML
- Application of XML in NGOSS
  - Low cost, flexible user interaction
  - Managing technology specific contract specs
  - Integration with legacy OSS
  - Smooth evolution of contract designs

NGOSS Vision

- Management Systems assembled from **COTS Components**
- Components interact through well-defined **Contracts**
- Components are wholly in one of **3 computing tiers**:
  - Human Interaction Tier
  - Process Automation Tier
  - Enterprise Information Tier
- Generic Requirements for components, contracts and supporting DPE in ACT's GB909
  - Being refined into a useable framework by the NG OSS teams

Relevant NGOSS Requirements

- Contract specifications implementable in different technologies,
  - e.g. CORBA, CMIP, SNMP, HTTP
- Interoperation of legacy OSS with NGOSS
- Smooth deployment of new contract versions

eXtensible Mark-up Language (XML)

- Web-ready derivative of SGML publishing markup
- Applications require an **agreed schema**, user defined tags and structuring rules
- Communities actively standardizing application specific tags and information structures
  - Emerging as the lingua franca of e-commerce
- Management Standards emerging:
  - e.g. DMTF's WBEM, T1M1's tML

XML Features of Interest

- Proliferation of cheap tools for XML document validation, manipulation
  - standard APIs e.g. SAX and DOM
- Breadth of application leading to rapidly growing skills base
- Separates information structure and presentation
  - Document Type Definitions/Schema define XML document structure, content type and tags
  - XML Stylesheet Language Transformations (XSLT) provides an open way of transforming between XML Schemas

XML for NGOSS: Overview

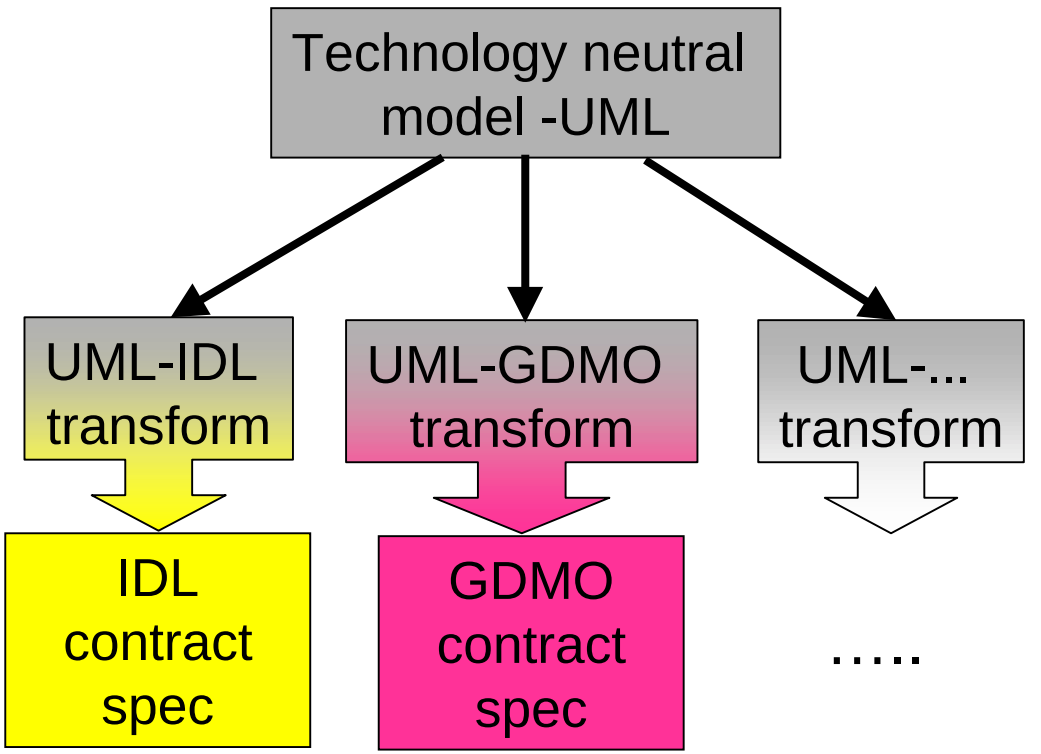
- XML does **not** provide a DPE solution
- XML already used for Component Deployment Descriptors
  - EJB and CORBA Component Model descriptors
- XML provides **flexible, low cost Infrastructure for HIT**
  - Presentation Style Sheets
  - HTTP Transport
- XSL for production of **Technology Specific Contract Specs**
- XSL supports **Interoperability across Technologies**
- XML supports **Release Independence**

XML for Human Interaction Tier

- XML Stylesheets allow the same management information to be presented and manipulated by users in different ways
- XML/HTTP protocols allow WWW infrastructure to be exploited:
  - Several protocols exist - SOAP, XML-RPC etc
  - Can use existing browsers and servers
  - Can use existing HTTP security mechanisms
- XML/HTTP protocol deficiencies:
  - Verbose encoding
  - Lack of common distributed services - location, transaction etc
  - Difficulties with notifications from HTTP server

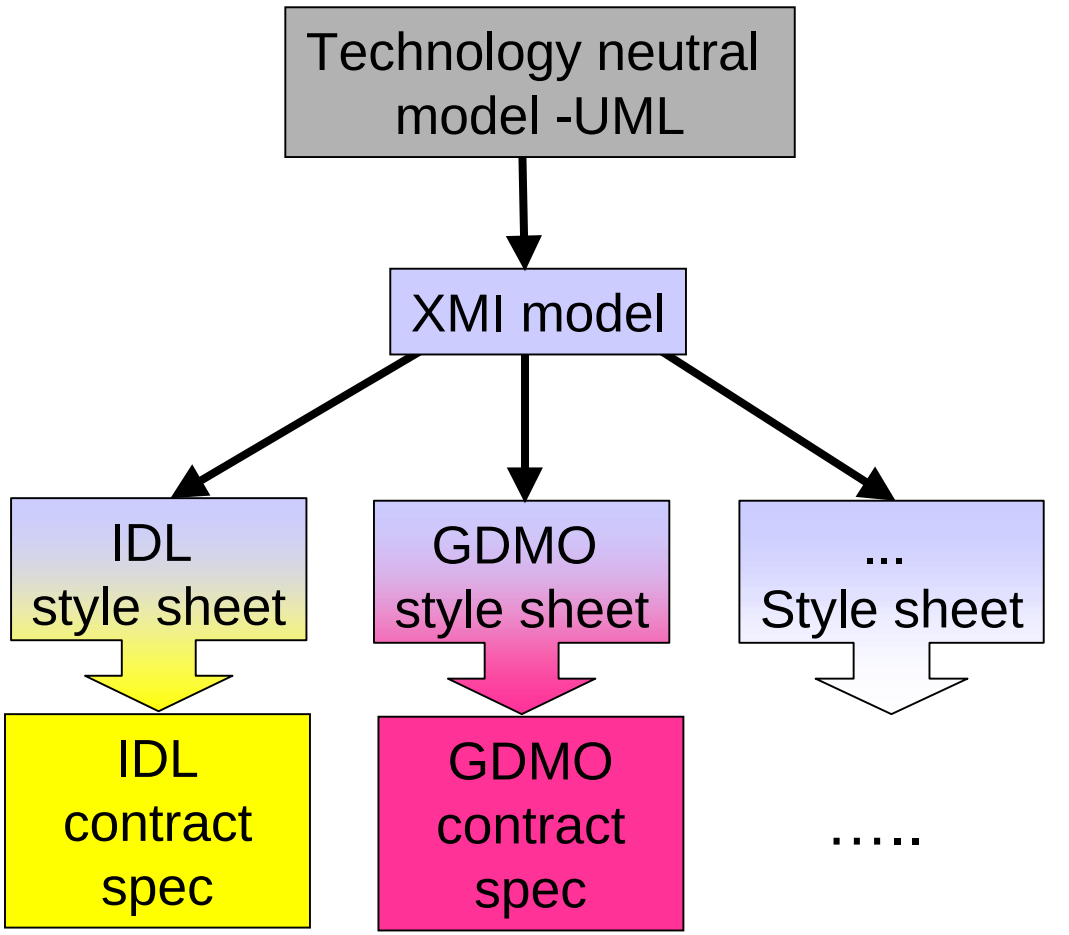
Technology Specialization from UML

- Technology Neutral UML to be used for NGOSS Contracts and Data Models
- Needs transforming to Technology Specific Models, e.g.GDMO, IDL, SMI
  - a bit of a black art - optimum transforms are difficult
  - implemented as custom transformation tool



XML for Technology Specialization

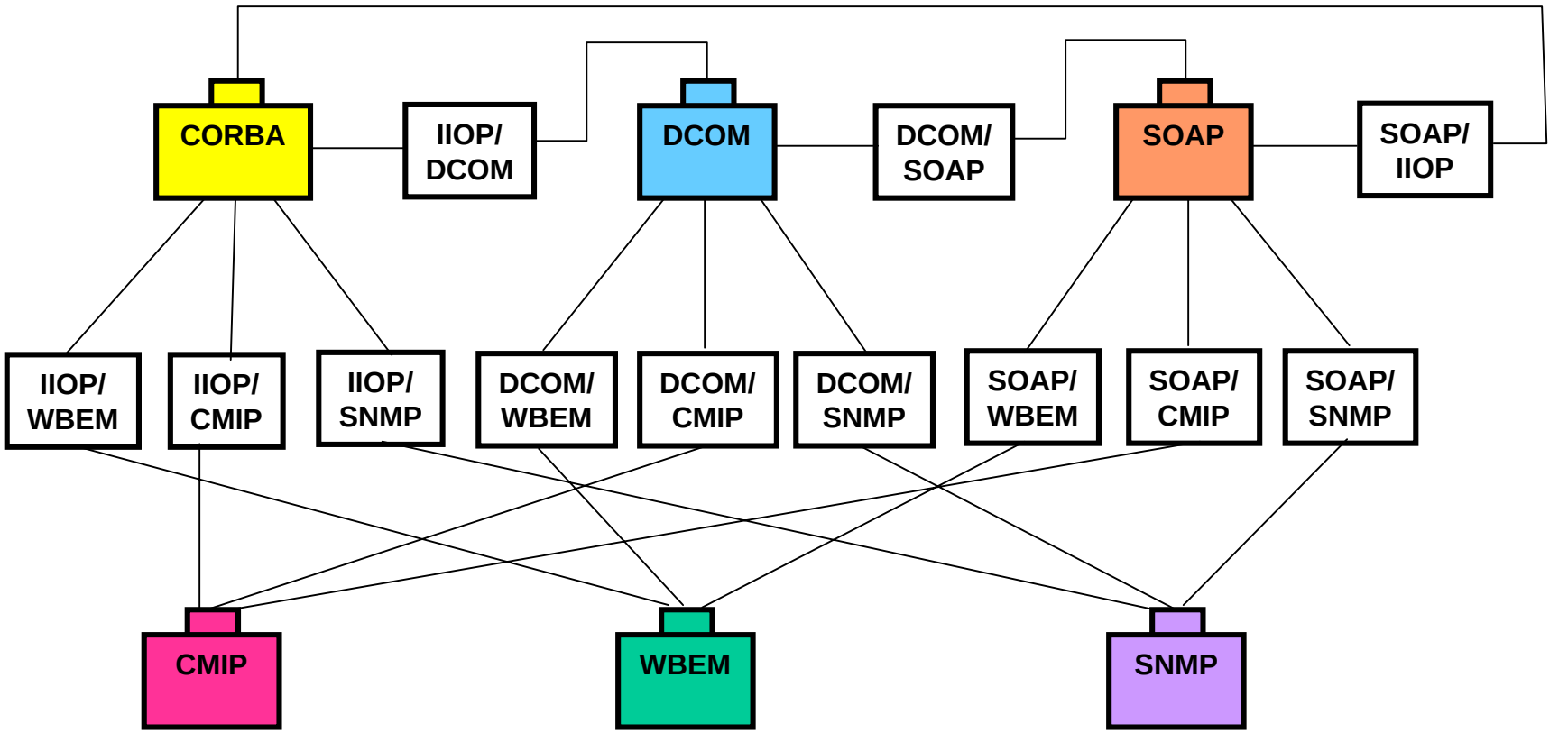
- Working from XML version of UML models **opens up** specialization process
  - Use of XSL for defining transforms
  - Use of XSL transform tools
  - Can start with XMI
  - Can support multiple transforms for a specific technology



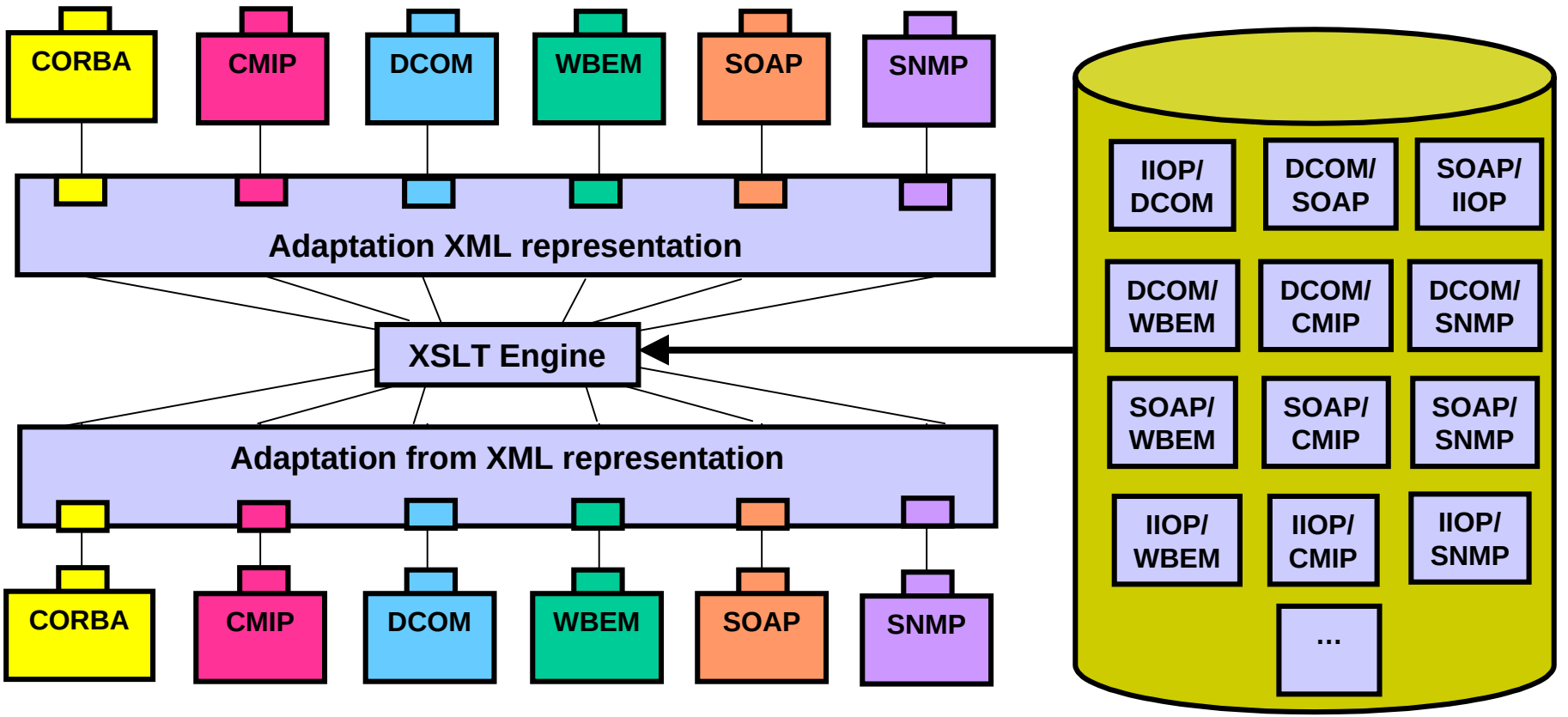
Interoperability across Technologies

- NGOSS implementations must interoperate with **legacy OSS**
- NGOSS should support **different technology bindings** for contract implementations, e.g. IIOP, DCOM, SOAP, CMIP, SNMP, WBEM
- **Adapters** therefore needed to interoperate between components with different contract technologies
- Potentially **n^2 adapter types** needed, plus optimizing variants

Individual Technology Adapters



XML-Based Adaptation



XML Based Technology Interoperability

- Define a **single XML profile** for each communication technology
- Adapters can be developed using **XSLT docs and generic transform engines:**
  - Can readily **prototype new transforms -** may need custom implementations for speed/scalability
  - Can develop **dynamically loadable** adapters in NGOSS DPE

Interoperability across Contract Versions

- NGOSS must support contract release independence
  - deployment of new contract versions must not impact other interacting components
- XML based messages or RPC support this
  - New contract versions specified using new tags
  - XML parsers ignore unknown tags
  - Components can avoid supporting multiple contract versions simultaneously

Conclusions

- XML can compliment component technologies to support key NGOSS requirements
  - XML transport over HTTP provides flexible low cost HIT component solutions
  - XSL provides an open transform technique to support technological heterogeneity in NGOSS
  - XML based contract implementations will ease release independence

Further Information

- NGOSS Architecture: **Technology Specific Group**
- EU sponsored R&D: **FORM Project**
  - http://www.uhc.dk/form/index.htm
- **tML**: framework for XML schemas for management:
- **DMTF**: XML for management
  - http://www.dmtf.org