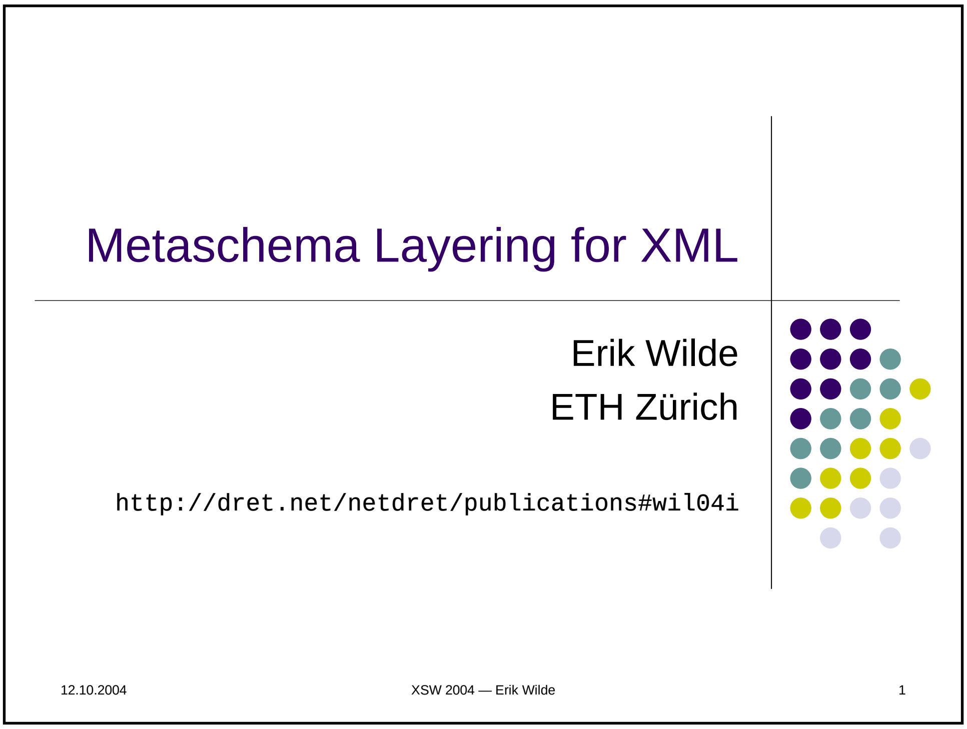
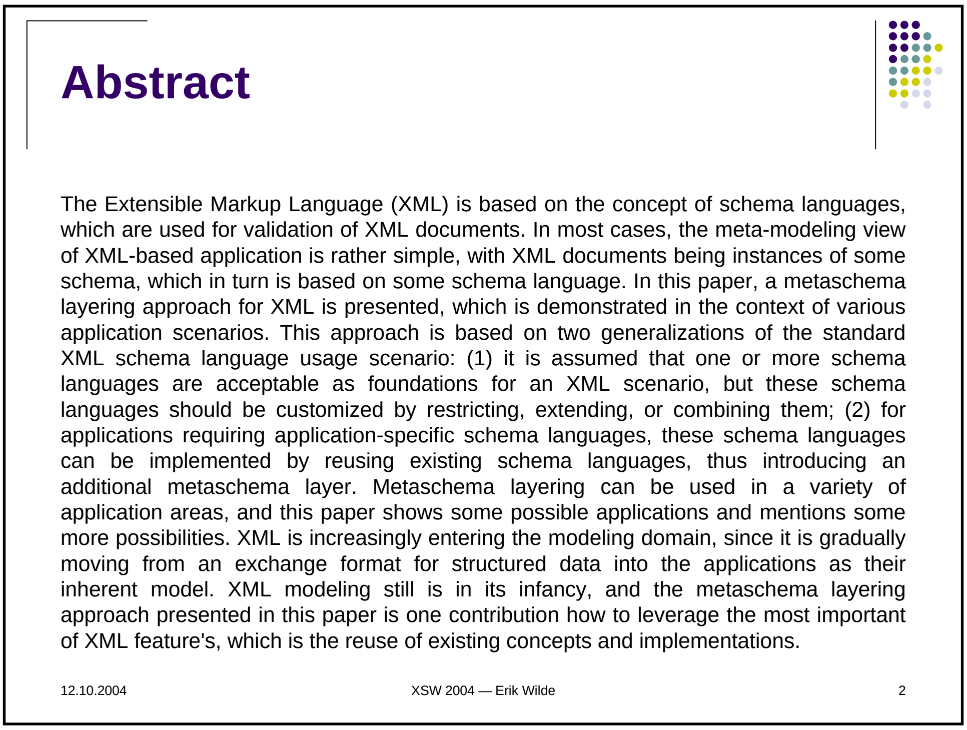


Metaschema Layering for XML

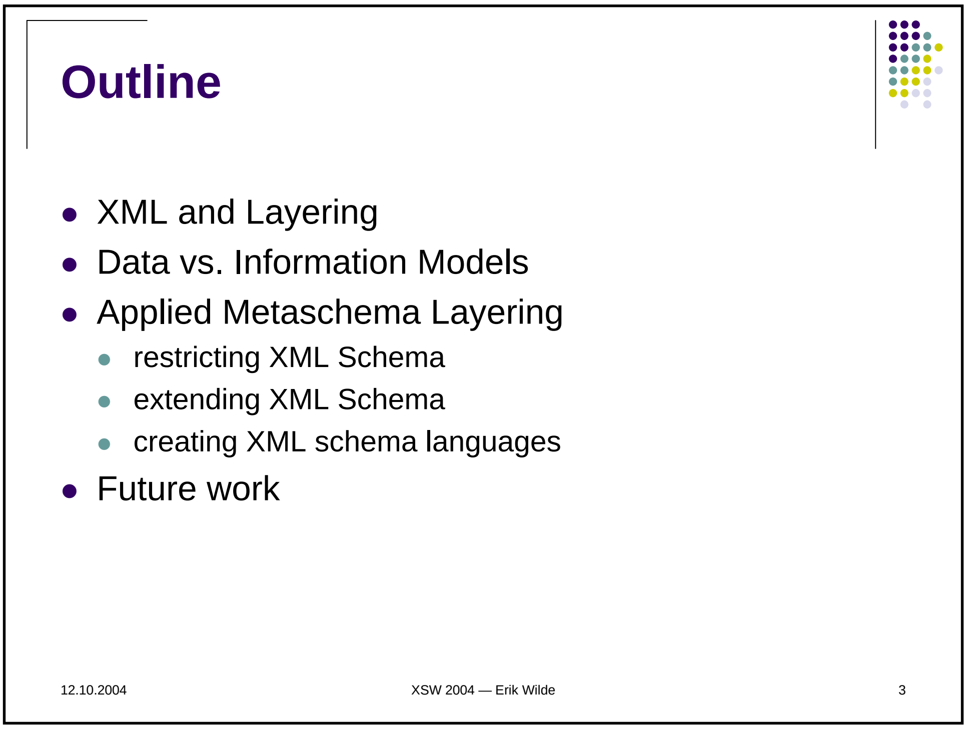
Erik Wilde
ETH Zürich

`http://dret.net/netdret/publications#wil04i`



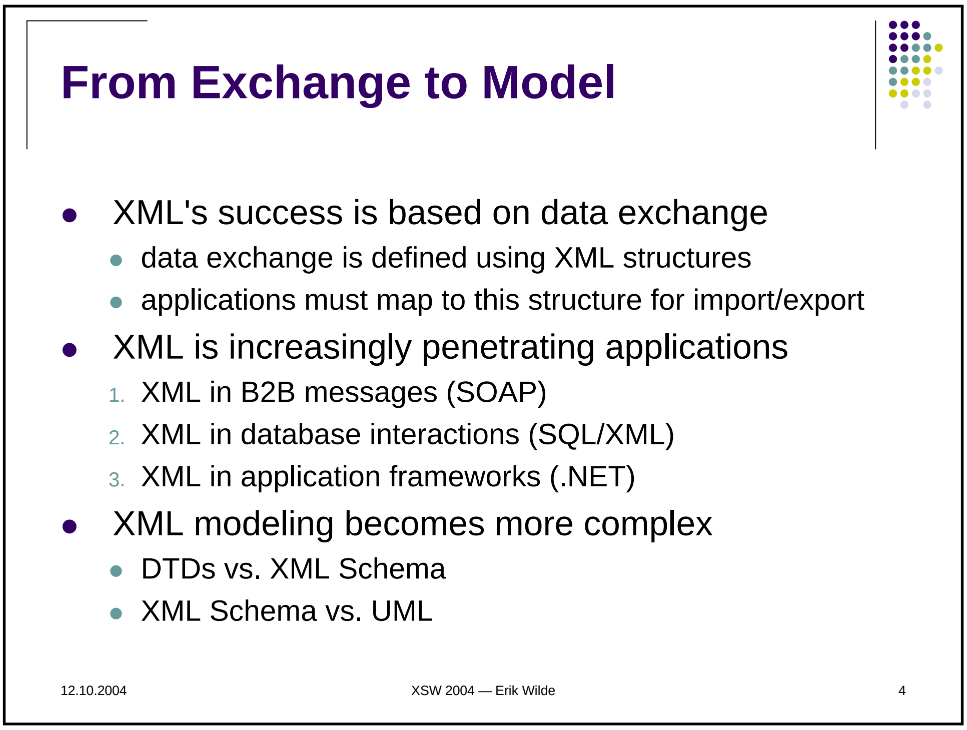
Abstract

The Extensible Markup Language (XML) is based on the concept of schema languages, which are used for validation of XML documents. In most cases, the meta-modeling view of XML-based application is rather simple, with XML documents being instances of some schema, which in turn is based on some schema language. In this paper, a metaschema layering approach for XML is presented, which is demonstrated in the context of various application scenarios. This approach is based on two generalizations of the standard XML schema language usage scenario: (1) it is assumed that one or more schema languages are acceptable as foundations for an XML scenario, but these schema languages should be customized by restricting, extending, or combining them; (2) for applications requiring application-specific schema languages, these schema languages can be implemented by reusing existing schema languages, thus introducing an additional metaschema layer. Metaschema layering can be used in a variety of application areas, and this paper shows some possible applications and mentions some more possibilities. XML is increasingly entering the modeling domain, since it is gradually moving from an exchange format for structured data into the applications as their inherent model. XML modeling still is in its infancy, and the metaschema layering approach presented in this paper is one contribution how to leverage the most important of XML feature's, which is the reuse of existing concepts and implementations.



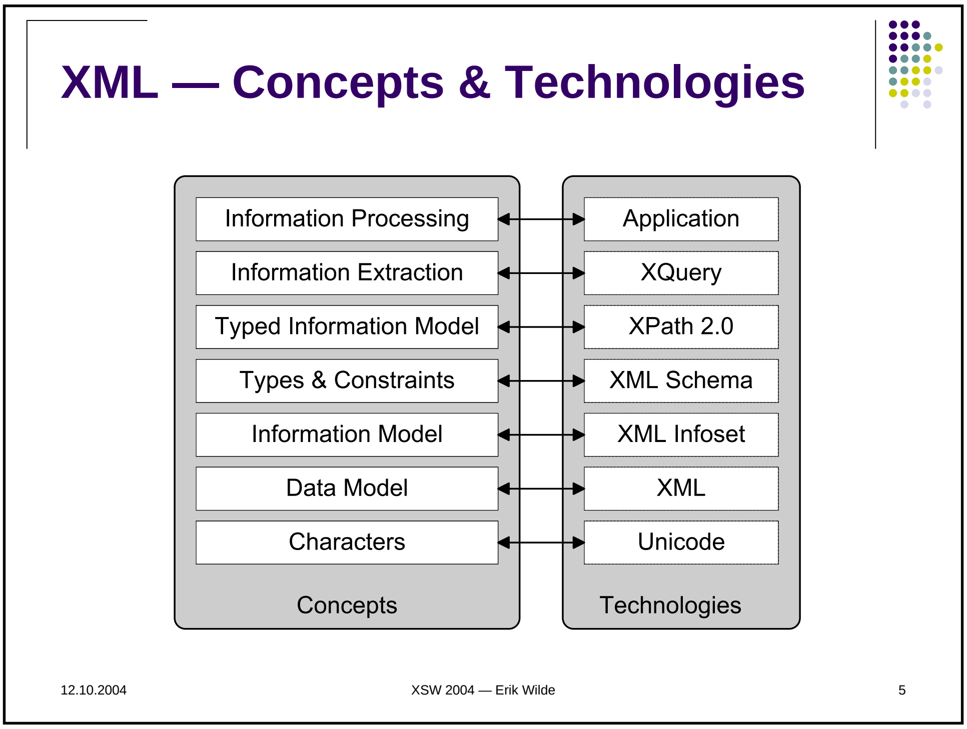
Outline

- XML and Layering
- Data vs. Information Models
- Applied Metaschema Layering
  - restricting XML Schema
  - extending XML Schema
  - creating XML schema languages
- Future work



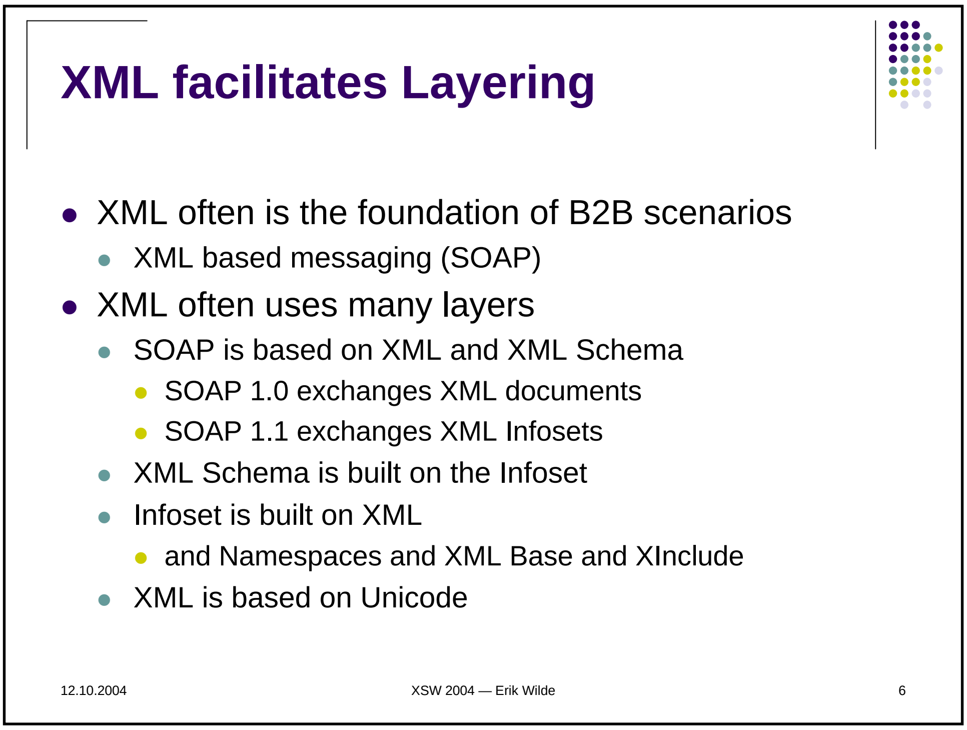
From Exchange to Model

- XML's success is based on data exchange
  - data exchange is defined using XML structures
  - applications must map to this structure for import/export
- XML is increasingly penetrating applications
  1. XML in B2B messages (SOAP)
  2. XML in database interactions (SQL/XML)
  3. XML in application frameworks (.NET)
- XML modeling becomes more complex
  - DTDs vs. XML Schema
  - XML Schema vs. UML



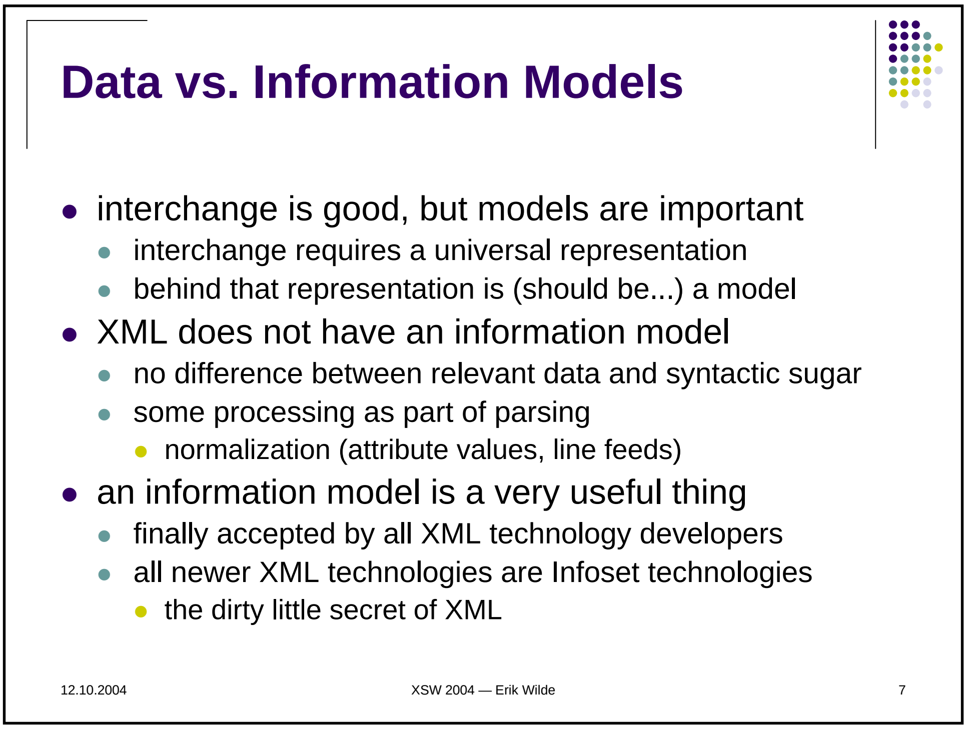
XML — Concepts & Technologies

Concepts	Technologies
Information Processing	Application
Information Extraction	XQuery
Typed Information Model	XPath 2.0
Types & Constraints	XML Schema
Information Model	XML Infoset
Data Model	XML
Characters	Unicode



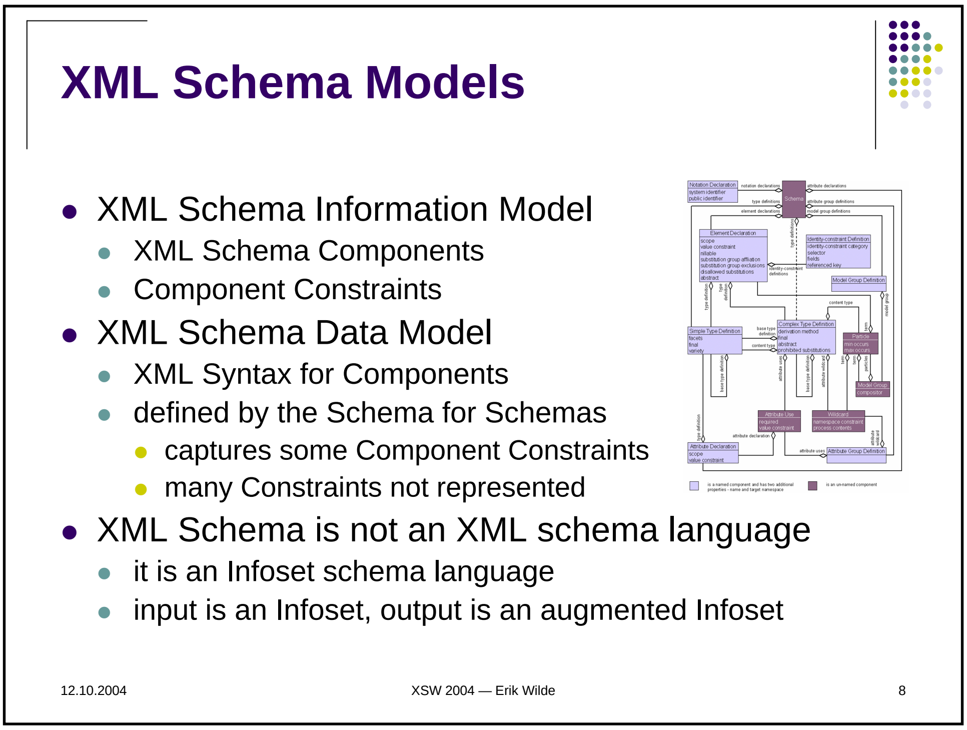
XML facilitates Layering

- XML often is the foundation of B2B scenarios
  - XML based messaging (SOAP)
- XML often uses many layers
  - SOAP is based on XML and XML Schema
    - SOAP 1.0 exchanges XML documents
    - SOAP 1.1 exchanges XML Infosets
  - XML Schema is built on the Infoset
  - Infoset is built on XML
    - and Namespaces and XML Base and XInclude
  - XML is based on Unicode



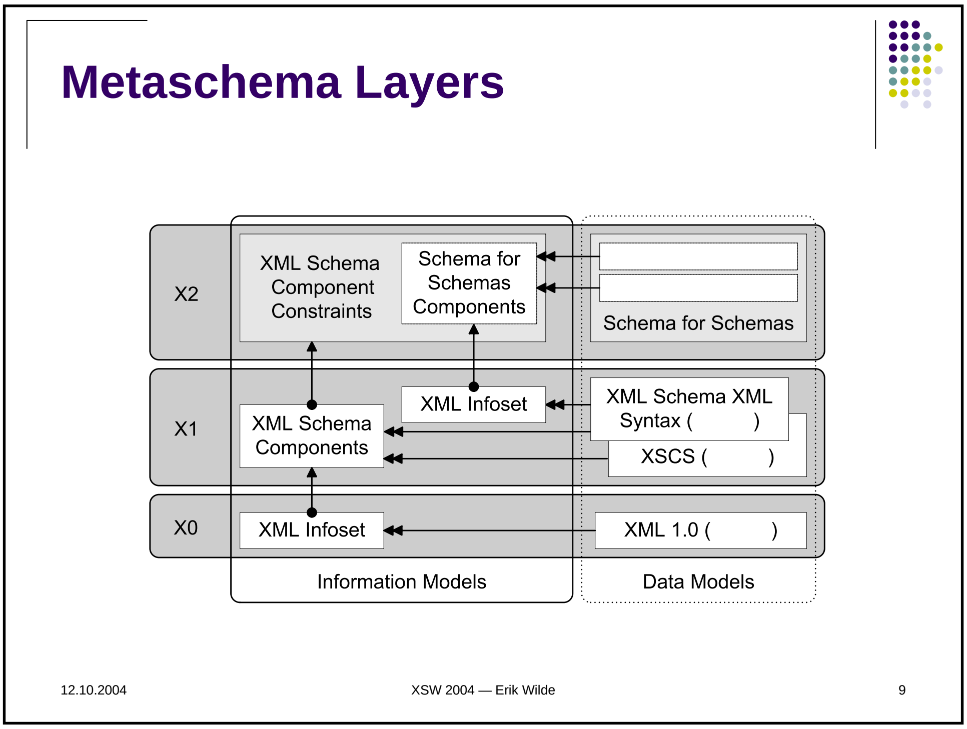
Data vs. Information Models

- interchange is good, but models are important
  - interchange requires a universal representation
  - behind that representation is (should be...) a model
- XML does not have an information model
  - no difference between relevant data and syntactic sugar
  - some processing as part of parsing
    - normalization (attribute values, line feeds)
- an information model is a very useful thing
  - finally accepted by all XML technology developers
  - all newer XML technologies are Infoset technologies
    - the dirty little secret of XML



XML Schema Models

- XML Schema Information Model
  - XML Schema Components
  - Component Constraints
- XML Schema Data Model
  - XML Syntax for Components
  - defined by the Schema for Schemas
    - captures some Component Constraints
    - many Constraints not represented
- XML Schema is not an XML schema language
  - it is an Infoset schema language
  - input is an Infoset, output is an augmented Infoset



Metaschema Layers

X2

XML Schema Component Constraints

Schema for Schemas Components

Schema for Schemas

X1

XML Infoset

XML Schema Components

XML Schema XML Syntax ( )

XSCS ( )

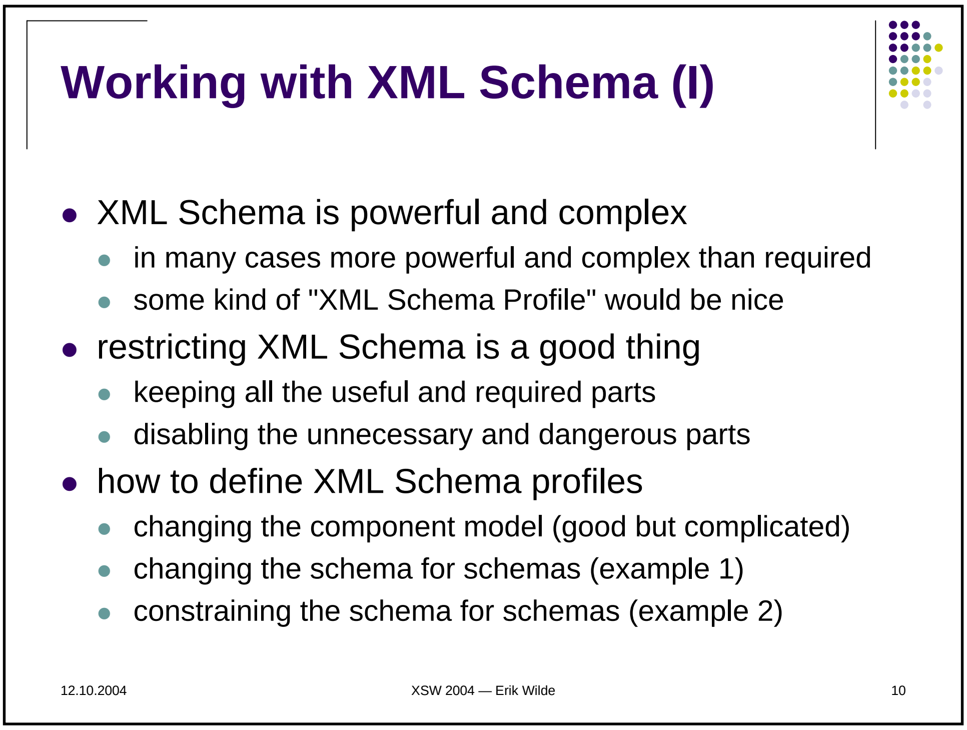
X0

XML Infoset

XML 1.0 ( )

Information Models

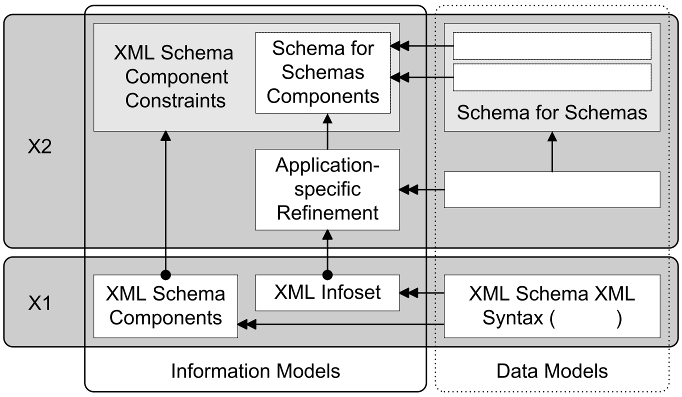
Data Models



Working with XML Schema (I)

- XML Schema is powerful and complex
  - in many cases more powerful and complex than required
  - some kind of "XML Schema Profile" would be nice
- restricting XML Schema is a good thing
  - keeping all the useful and required parts
  - disabling the unnecessary and dangerous parts
- how to define XML Schema profiles
  - changing the component model (good but complicated)
  - changing the schema for schemas (example 1)
  - constraining the schema for schemas (example 2)

Restricting XML Schema (1)



X2

XML Schema Component Constraints

Schema for Schemas Components

Schema for Schemas

Application-specific Refinement

X1

XML Schema Components

XML Infoset

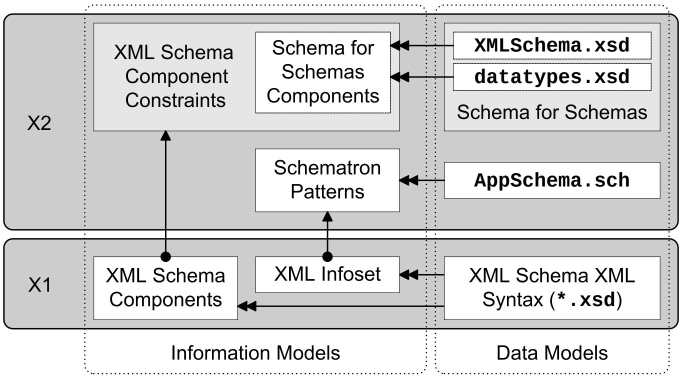
XML Schema XML Syntax ( )

Information Models

Data Models

12.10.2004 XSW 2004 — Erik Wilde 11

Restricting XML Schema (2)



X2

XML Schema Component Constraints

Schema for Schemas Components

`XMLSchema.xsd`

`datatypes.xsd`

Schema for Schemas

Schematron Patterns

`AppSchema.sch`

X1

XML Schema Components

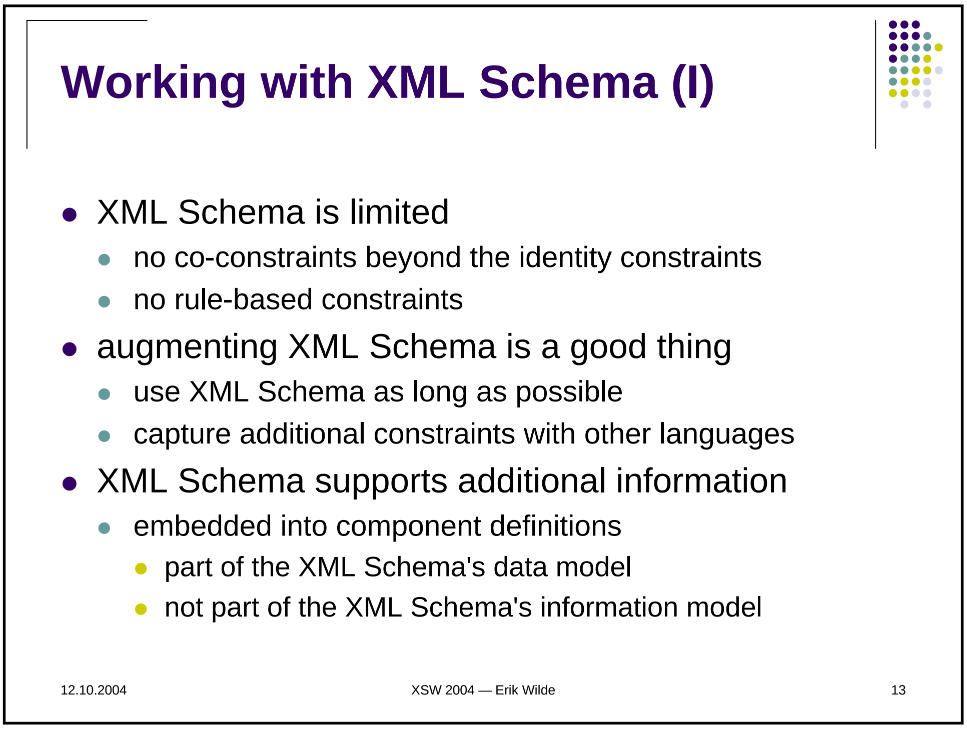
XML Infoset

XML Schema XML Syntax (`*.xsd`)

Information Models

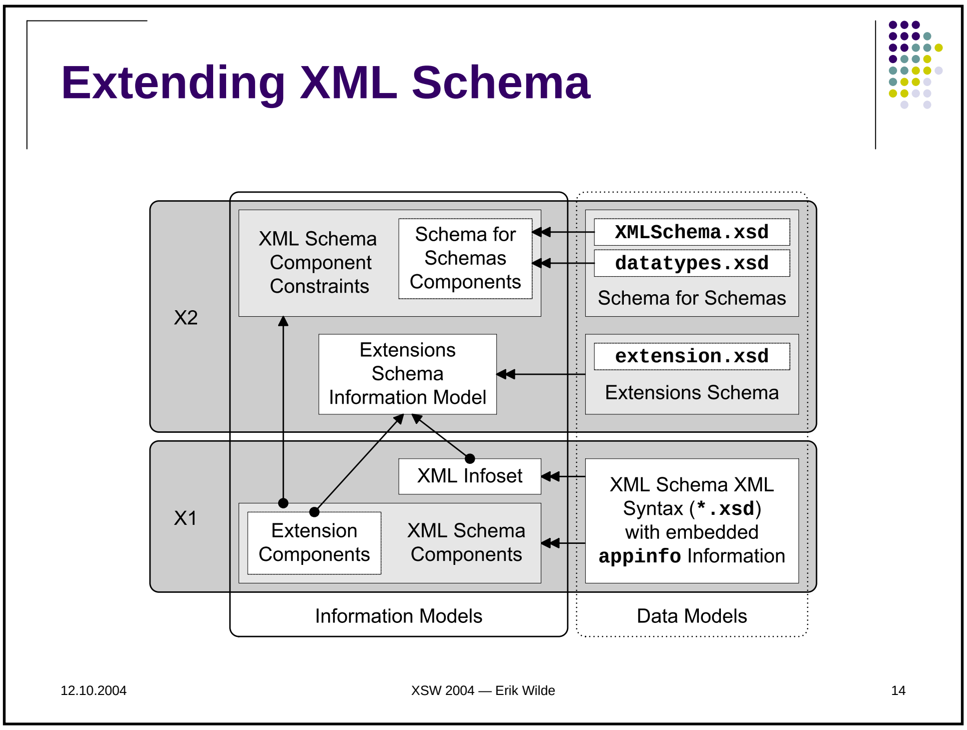
Data Models

12.10.2004 XSW 2004 — Erik Wilde 12



Working with XML Schema (I)

- XML Schema is limited
  - no co-constraints beyond the identity constraints
  - no rule-based constraints
- augmenting XML Schema is a good thing
  - use XML Schema as long as possible
  - capture additional constraints with other languages
- XML Schema supports additional information
  - embedded into component definitions
    - part of the XML Schema's data model
    - not part of the XML Schema's information model



Extending XML Schema

X2

XML Schema Component Constraints

Schema for Schemas Components

XMLSchema.xsd

datatypes.xsd

Schema for Schemas

Extensions Schema Information Model

extension.xsd

Extensions Schema

X1

XML Infoset

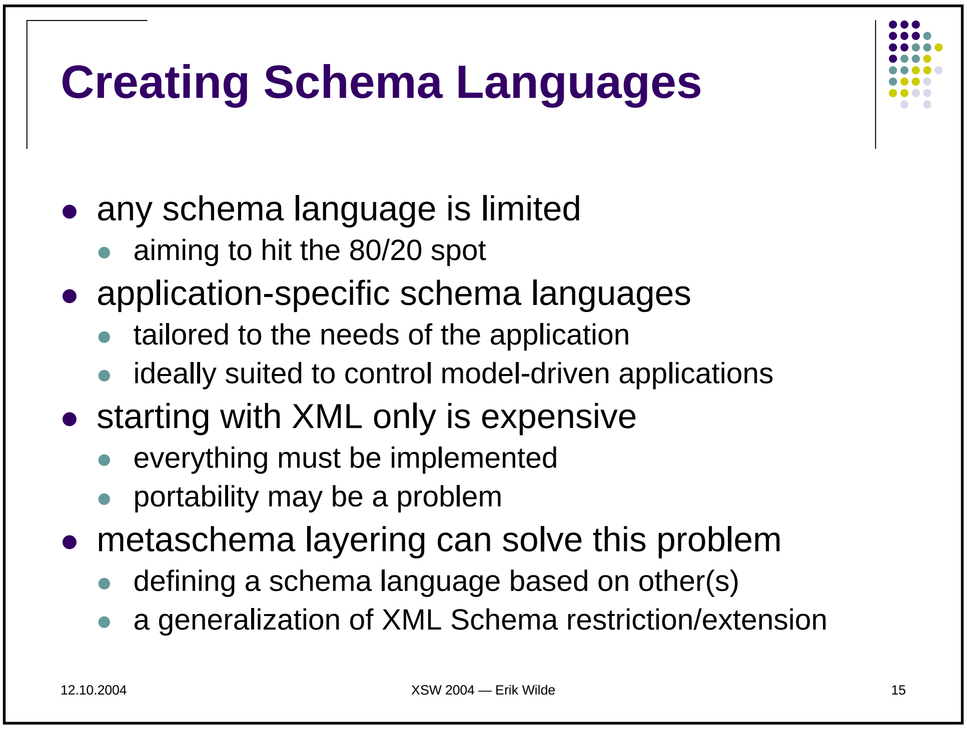
Extension Components

XML Schema Components

XML Schema XML Syntax (***.xsd**) with embedded **appinfo** Information

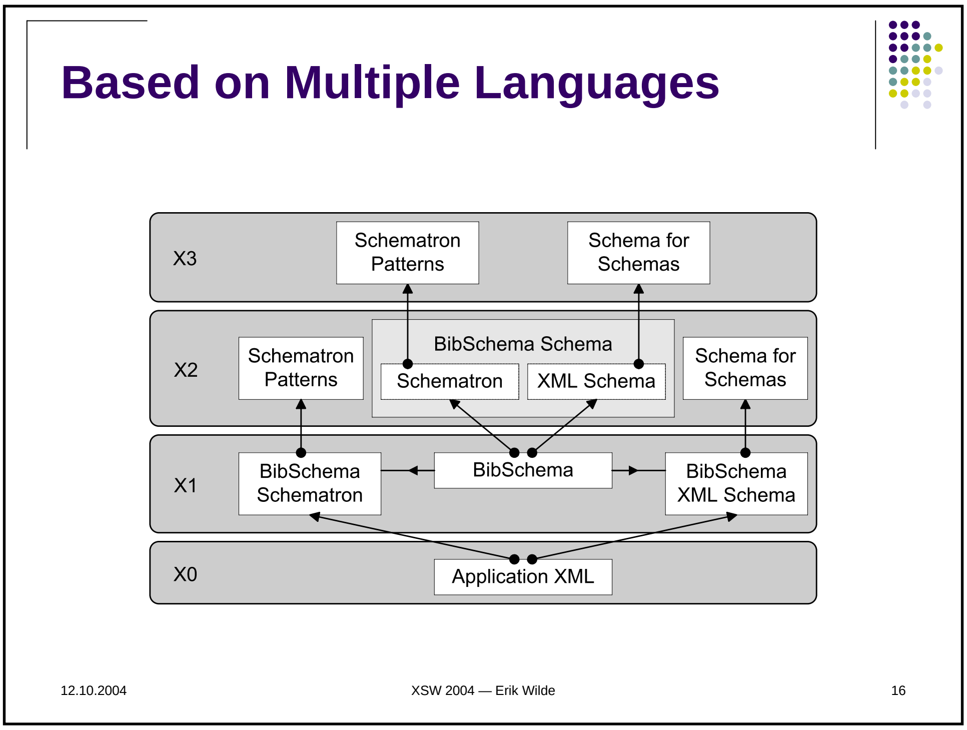
Information Models

Data Models



Creating Schema Languages

- any schema language is limited
  - aiming to hit the 80/20 spot
- application-specific schema languages
  - tailored to the needs of the application
  - ideally suited to control model-driven applications
- starting with XML only is expensive
  - everything must be implemented
  - portability may be a problem
- metaschema layering can solve this problem
  - defining a schema language based on other(s)
  - a generalization of XML Schema restriction/extension



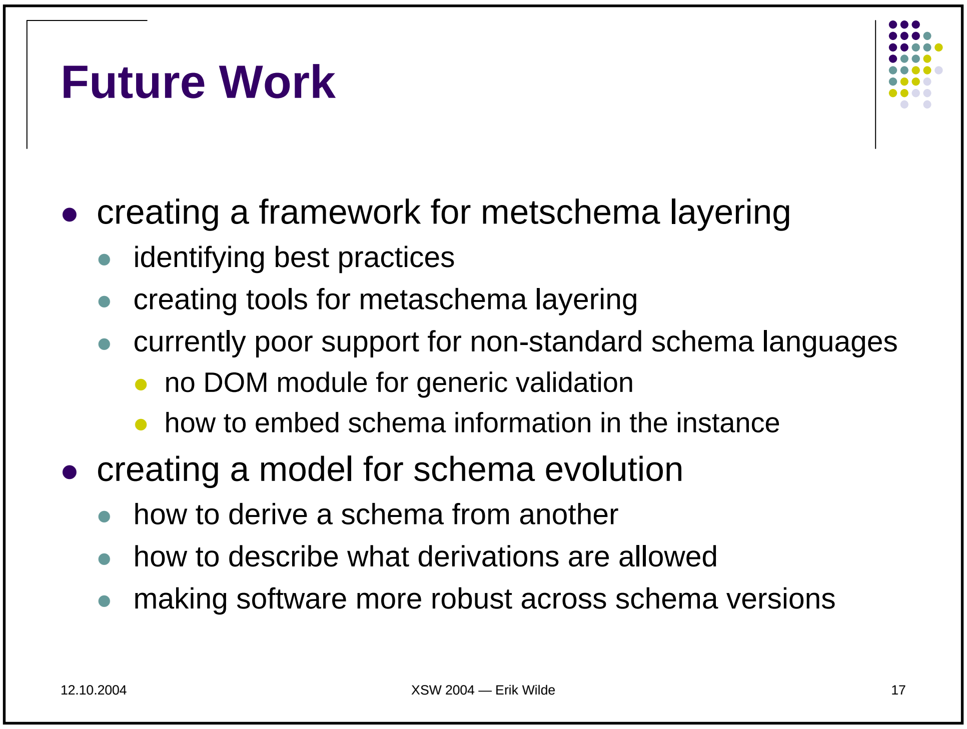
Based on Multiple Languages

X3
Schematron Patterns
Schema for Schemas

X2
Schematron Patterns
BibSchema Schema
Schematron
XML Schema
Schema for Schemas

X1
BibSchema Schematron
BibSchema
BibSchema XML Schema

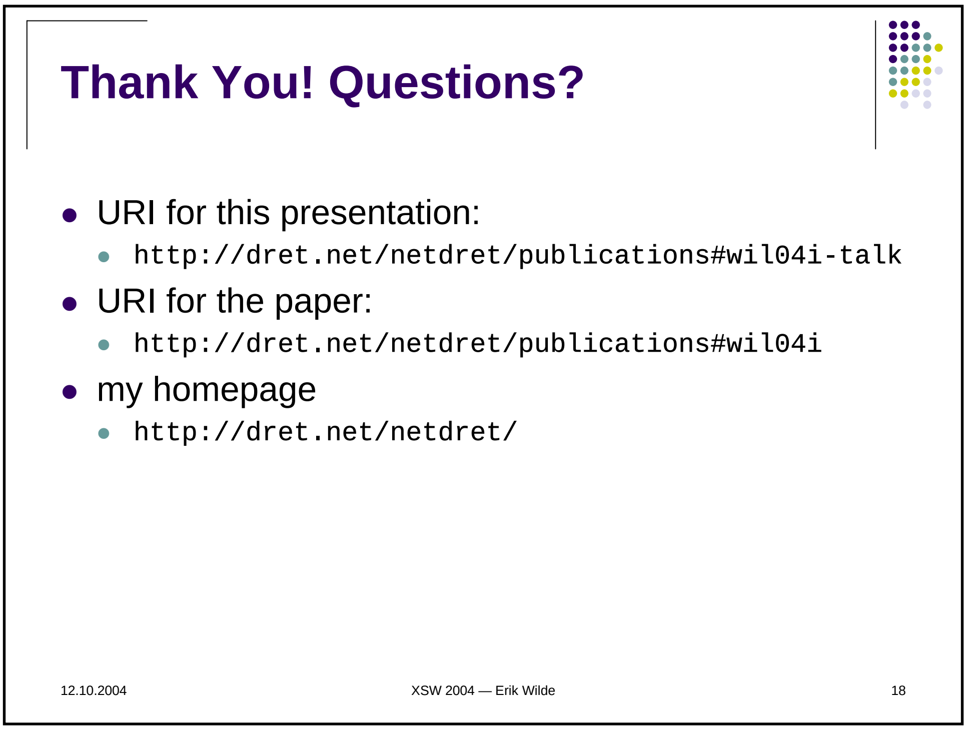
X0
Application XML



Future Work

- creating a framework for metschema layering
  - identifying best practices
  - creating tools for metaschema layering
  - currently poor support for non-standard schema languages
    - no DOM module for generic validation
    - how to embed schema information in the instance
- creating a model for schema evolution
  - how to derive a schema from another
  - how to describe what derivations are allowed
  - making software more robust across schema versions

12.10.2004 XSW 2004 — Erik Wilde 17



Thank You! Questions?

- URI for this presentation:
  - `http://dret.net/netdret/publications#wil04i-talk`
- URI for the paper:
  - `http://dret.net/netdret/publications#wil04i`
- my homepage
  - `http://dret.net/netdret/`

12.10.2004 XSW 2004 — Erik Wilde 18