



femML: Finite Element Modeling Markup Language

Presented by

J. Michopoulos

(johnM@cms.nrl.navy.mil)

NAVAL RESEARCH LABORATORY

Composite Materials and Structures Group, CODE 6304

Washington DC 20375

ASME 2001 Design Engineering Technical Conferences
& Computers and Information in Engineering Conference
Pittsburgh, Pennsylvania
11 September, 2001

Overview

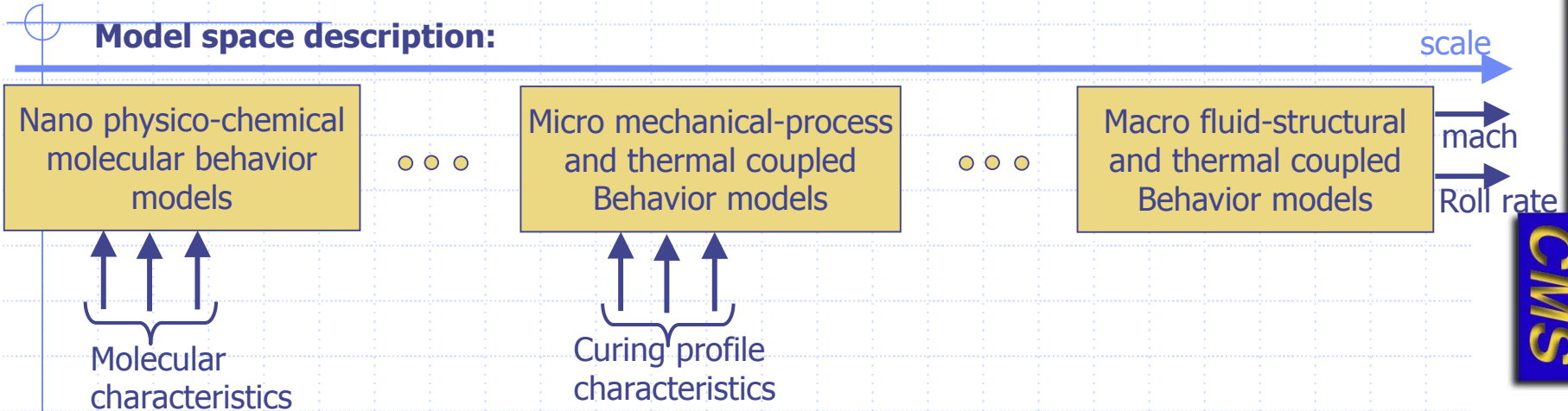
- **Vision**
- **Where we Are (CMS Space)**
- **Motivation**
- **Background**
- **Problem and Issues**
- **Usage and Definition of XML**
- **Objectives and Approach**
- **Progress**
- **Open call for collaboration**



A Vision for Computational Material/Structural Science

Be able to answer Questions like this:

What the curing profile of a composite laminate, and macromolecular characteristics of a resin should be in order to be able to sustain a given roll rate for a given time in a Mach 3 mission?



Approach space description: Design Optimization

Design Variables: Molecular Characteristics, Curing Profile Characteristics

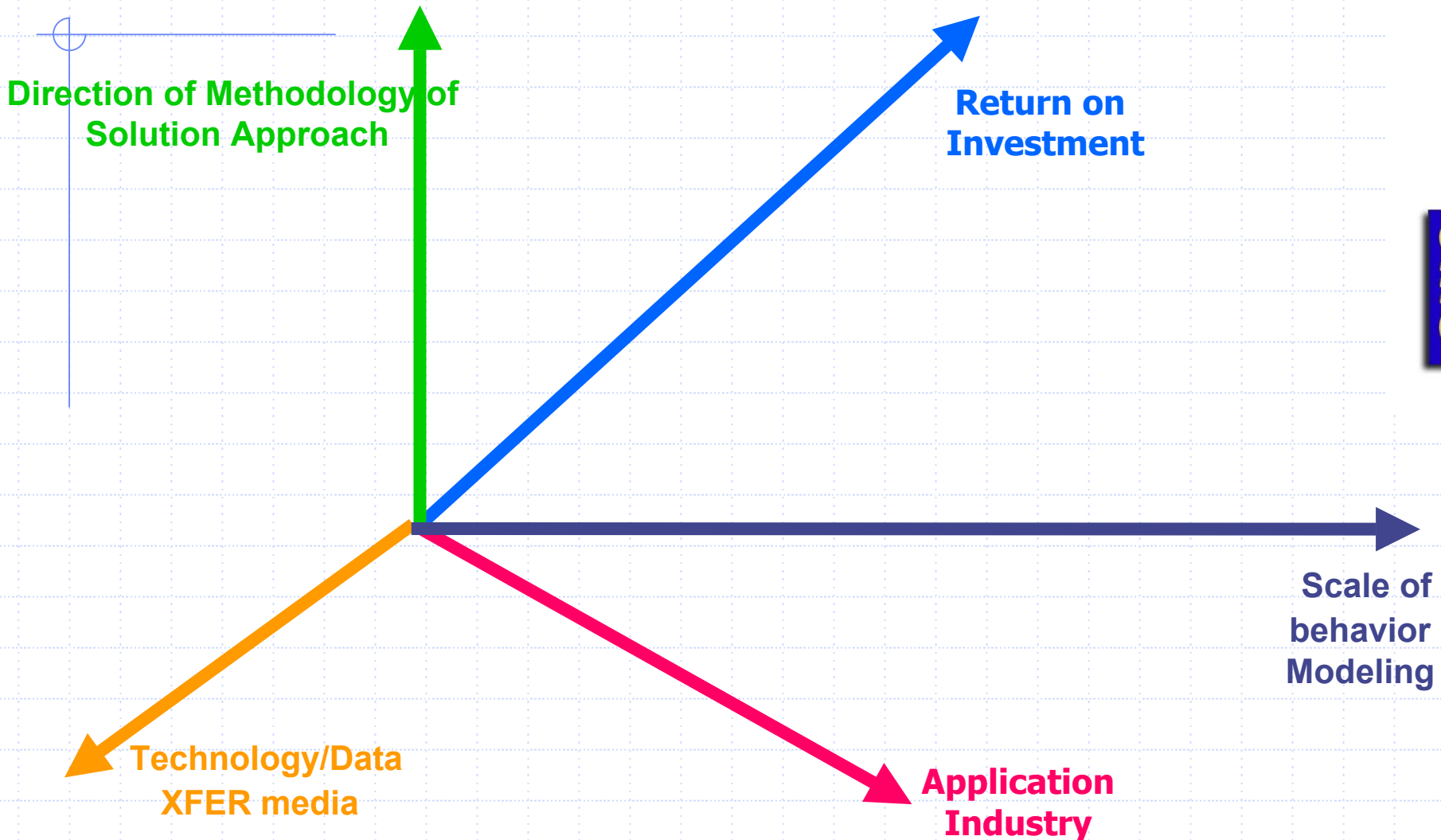
State Variables: Mach, Roll rate

Objective Function: Dissipated energy, Cost

Nonlinear Constrains: Positive definite dissipated energy and cost etc.

Implementation space description: Dynamic Distributed Virtual Environments

Computational Materials Science & Technology Activity Space



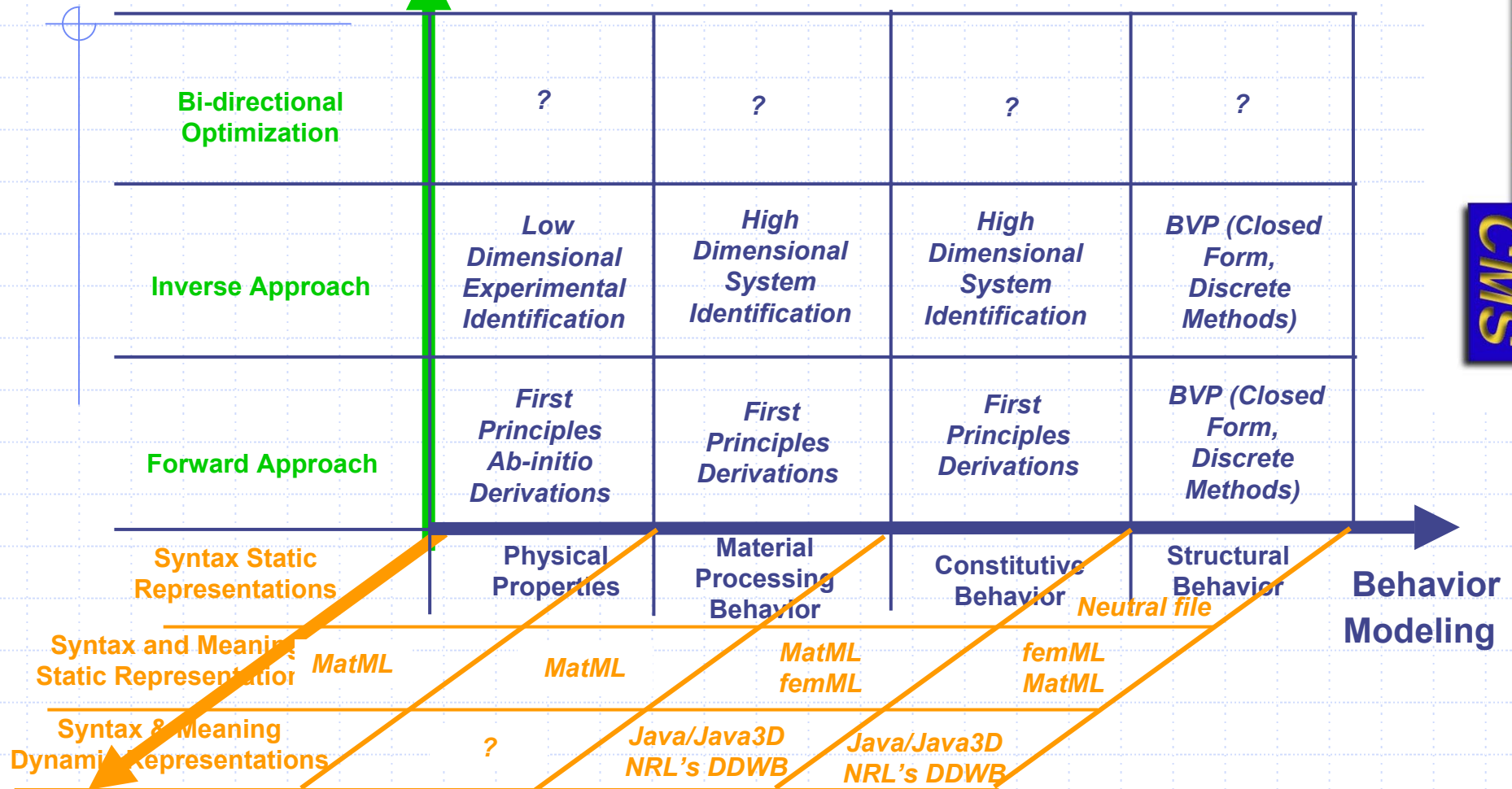
naval research laboratory

code 6304

CMS

Computational Materials Science & Technology Activity Subspace

Methodology of Solution Approach



naval research laboratory
code 6304

CMS

Technology/Data
XFER media

Motivation

Science Applications push

- **Distribution of static digital information through the WWW**
 - ✓ **Multiplicity of custom & commercial applications**
 - ✓ **Manufacturer Data Sheets**
 - ✓ **Materials Databases**
 - ✓ **R & D Publications**
 - ✓ **etc.**
- **Collaborative dynamic computing through the WWW**
 - ✓ **Distributed Applications**
 - ✓ **Problem Solving Environments**
 - ✓ **Virtual Design & Prototyping**
 - ✓ **Agent-based Applications**

Technology Pull

- **Multi-industry XMLware proliferation**
- **XML-Java Integration**
- **XML-Data Base Technology Integration**
- **XML-middleware plethora**



Background: Current state



◆ FILE FORMATS

- Lots of custom CAD exchange formats (ACIS, Parasolid, IGES (flavored & standard), STEP, STL, VDAFS, CATIA, CADD5 etc.)
- Very few custom FEM model exchange file formats (STEP 209)
- Very few EDI file formats (ANSI X12, EDIFACT)

◆ DATA exchange and interchange tools

- Custom applications (FEMAP)
- Custom translators

Background (2): state of the art



◆ TECHNICAL RESOURCES

- AP209 ISO/DIS 1030-209 Composite and Metallic Structural Analysis and Related Design
 - ◆ Satisfies the need for the exchange of computer-interpretable composite and metallic structural product definitions, including product shape, associated FEA models, material properties and analysis results.
 - ◆ Currently has a Non-XML markup description.
 - ◆ Ongoing efforts for developing XML translation and DTD
- XSIL: Extensible Scientific Interchange Language
 - ◆ Satisfies the need for flexible, hierarchical, extensible, transport of scientific data objects (vectors, arrays, tables, etc.
 - ◆ XML-based with existing DTD.
 - ◆ Non application specific/optimized.

Background (3): other efforts



◆ Business Industry Resources

- ANSI X12 and UN/EDIFACT efforts for Electronic Data Interchange (EDI)
 - ◆ Heavy industry support
 - ◆ Plethora of EDI/XML resources and examples
 - ◆ Object facilitation layers allowing OMG, NOF and UML technologies to be used with XML repositories

Problems



- **Integration** of FEM models encoded in multiple data formats from multiple data sources, with current end-user applications and future data exchange systems between applications.
- Data **interpretation** varies from data source to data source and therefore introduces semantic correctness uncertainty that destroys robustness of **interoperability** between applications and data receptacles.

Bigger problem of the moment

◆ We want to use the Internet as the Network for everything

- moving
- publishing
- **engineering**
- finding
- processing
- commerce
- business
- inter/intra/extra

- This requires standards
 - for the network (TCP/IP)
 - for delivery (HTTP)
 - for programs (Java)
 - for security (Public Key)
 - for content w. meaning (...)

Oh yes – and we still want to be able to use our old systems and content!



Interpretation Problem

- **Neutral/Custom file format case:**
 - **FACT:** The majority of custom codes can import and export into non generic highly custom file formats
 - **PROBLEM:** custom formats describe syntax of the data. Meaning is defined in the manuals.
- **WWW case:**
 - **FACT:** The majority of Web-based documents are prepared using the Hyper-Text Markup Language (HTML).
 - **PROBLEM:** HTML describes the presentation/typesetting of data but not their meaning.

Custom formats and HTML require that users have detailed knowledge of:

- the data meaning
- the underlying document structure

before using the data in “downstream” or “upstream” applications



Interoperability Problem

Data interpretation varies from data source to data source and therefore introduces semantic correctness uncertainty that destroys robustness of interoperability between applications and data receptacles.



Solution: Utilize XML Technology

Advantages of XML

- ◆ Universal Standard format for data interchange/exchange
- ◆ Simultaneous Semantic and Syntactic encapsulation
- ◆ Human-readable
- ◆ Machine-readable (easy to parse)
- ◆ Possible to validate
- ◆ Extensible
 - can represent any data
 - can add new tags for new data formats
- ◆ Hierarchical structure (nesting)
- ◆ *Great amount of tools that facilitates understanding, usage and implementation*



What is XML? - Core idea

`<bold>Apple</bold>`

`<fruit>Apple</fruit>`

`<computer>Apple</computer>`

`<computerManuf>Apple</computerManuf>`

`<structure>Apple</structure>`

`<materialSys>Apple</materialSys>`

`<FEMmodel>Apple</FEMmodel>`

- ◆ Does not drop or infer meaning from syntax but it embeds meaning together with syntax



What is XML?

- ◆ Extensible Markup Language
- ◆ XML is a meta-language for developing an unlimited number of special-purpose data languages
- ◆ A W3C standard approved as “Recommendation” in February 1998
- ◆ Core of a family of generic standards
- ◆ A simplified form (subset) of SGML
- ◆ A standard framework for encoding agreements about communication

Examples of S&T related efforts

- ◆ **MatML Extensible Markup Language (XML) for Materials Property Data** is a DTD with examples under development for the exchange of material properties information. It's spearheaded by Dr. Ed. Begley at NIST and a working group.
- ◆ **CML Chemical Markup Language 1.0** Reference with examples of Chemical Markup Language
- ◆ **GAME DTD (Genome Annotation Markup Elements)** is a syntax for the exchange of genomic annotation.
- ◆ **GEML** The Gene Expression Markup Language is a file format for storing DNA microarray and gene expression data.
- ◆ **GXL - Graph Exchange Language** is an XML language designed to be a standard exchange format for graphs, and to support interoperability between graph-based tools.
- ◆ **Mathematical Markup Language (MathML) Version 2.0** MathML is an XML application for describing mathematical notation and capturing both its structure and content.
- ◆ **MODL** Molecular Dynamics Markup Language is used to help make sense of the huge amounts of data typical of chemical simulations.
- ◆ **Systems Biology Markup Language (SBML)** is an XML-based language for describing simulations in systems biology.
- ◆ **XGMML (eXtensible Graph Markup and Modeling Language)** is an XML application based upon Graph Modeling Language (GML) that uses XML to describe graphs rather than GML's text format.

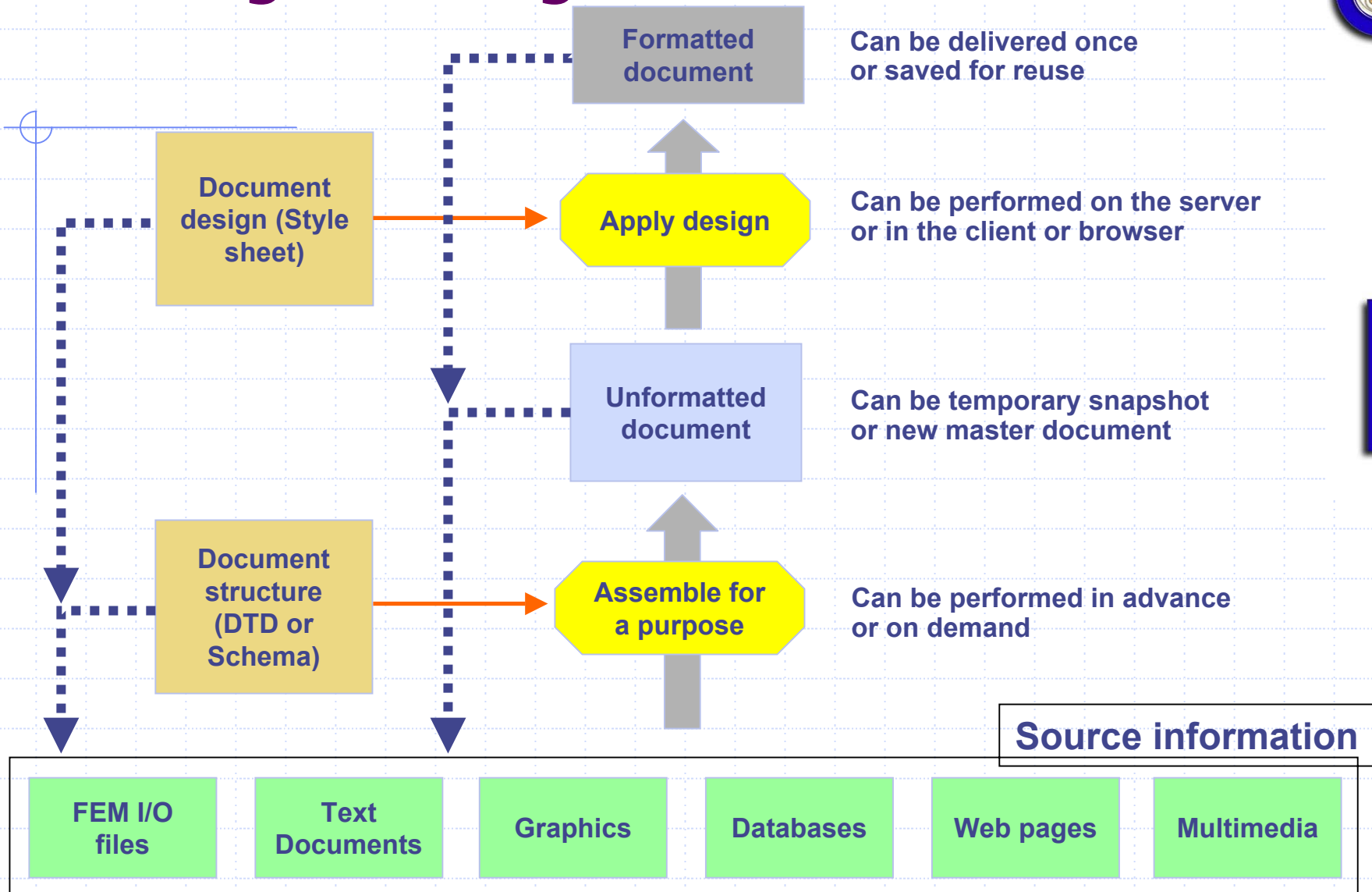


Classes of Application

- ◆ **information delivery** – enabling information to be assembled from multiple sources to meet individual requirements
- ◆ **inter-application messaging** – enabling data transfer within and between organizations to facilitate EDI and system interoperability
- ◆ **intra-application messaging** – to supplement or replace such protocols as CORBA, COM/DCOM and Enterprise Java Beans in the development of distributed computing applications



Putting it all together



Very Efficient Tools i.e. BizTalk Mapper or DataJunction

- ◆ Map between DTDs/schemas
- ◆ Intuitive GUI
- ◆ Extensible
- ◆ Produces XSLT



The screenshot displays the Microsoft BizTalk Mapper interface. The main window is titled "Microsoft BizTalk Mapper - CanonicalReceiptTod98b-CTRL.xml". It features a central workspace with a complex network of lines representing mappings between source and destination elements. On the left, the "Source Specification" tree shows a hierarchy starting with "Group" and "Document", containing various fields like "name", "sender", "receiver", "major_version", "id", "ack_code", "docs_received", "docs_accepted", "docs_included", "type", "minor_version", "id", "ack_code", "records_received", "Details", "SyntaxErrorCode", "Record", "tag", "position", "loop_id", "error_code", and "Field". On the right, the "Destination Specification" tree shows a similar hierarchy with elements like "S003", "UCN03", "UCN04", "UCN05", "S011", "UCSLoop1", "UCF", "UCF01", "S006", "S00601", "S00602", "S007", "S00701", "S00702", "UCF04", "UCF05", "UCF05", "S011", and "UCF05". Below the main workspace, there are two tables: "Source Attributes" and "Destination Attributes".

Property	Value	Property	Value
Name	CanonicalReceipt	Name	EFACT_d98b_CTRL
Description	Canonical Receipt Report	Description	Syntax and Service Report Message
Type	Element	Type	Element
Model	Open	Model	Closed
Content	Element Only	Content	Element Only
Schema Name	CanonicalReceipt	Schema Name	EFACT_d98b_CTRL
Standard	>XML	Standard	EFACT
Standards Version		Standards Version	d98b
Deployment Time		Deployment Time	CTRL



Java Technologies cross leveraging

Why Java/XML?

- ◆ XML Structures can map *homomorphically* to Java Objects
- ◆ XML tags map *well* to Java Objects
 - late binding
 - hierarchical (OO) data model
- ◆ Unicode support in Java
- ◆ Portability
- ◆ Network friendly



XML and Object Mapping



◆ Java -> XML

- Start with Java class definitions
- Serialize them - write them to an XML stream
- Deserialize them - read values in from previously serialized file

◆ XML -> Java

- Start with XML document type
- Generate Java classes that correspond to elements
- Classes can read in data, and write in compatible format (shareable)

XML-Java Endless possibilities



- ◆ light-weight asynchronous processes
implementation of distributed, migrating,
dynamic and intelligent agents for each one
of the femML entities
- ◆ composition/synthesis of complex models
just by simple messaging between dynamic
object-ware units automatically produced by
XML<->Java toolsets

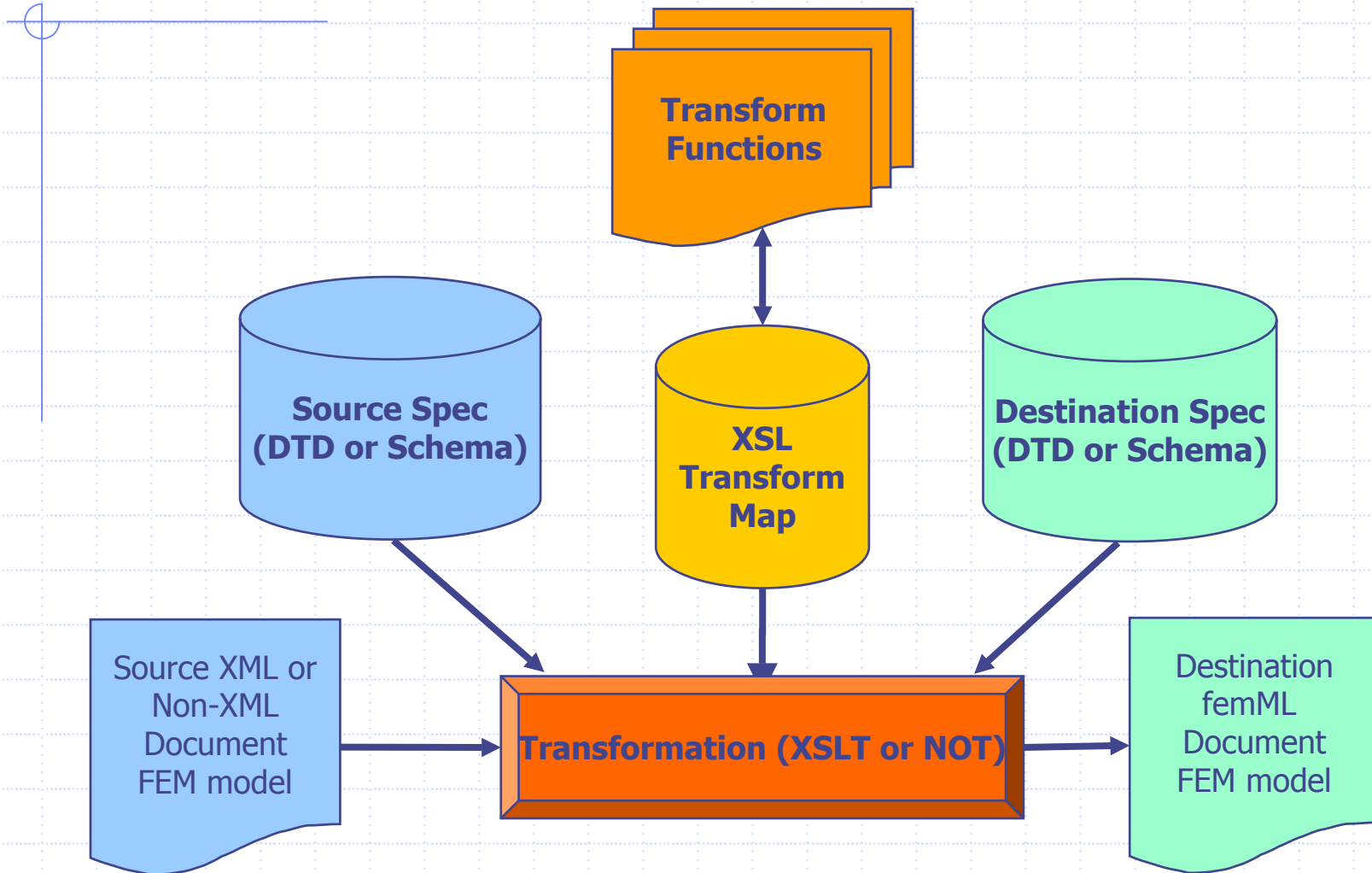
femML Objectives



- ◆ Define a standard for the exchange of FEM data (including product shape, associated FEM models, material properties and analysis results) that will allow a *person* or a *computer application* to interpret and use the data *regardless of its source or target* and *regardless of the taxonomic order of the FEA model*.
 - Set of XML Tags
 - Document Type Definition (DTD) or/and Schema
- ◆ Define and develop a set of examples that follow the standard.
- ◆ Define and develop a set of tools for the utilization of this standard from and to other applications.
- ◆ Develop examples of using this tools.

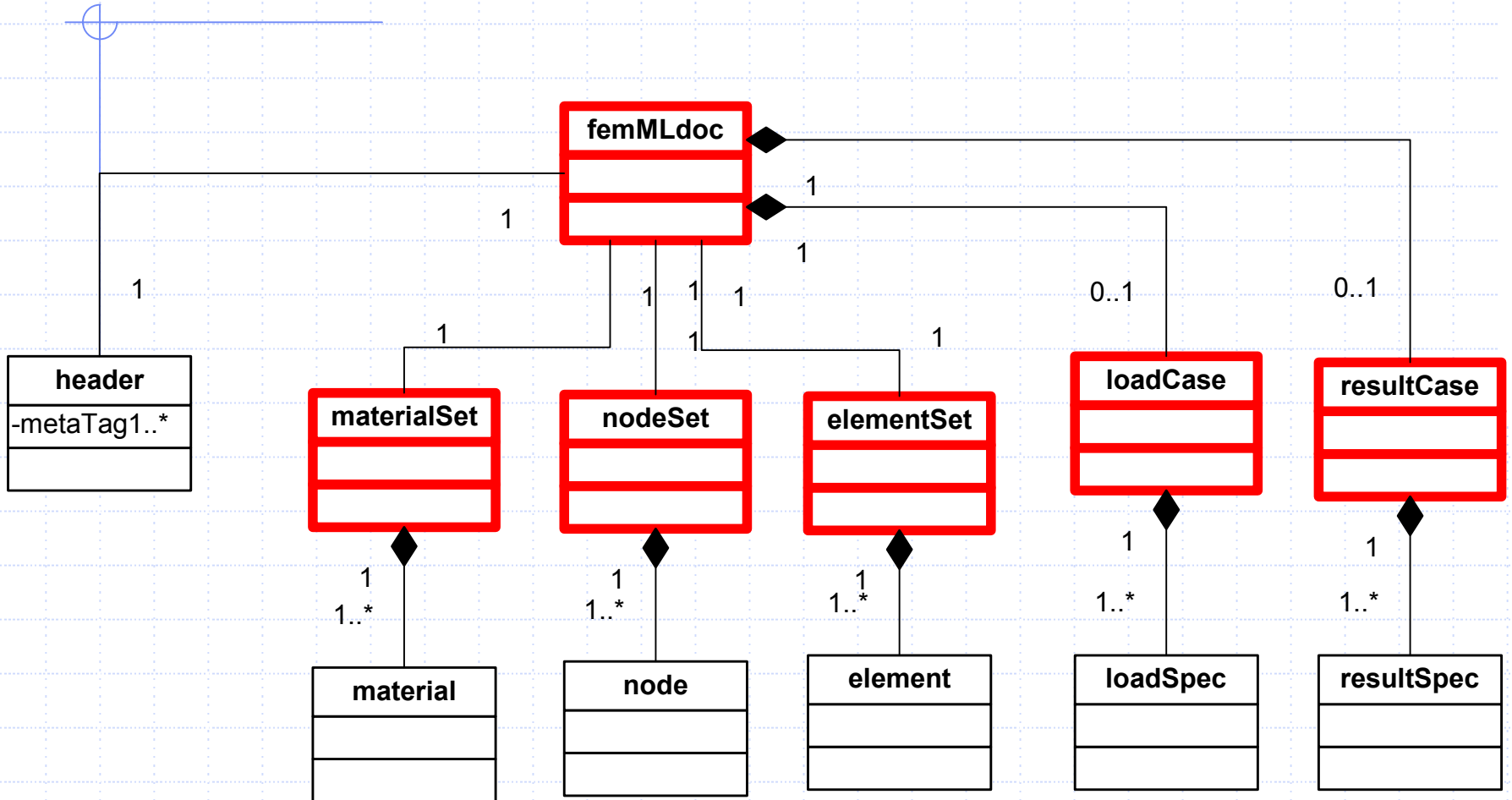
Approach: The XML S2S exchange

Employ a Station to Station (S2S) exchange based on XML technology



Current femML document structure

UML representation of femML DTD



Issues to be resolved

◆ Accommodate the entire set of possible system representations:

- **Finite Element**
 - ◆ Structured
 - ◆ Unstructured
 - ◆ Blocked
 - ◆ Hierarchical
 - ◆ Spectral
 - ◆ Stochastic
- **Finite differences**
 - ◆ Structured
 - ◆ Unstructured
 - ◆ Blocked
- **Boundary elements**
- **Hybrid elements**
- **Non-Discrete Model Representations**
 - ◆ Analytic BVP Symbolic Solutions
 - ◆ Continuous



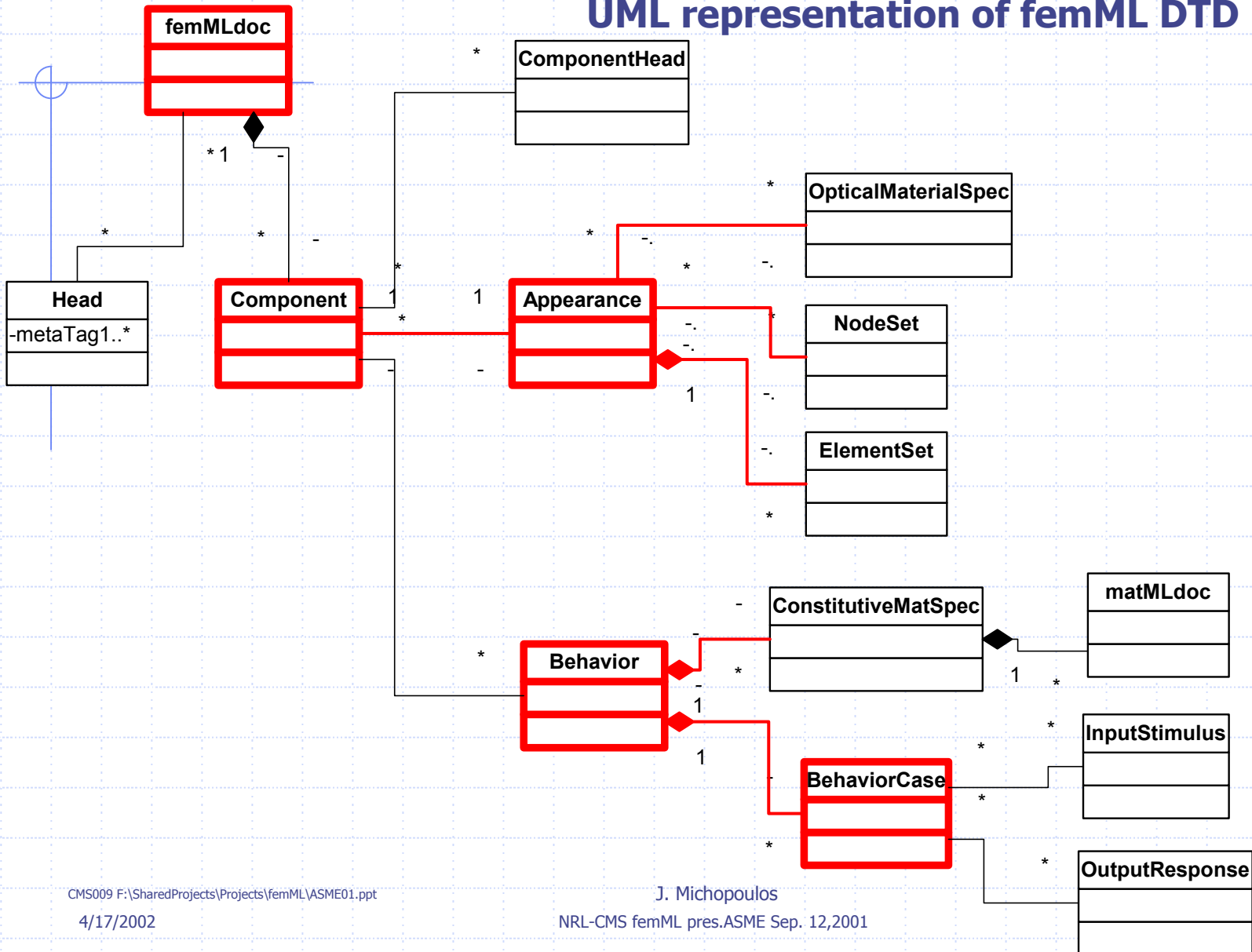
Issues to be resolved (cont.)

- ◆ **Separation between Appearance and Behavior**
- ◆ **Utilize/Leverage existing XML representations for XML substructures when available through namespace uniqueness (i.e. MatML for material properties specification)**
- ◆ **Maintain transformability to other Data exchange formats (i.e. thing isomorphically to existing DTDs like XSIL, X3D etc.)**
- ◆ **Maintain View-ability of implicit or explicit scene graph representations of the appearance components of datasets through providing transformation capability by appropriate DTD/Schema Factorability**
- ◆ **Maintain factoring and composition homomorphism between femML documents and structural models**
- ◆ **DTD or/and SCHEMA**
- ◆ **Incremental vs. Shotgun Approach**



Potential femML document structure

UML representation of femML DTD



Desired Approach Methodology



- ◆ Form working group with members from Academia, Industry, Government, Professional societies and Standards Organizations
- ◆ Identify issues to be resolved and their priority
- ◆ Develop and implement strategy for addressing issues
- ◆ Utilize "Open Source Development Network" resources like the "SourceForge"
<http://sourceforge.net/> development and deployment repository for DTD/SCHEMA/Examples/XSLTware and custom format translator components

Schedule



Steps	femML	MatML
Establishment of working group	2000-01	2000
Delineation of the scope and specifications	1999-02	2000-01
Development of the formal Document Type Definition and/or Schema	2001-...	2000-01
Development of a catalog of examples	2000-...	2000-...
Application development and acceptance testing	Future	Future
Dissemination	Future	Future

Open Call for Participation

Contact Info

◆ femML

- Contact: J. Michopoulos (john.michopoulos@nrl.navy.mil)
- URL: <http://cms.nrl.navy.mil/pub/femML>
- e-mail: femML@cms.nrl.navy.mil

◆ MatML

- Contact: E. Begley (begley@nist.gov)
- URL: www.ceramics.nist.gov/matml/matml.htm
- E-mail: begley@nist.gov

THANK YOU FOR YOUR ATTENTION!

