NRL CODE 6304 CMS
Composite Materials & Structures Group

finite element modeling Markup Language

J. Michopoulos
(johnM@cms.nrl.navy.mil)
CMS group, Code 6304
NRL, Washington DC 20375

FEMCI Workshop 2002,
NASA Goddard Space Flight Center

May 23, 2002

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

naval research laboratory code 6324

CMS

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?

Model space description:

scale

Nano physico-chemical molecular behavior models ○ ○ ○ Micro mechanical-process and thermal coupled Behavior models ○ ○ ○ Macro fluid-structural and thermal coupled Behavior models

mach

Roll rate

Molecular characteristics

Curing profile characteristics

Approach space description: Design Optimization

Design Variables: Molecular Characteristics, Curing Profile Characteristics

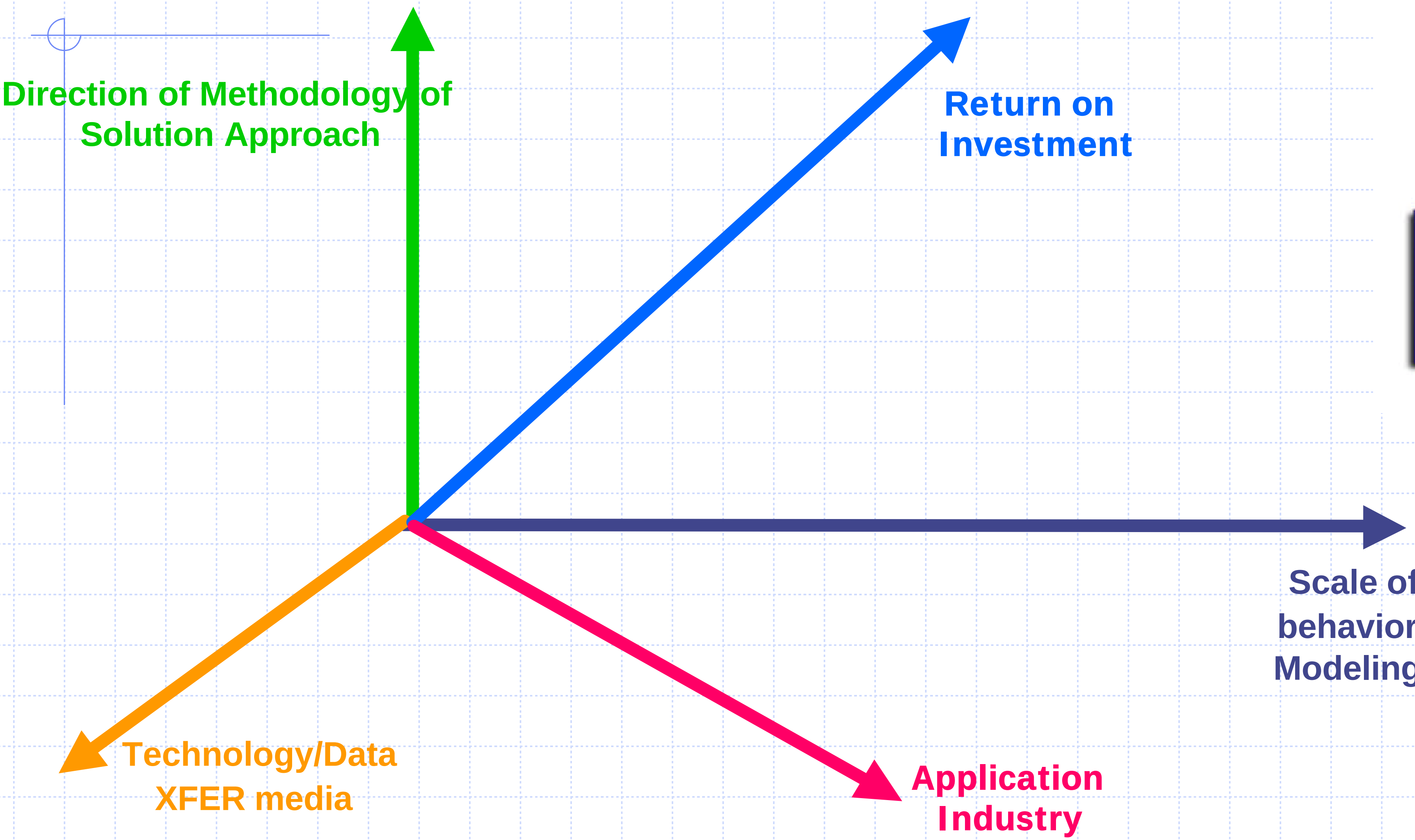
State Variables: Mach, Roll rate

Objective Function: Dissipated energy, Cost

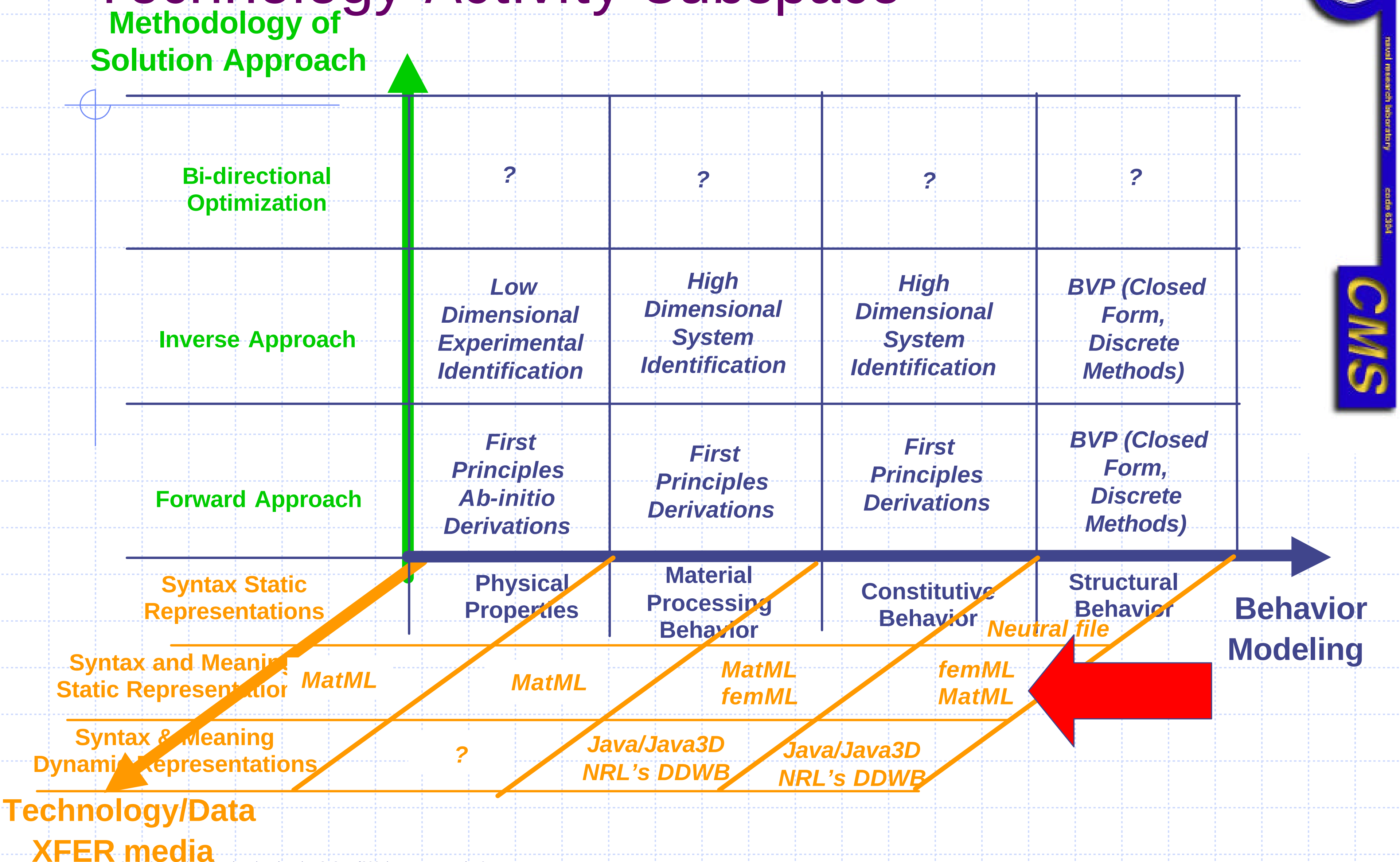
Nonlinear Constrains: Positive definite dissipated energy and cost etc.

Implementation space description: Dynamic Distributed Virtual Environments

Computational Materials Science & Technology Activity Space



Computational Materials Science & Technology Activity Subspace



Motivation

Science Applications push

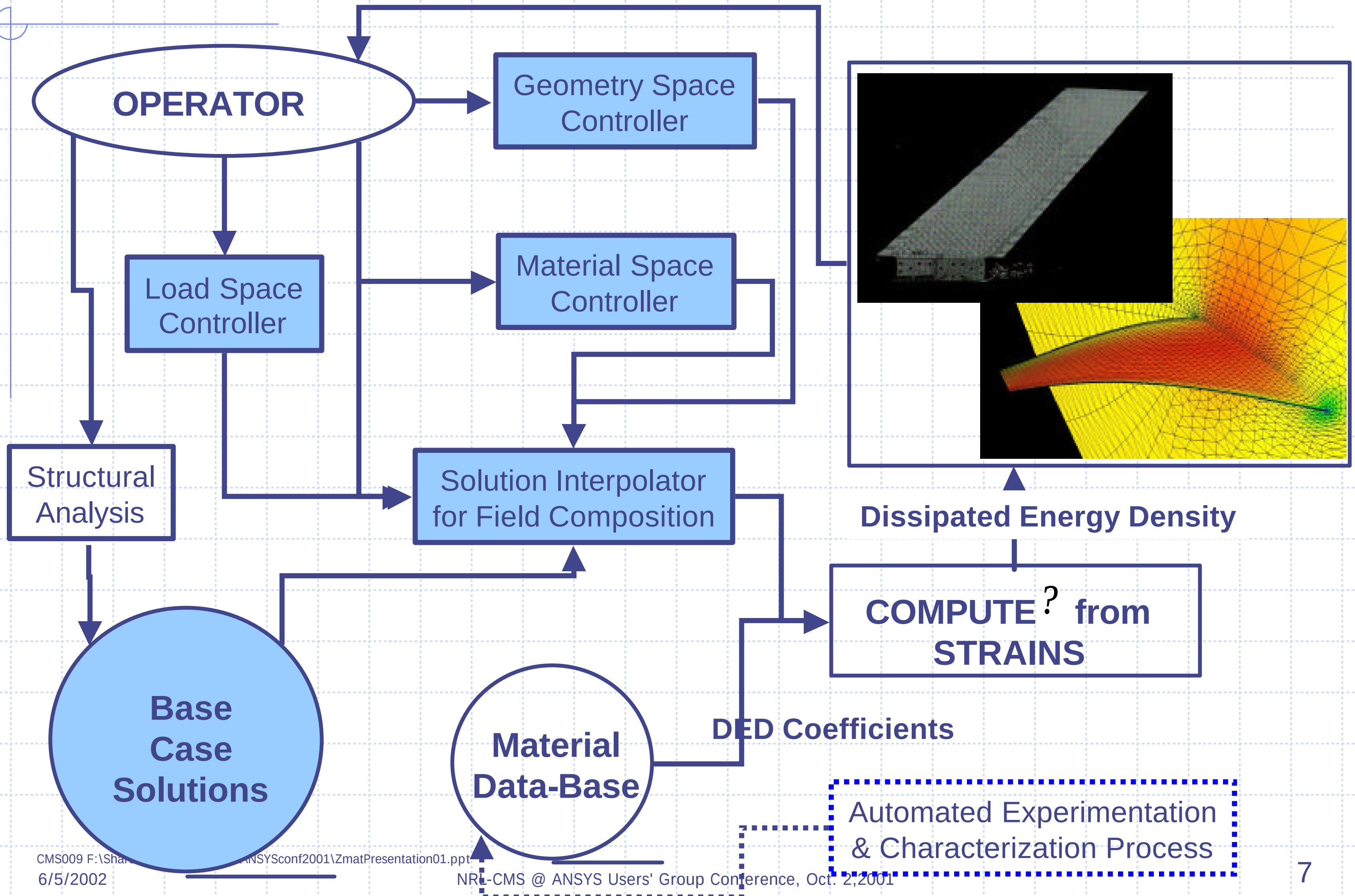
- Distribution of static digital information through the WWW
  - Multiplicity of custom & commercial applications
  - Manufacturer Data Sheets
  - Materials Databases
  - R & D Publications
  - etc.
- Collaborative dynamic computing through the WWW
  - Distributed Applications
  - Problem Solving Environments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

Technology Pull

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

Data Driven Mutliphysics Simulation

DDWE Simulator Architecture



FEM EDI^3 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.

Background: Current state

- FILE FORMATS
  - Lots of custom CAD exchange formats (ACIS, Parasolid, IGES (flavored & standard), STEP, STL, VDAFS, CATIA, CADDS5 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

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!

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

- 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.

Examples related to our efforts

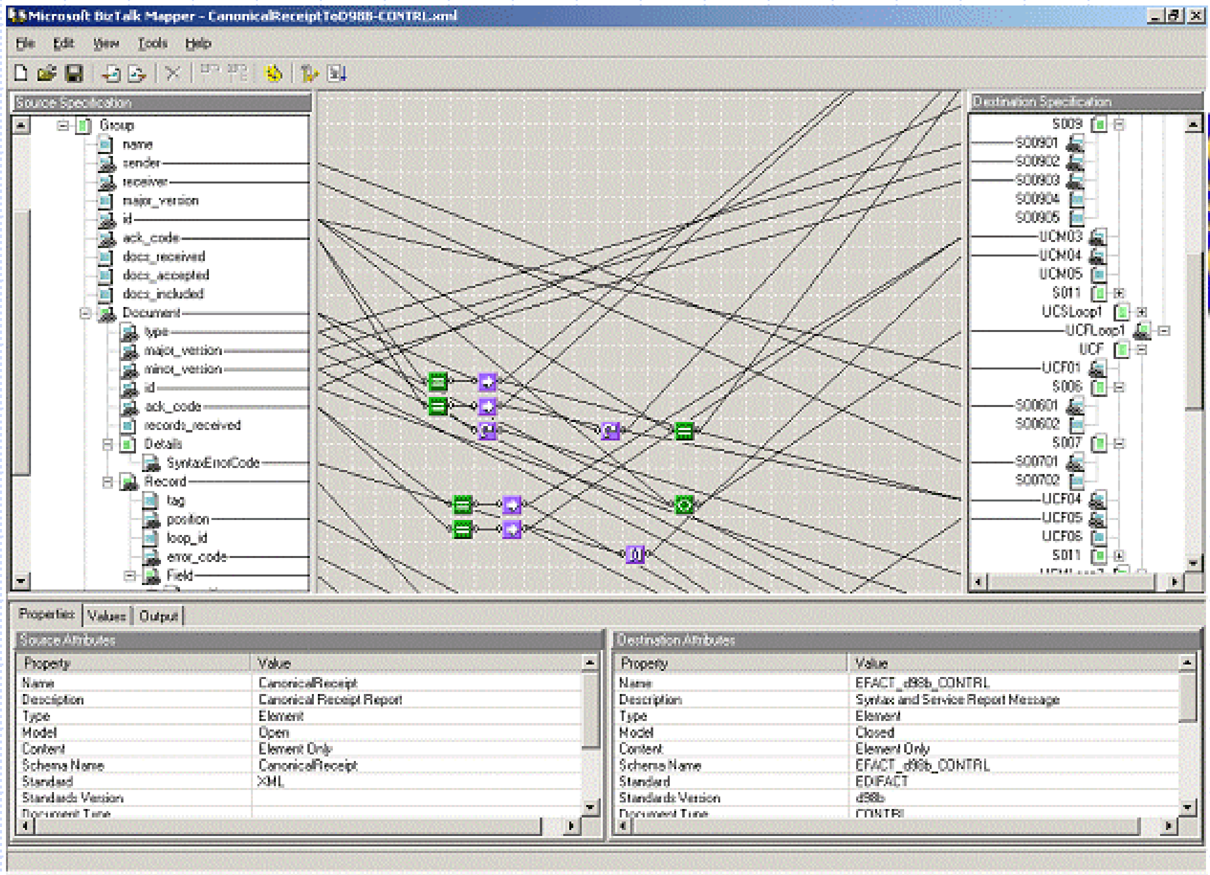
- MatML Extensible Markup Language (XML) for Materials Property Data is a DTD with examples under development for the exchange of material properties information. It's spearheaded by Ed. Begley at NIST and a steering group.
- XSIL The Extensible Scientific Interchange Language (XSIL) is a flexible, hierarchical, extensible, transport language for scientific data objects. Coordinated by Roy Williams at Center for Advanced Computing Research at the California Institute of Technology.
- FieldML-MeshML-RegionML The Physiome set of languages for describing time-varying and spatially-varying fields. The language will eventually serve as a replacement for the ".exelem" and ".exnode" files used by ***CMISS***, and is intended to be useful for other groups interested in the field description problem. Coordinated by Warren Hedley, at the Engineering Science Department at the University of Auckland.

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

Very Efficient Tools
i.e. BizTalk Mapper or DataJunction

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



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

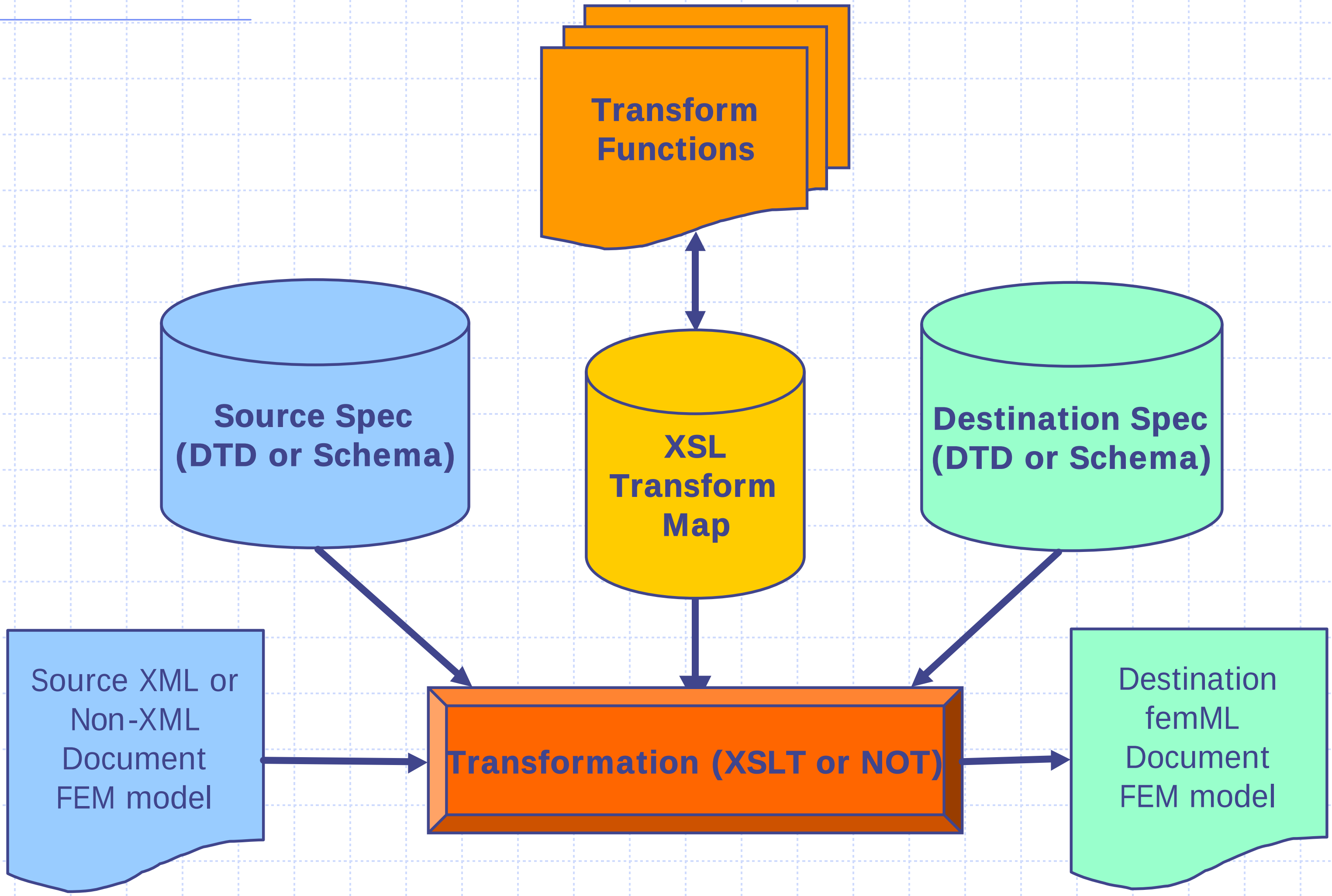
- light-weight asynchronous processes implementation of distributed, migrating, dynamic and intelligent agents for each one of the femML entities.
- composition/synthesis of complex models just by simple messaging between dynamic object-ware units automatically produced by XML<->Java toolsets (SOAP,UDDI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

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

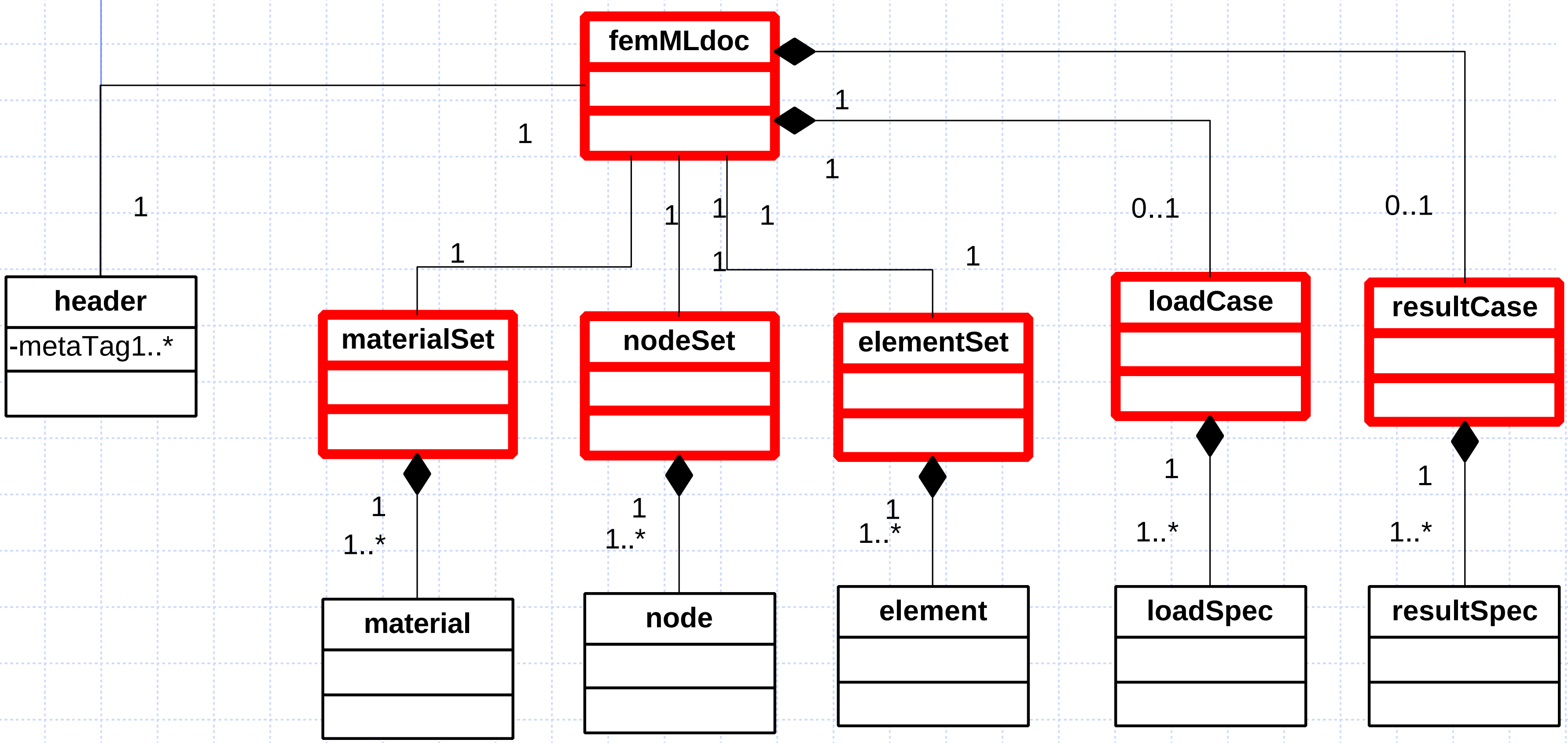


Current femML status

- created first (v1.02) architecture of femML with associated DTD and Schemas
- built femML to ANSYS S2S tools except of femML direct parser in APDL
- adopted matML for material properties
- adopted a matML variation for composites
- created a decomposable version (v2.99b) of femML architecture with corresponding DTD & Schema

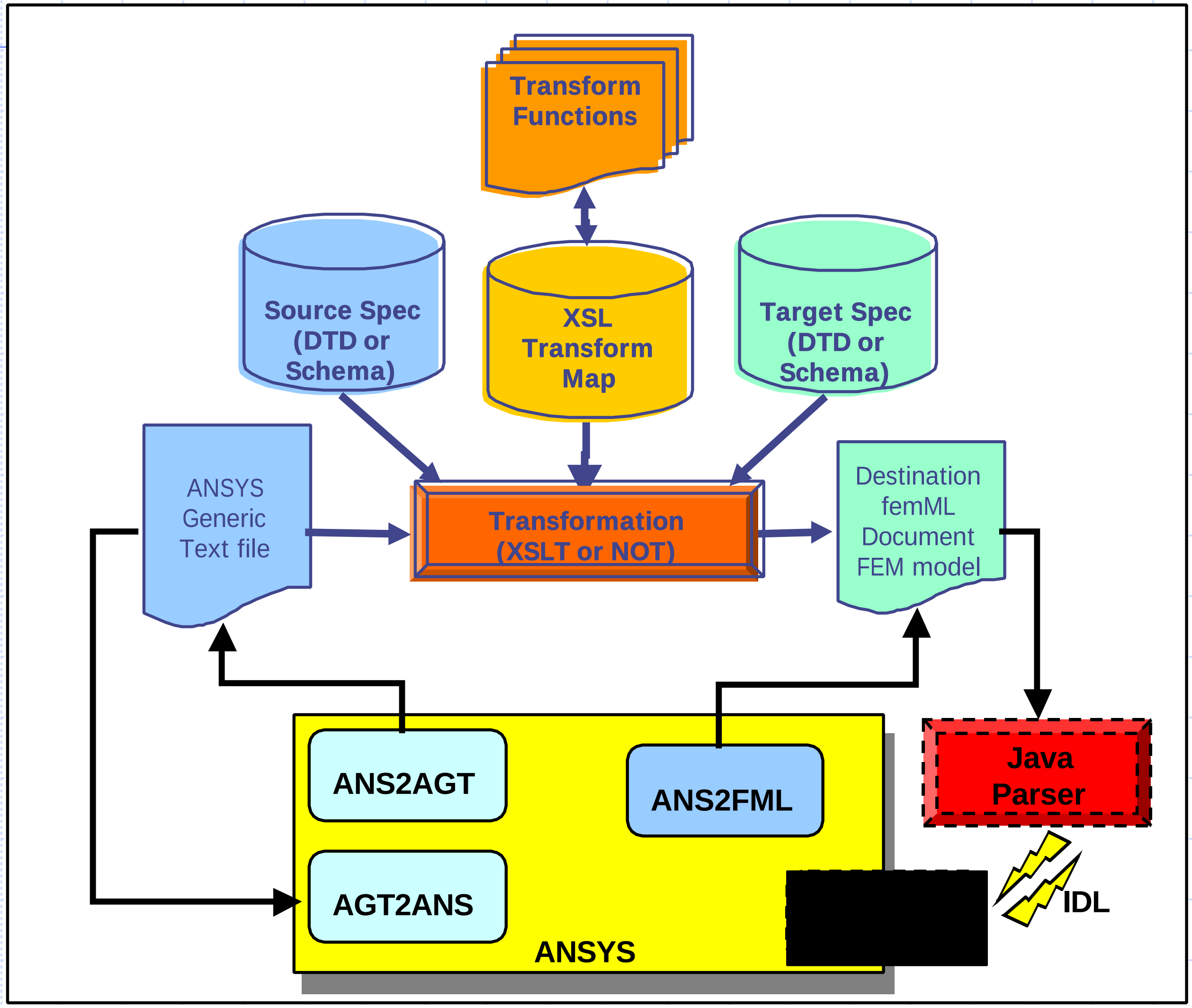
Current femML document structure

UML representation of femML DTD



Approach: The XML S2S exchange

ANSYS based Station to Station (S2S) exchange



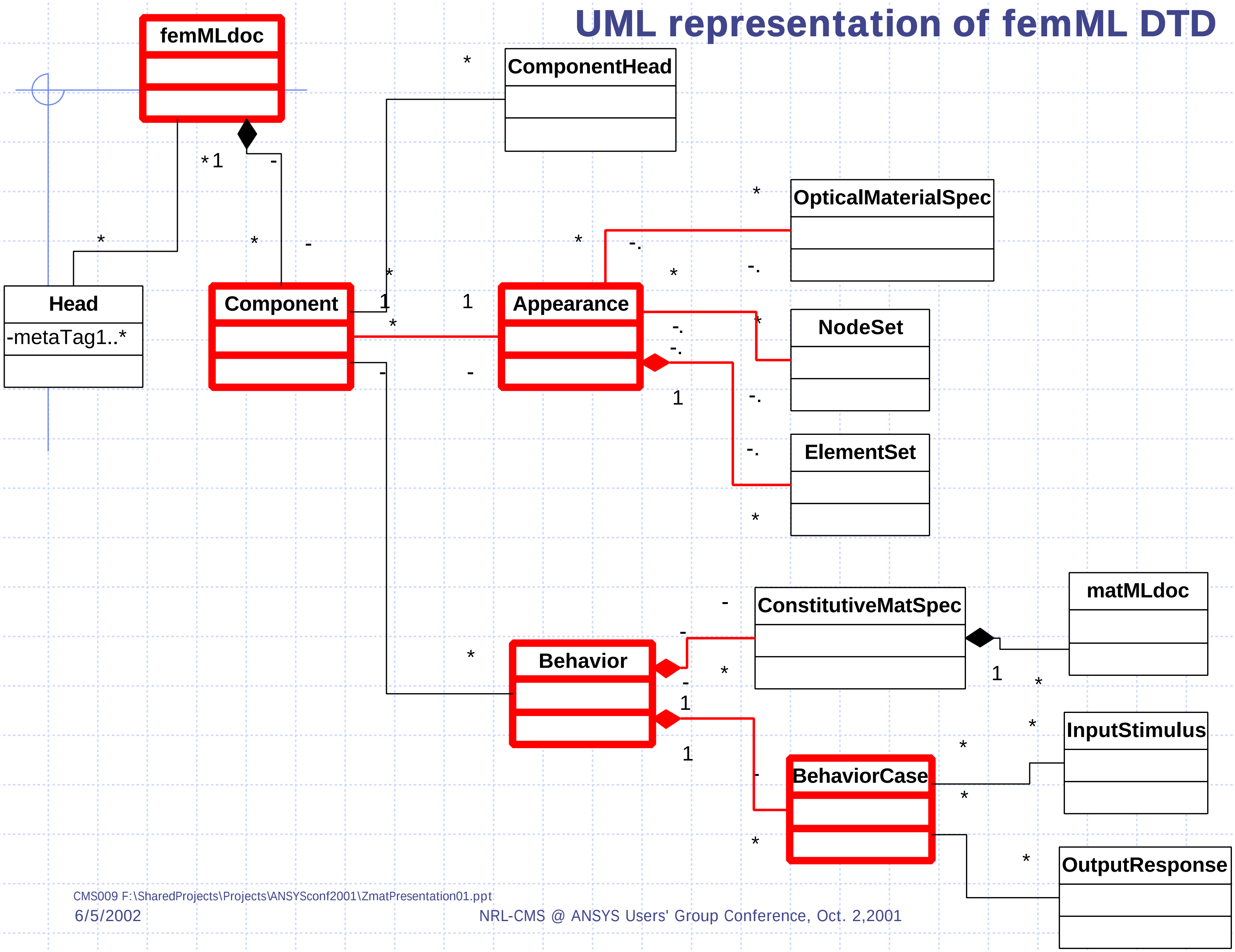
Issues to be resolved

- **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



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

Open Call for Participation Contact Info

- femML
  - Contact: J. Michopoulos (**john.michopoulos@nrl.navy.mil**)
  - URL: **www.istos.org/femML (default site)**
  - **URL: femml.sourceforge.net (developer's site)**
  - **URL: sourceforge.net/projects/femml (code site)**

  - e-mail: **femML@cms.nrl.navy.mil**

THANK YOU FOR YOUR ATTENTION!