Proceedings of DETC'02
ASME 2002 Design Engineering Technical Conferences and Computers and Information in Engineering Conference
Montreal, Canada, September 29-October 2, 2002

DETC2002/CIE-34406

DEVELOPMENT OF THE FINITE ELEMENT MODELING MARKUP LANGUAGE

J. G. Michopoulos

U.S. Naval Research Laboratory
Composite Materials and Structures Group, Code 6304
Washington DC, 20375
USA
e-mail: john.michopoulos@nrl.navy.mil

ABSTRACT

The **f**inite **e**lement **m**odeling **M**arkup **L**anguage *(femML)* effort is addressing the problems of data interpretation and exchange for intra- and inter- application interoperability in the Finite Element Modeling domain. This is achieved through the development of an extensible markup language (XML) variant for finite element model data that will permit the storage, transmission, and processing of finite element modeling data distributed via the World Wide Web and related infrastructure technologies. The focus of this work was to utilize the XML's power of semantic encapsulation along with the existing and continuously improving associated technology to develop a dialect for exchanging FEM data across various codes with heterogeneous input format syntactic specifications. The main aspects of a finite element definition have been used as archetypes for defining the XML element taxonomy definitions. Namely, the geometry, the material, and the loading aspects of a structural component specification are used to define the first level elements of the associated Document Type Definition (DTD). The element list has been amended with a behavior element specification that represents the solution data to be exchanged or visualized. Various tools have been developed to demonstrate associated concepts along with the ANSYS set of tools.

NOMENCLATURE

Data Exchange, Interoperability, Data Integration, Finite Element Modeling, Finite Element Analysis, Internet, World Wide Web, Extensible Markup Language.

INTRODUCTION

General problems

The main problems associated with all computationally assisted data exchange, interchange and integration activities can be approached from multiple points of view depending on the needs at hand. However, there is a global point of view that is common to all industries in need of data exchange. In the engineering industries it unfolds as a need for *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. However, data *interpretation* (semantics) varies from data source to data source and therefore introduces semantic correctness uncertainty that destroys robustness of *interoperability* between applications and data receptacles in general.

We have experienced this issue from a very close distance when the time came to implement the Data Driven Design Workbench (D^3W) used as a virtual wind tunnel environment for design of composite structures and qualification and certification of composite materials systems [1]. This architecture has evolved through time and the whole environment along with the Dissipated Energy Density (DED) methodology have been utilized in various applications including health prediction and sensor optimization of smart structures [2-4]. Inter-modular data flow invited the implementation of some highly structured data format for satisfying the data transfer requirements. One of the most dominant issues on the development of this environment was clearly the semantic biasing and preference from each one of

Copyright © 2002 by ASME

the modules as well as the lack of a common representation for the exchanged data. It is exactly for cases like this that eXtensible Markup Language (XML) has been used to develop the finite element Modeling Language (femML).

In addition to this general need, the proliferation of the specific needs of particular domains of application generate a science push for solving the data structure and heterogeneity of meaning problems within the pertinent vertical industry context. More specifically, digital content with the WWW as a transport medium is available in many forms i.e. multiple commercial applications, manufacturers data-sheets, materials databases, and research and development electronic publications, neutral and custom file formats etc. The need for collaborative dynamic computing through the WWW, strengthens the push for solving the heterogeneity problem by imposing a demand for distributed applications, for problem solving environments, for virtual design and prototyping and for agent-based applications. On the other hand, the multi-industry support and proliferation of XMLware, the Java-Database-XML integration technology, and the XML middleware plethora create a technological pull for the utilization of XML-based solutions.

Recognition of the problem by the industry

The data exchange, interchange and integration problems have been recognized very early by multiple industries of human activity that entails data transfer.

Industrial automation technology has improved dramatically over the past decades. CAx systems ("Computer-Aided anything", or: CAD, CAM, CAE, CIM, etc.) have provided engineering applications with high-performance solutions. Integration of these technologies is a major issue for industrial competitiveness. From numerical control (NC) in the fifties, through the first design graphics applications and computer controlled production operations in the sixties, Computer Numerical Control (CNC) and Distributed Numerical Control (DNC) in the seventies, and Flexible Manufacturing Systems (FMS) and solid model-based design workstations in the eighties, automation technology has continued to advance and become more sophisticated in order to meet the individual needs of industry. In terms of horizontal integration the CAD industry has responded to geometry data integration and exchange with multiple specific file format specifications. Examples are ACIS, Parasolid, IGES (flavored & standard), STEP, STL, VDAFS, CATIA, CADDS5 etc. [5]

However, as industry moves into the 21st century, a new industrial need is becoming the critical problem to solve: the vertical integration of these diverse automation systems (e.g., CAD, CAM, CIM, CAE) [6].

The complex nature of engineering data may hinder the integration of engineering applications. The major “stumbling blocks" that prevent the effective integration of CAx systems are [7]:

1. Current CAx systems have been designed to input and output data rather than information; and

2. Current CAx tools operate on different levels of abstraction of the mechanical product.

Therefore, information (data with meaning) modeling is a major issue for CAx systems integration. Moreover, data has to be transferred between applications.

An obvious recognition of the importance of XML for information exchange in general, can be evidenced by the plethora of special XML variants developed by and for many non-CAx industries.

Attempts to solve the CAx data exchange problem

Non XML Efforts

The ever present need for data translation to fit the receiving system's data model has been identified as the dominant problem of application integration. To deal with this problem, the International Standards Organization (ISO) launched the STandard for the Exchange of Product model data - STEP (ISO 10303-1 1994) [8], aimed at the representation of all information about a product throughout its entire life cycle.

STEP allows different applications to exchange information using a standard format. All data models in STEP are normalized (i.e., in conformity with the normal forms, described in section 2.1.2) and written in EXPRESS (ISO 10303-11 1994) [9], an "object-flavored information model specification language" [10] allowing for the specification of complex data models with multiple inheritance.

Relative to finite element modeling efforts, the data exchange problem has been traditionally cast under the framework of the product data exchange (PDE) category for most of the historical efforts. Early data exchange specifications focused primarily on geometrical data. Among these were proprietary specifications like Autodesk's DXF, and national standards such as IGES (United States), SET (France), and VDA/FS (Germany). The most significant of these efforts in terms of FEM data representation, is the AP209 ISO/DIS 10303-209 or the STEP 209 Composite and Metallic Structural Analysis and Related Design standard[11]. STEP is a complex standard with huge-sized documents, and was developed as if it was a database itself, adopting the ANSI/SPARC architecture for database systems [12]. Its most significant characteristic is that it allows transfer of conceptual information content in addition to raw data. The standard is comprehensive and is made out of a very extensive but well structured document series [8]. Perhaps its massive specifications and custom and proprietary related tool availability are its two greatest disadvantages.

XML Efforts

With the advent of XML and especially since its being adopted as a standard specification by the World Wide Web Consortium (W3C) on 1998 [13], many applications became available very quickly. However, only recently we have seen a utilization of XML technology for the engineering applications industries.

Copyright © 2002 by ASME

It's indicative of the trend that proprietary products to translate STEP documents to an XML form have already been developed to facilitate STEP document transfer. Some companies have decided to integrate XML technology with their existing product lines.

Router Solutions Inc. has recently announced [14] their custom CAx integration solution strategy.

eXT, is an XML-based 3D MCAD interoperability standard recently introduced by UGS. In short, UGS is wrapping XML around its Parasolid XT format [15,16].

Autodesk Inc.[17] has also recently announced the new XML/Data Extension tool for its AutoCAD 2000i family of products. The XML/Data Extension is part of a broader Autodesk XML initiative to create a common, open standard for delivering design data to the Web, ensuring compatibility between products in different segments of the design industry, and facilitating e-commerce around design specifications. In addition, because of the cross-platform, cross-industry acceptance of XML, the XML/Data Extension will allow developers to create tools that help designers share their design data with other mainstream business functions such as marketing, sales, operations, and customer support.

The aecXML project was initiated in August 1999 by Bentley Systems, Incorporated with the desire that it be a unifying force for progress in the development of a project communications framework for architecture/engineering/construction (A/E/C). Bentley has developed an initial specification for aecXML, a framework of XML-based schemas to facilitate communications related to designing, specifying, estimating, sourcing, installing and maintaining construction products and materials over the Internet. Building on the success of aecXML, Bentley and Bluestone have entered a three-year agreement to develop engineering software solutions based on Bluestone's Sapphire/Web Application Server, Bluestone XML Suite™ Integration Server, and Bluestone's comprehensive standards-based, e-business solution [18].

In addition to this proprietary efforts there are four public domain efforts very relevant to the engineering data exchange endeavor. These are the Extensible Scientific Interchange Language (XSIL), the FieldML, the X3D (the successor of VRML) that was just released the summer of 2001 and the MatML work in progress for material properties exchange applications.

The Extensible Scientific Interchange Language (XSIL) is a flexible, hierarchical, extensible, transport language for scientific data objects. It has been developed at [19]. The entire object may be represented in the file, or there may be metadata in the XSIL file, with a powerful, fault-tolerant linking mechanism to external data. The language is based on XML, and is designed not only for parsing and processing by machines, but also for presentation to humans through web browsers and web-database technology. It comes with a Java object model that is designed to be extensible, so that scientific data and metadata represented in XML is available to a Java code. There is also a powerful Swing-based object browser called Xlook that is also designed to be extensible.

The FieldML is an XML-based language for describing time and spatially-varying fields. It is a part of the Physiome Markup languages effort from the university of Auckland in New Zealand [20].

The X3D file format was created to substitute VRML for web based 3D geometries by the WEB3D consortium [21] is basically an XML version of VRML [22] in order to enhance functionality, portability and leverage the Java-XML resources that have been created to support the e-business industry. It can be thought as an XML-interoperable scene graph architecture and encoding standard.

Both of these public formats are very useful to the present effort because they represent an extended body of work capable of dealing with the geometry encapsulation, representation and visualization of FEM geometries.

The MatML effort [23] is being coordinated by the National Institute of Standards and Technology, and is driven by the MatML Working Group, whose members include several ASM International Fellows, and members from various cross industry organizations. In the 2001 MatML conference [24] a steering committee has been founded to organize the strategic objectives of MatML. The main goal of this effort is the development of the MatML Document Type Definition (DTD), and associated examples and applications, that will facilitate the transfer, exchange and integration of material properties data related to the needs of most CAx industries.

CURRENT STATUS OF FEMML

Historical note

As described earlier, femML has been developed as a necessary outgrowth of our core research efforts, to solve the structured data intensive exchange problem between modules of our custom applications or even between custom stand alone application such as RCfem [25] and existing legacy commercial applications such as ANSYS [26] and ABAQUS [27]. The idea for its creation was naturally generated in the summer of 1999 and has been evolving ever since. FemML's development went from the conceptualization to the implementation phase when we searched the XML repositories and found nothing relevant to this. Special encouragement for the final push was the lack of responses when we posted inquiries about the possible existence of such an XML variant on October 6, 2000 at various mailing lists like the XANSYS one for ANSYS users. The only relevant XML variants were XSIL, X3D and MatML, all dealing with a partial collection of issues associated with the FEM data exchange, but none of them was directly dealing with the entirety of the main problem.

FemML Definition

The finite element modeling Markup Language, is an XML variant designed to facilitate the data transfer, exchange, interchange and integration between finite element modeling

Copyright © 2002 by ASME

applications and their modules. It is work in progress that has accomplished the creation of a DTD, a SCHEMA and certain FEM code specific file generation and parsing tools. It is in a pre-recommendation stage and our focus is to offer it for public discussion, development and distribution. This is not to say that femML cannot eventually evolve to forming the kernel of a set of technologies that will not be solving the data exchange problem, but it will lead to an alternative way of working with FEM data discretizations. A way that would use just three component technologies:

- femML as a transport file format,
- an ordinary Relational Data Base Management System (RDBMS) for dynamic data management,
- and a visualization module.

Such a combination of technologies allows composition and factoring of FEM data for the needs of model synthesis and combination as well as the needs for model decomposition and simplification of the design and prototyping industries.

FemML Objectives

Despite the fact that femML began as a custom effort specific to the data exchange needs within the context of the activities of our group, the objectives employed to motivate the effort of the femML development were very specific and quite general:

- 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*. This effort minimally corresponds to defining: i. A set of XML Tags, ii. Relationships and constraints on these tags, and iii. 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 these tools.
- Develop a long-term framework for utilization of legacy RDBMS systems, 3D visualization viewer systems, and light-wait asynchronous processes architectures (i.e. agents), for achieving a truly distributed and transparent capability to utilize FEM techniques in highly functional, economical, and ubiquitous manner.

By the term "regardless of the taxonomic order" we mean the development of an XML dialect for FEM data exchange that can accommodate all, or most of the FE varieties, i.e. structured, unstructured, blocked, hierarchical, spectral, stochastic etc.

femML Document Type Definition (DTD)

The current state of affairs has been progressed into the development of a DTD that can definitely cover all of the taxonomic categories of FEM data, except the stochastic ones.

The strategy followed for developing femML's vocabulary of terms, relationships and constrains as well as the DTD that encapsulated them was a special application of the process described by the Unified Modeling Language (UML) [28] activity diagram given in figure 1.

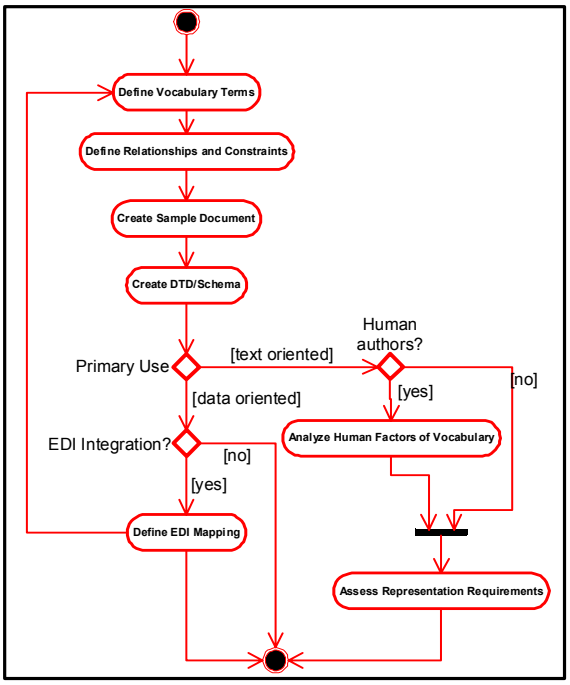

Figure 1 UML activity diagram representation for DTD/SCEMA development process.

This process in itself is a special case of the general process for developing an XML variant that although obvious, has been formally captured in a UML activity diagram in [29]. Clearly the pertinent path for our case implies engineering electronic data interchange (EEDI), and can only reach its terminal state if successful EEDI has been achieved. Otherwise it cycles through the feedback path on the left of the activity diagram that connects the "Define EEDI Mapping" node with "Define Vocabulary Terms" node.

The EditML [30] application was used to draft the sample document and its built-in capability for automatic DTD/SCHEMA generation was exploited to generate the initial stamp of the particular femML DTD and SCHEMA. The currently implemented DTD for femML appears as a UML class diagram shown in figure 2.

The ***femML*** node can be repeated multiple times within a single document to allow capturing the FEM data specification of multiple parts or domains.

The ***header*** element is there only to ensure there is a transport mechanism within the document for the meta-data.

Copyright © 2002 by ASME

These meta-data are implemented through meta-tags that encode information about the human author, the application generating the entire document, date of creation, project it belongs to and other non FEM specific data.

he ***materialSet*** element is responsible for carrying the material properties information associated with the FEM representation of the part or component encapsulated in the document at hand. ***NodeSet*** is the element that caries the nodal geometry information of the discretized component, while

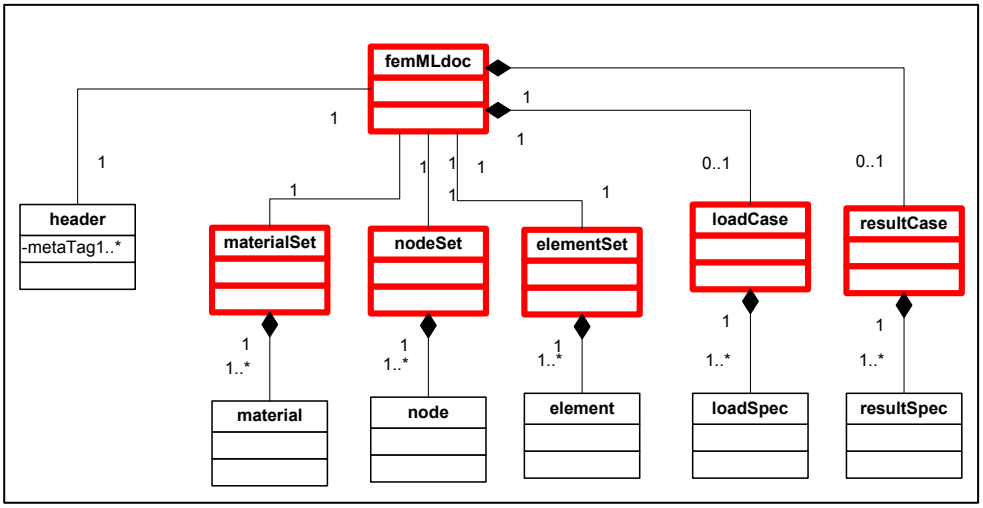

Figure 2 UML representation of the top two levels of femML DTD structure

elementSet carries the elemental information of the model. These last two elements/nodes are responsible for carrying the appearance information of the model from a 3D geometrical point view.

The ***loadCase*** element is responsible for carrying the loading or/and boundary conditions on a nodal basis. Finally, the first level of femML nodes is completed by the ***resultCase*** node that carries the nodal results data (i.e. displacements, stresses, strains, energies and any other scalar, vector or tensor component quantities associated with the nodes).

Each one of these nodes can have many individual children nodes to carry the specific data associated with each one of them like coordinate elements for each FEM node.

It is anticipated that the XSIL, X3D and MatML efforts will play a pivotal role in the development, testing and integration of femML. XSIL and X3D can serve as target translation languages for exploiting their visualization resources for 3D repreentation of FEM models. When MatML is completed but also even now that is under development, can be integrated through adaptation of the appropriate namespace to define material properties necessary for femML under the materialSet element by borrowing MatML's capability to describe material properties in place of the existing ***material*** node under ***materialSet***. Our interest in describing material properties for composite materials forms the basis of our current and future cooperation with the MatML working group.

STATION TO STATION DATA EXCHANGE VIA FEMML

General case of S2S implementation

Despite the benefits of extending the expressive power of XML with the dynamic data representation and manipulation capability of Java enabled applications, we have decided to focus onto the simplest of the approaches for exchanging FEM model data, by utilizing the Station-to-Station (S2S) approach that is built entirely on XML technology (see figure 3).

This decision does not preclude the future exploitation of the XML-to-Java (or vice versa) cooperative benefits in terms of dynamics, scalability, deployability and economy.

The S2S model assumes the existence of a source and a destination data-document and it is not different than the most generic of the business to business (B2B) models that dominated the Internet during the last few years.

The transformation can be either implemented trough a common XSLT processor or through a Java application that utilizes the parsed DOM equivalent of the source document structure and subsequently rewrite it to a new one that can then be converted to the target document.

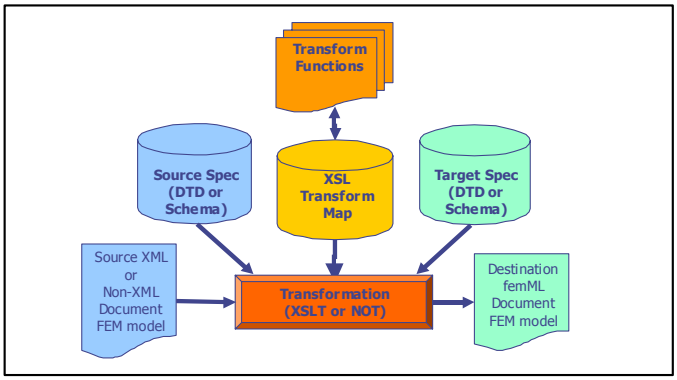

Figure 3 S2S data transformation architecture.

In either case, the transformation processor requires a transformation definition defined via a set of transform functions that may or may not implement templates, while at the same time it ensures that both source and target documents/files are valid according to their corresponding DTDs or Schemas. The transformation can be implemented in a multidirectional manner. Users with not extensive XML or/and Java experience can use of the shelve tools to construct the transformation engines as a byproduct of utilizing intuitive tools with simple graphical user interfaces (GUIs). An example

Copyright © 2002 by ASME

of such an application is BizTalk Mapper that belongs to BizTalk [31] set of tools that is a Microsoft led XML initiative

There are many other applications that allow the automation of document transformation design and implementation are that can be found either as stand alone applications or as parts of tool suites. Information about such tools can be found at the XML specific portals like xml.org, xml.com, etc.

ANSYS case of S2S implementation

When it comes to considering ANSYS as being one of the two applications that have to be used for exchanging FEM data, there are several factors to be considered regarding the existence of target and source file formats as well as means for linking to the ANSYS database.

The existence of the very powerful ANSYS Parametric Design Language (APDL) [32], as well as the ANSYS programmable features that allow linking of custom functionality into the main executable [33] through Fortran, C, C++ source code, present a very interesting variety of approaches towards parsing and generating input and output data files. The architecture that has been implemented can be seen in the figure 4. The doted line components have not been implemented yet. Instead of describing the additional components of this architecture to justify the ANSYS specialization of our general S2S approach we will rather describe the followed strategy because it describes the motivation behind the adaptation of this approach.

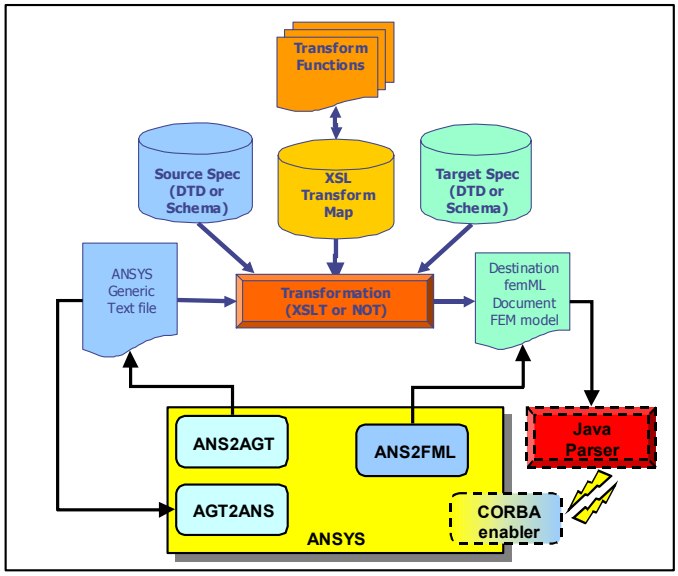

Figure 4 ANSYS based S2S data transformation architecture.

Thus the following steps can describe the strategy followed here:

1. Authored the ANS2AGT macro in APDL. This macro can be executed after one has a working model database complete with results. It reads the database and it creates an ASCII file that contains all the FEM pertinent data (geometry information, material definition, loading specification and corresponding results). This file has the extension ".AGT" to signify that it is an ANSYS Generic Text file. Other ANSYS file formats could be used (ANF, CBD) as well. The file structure is not important as the particular selection of represented data is. This selection has been designed such as when the data in this file are read back into ANSYS while the database is originally empty, the produced database contains complete information of a bottom-up built model (from nodes and elements), that can be solved and results can be plotted in ANSYS itself.
2. Authored the AGT2ANS macro in APDL. This macro implements the inverse functionality of ANS2AGT. Namely, it reads a .AGT ANSYS Generic Text file into the ANSYS database. The main reason for creating this macro was to establish an independent validation path for the integrity of the data exported by ANS2AGT.
3. Created the ANS2FML macro in APDL. This macro allows exporting the femML file that contains the data for an existing model in the ANSYS database.
4. Considered the strategy for reading femML files into ANSYS. The obvious solution would be to write a parser based on a C++ implementation of the DOM. A second strategy would be to use a Java based parser but since this would require to call Java from C++ that would require to develop a Java wrapper of ANSYS by utilizing the Java Native Interface (JNI). A third approach could be embedding a Java parser that takes the femML DOM information and passes it to the ANSYS database custom routines available as User Programmable Features (UPFs). This approach would require a mechanism for calling java bytecode from C++ that can certainly be implemented by using the JNI as well. A fourth strategy would be to exploit an application like Mathematica that can both talk to a Java parser of femML and the C++ extensions of ANSYS via the J-Link and Math-Link interfaces. This would make Mathematica a communications arbitrator between the parser and ANSYS. A fifth strategy would be to use a Java parser that can communicate with a custom version of ansys.exe that is enhanced by custom C++ routines for defining database entities. The communication will occur over the TCP/IP layer and will be implemented through one of the object oriented communication technologies like OMG's IDL on CORBA [34] objects. This last approach is something we are seriously considering pursuing given the time and programming resources will become available. None of these approaches could be implemented fast enough to be fully functional by the time the first femML DTD was ready.
5. Finally an indirect approach was used for this purpose. Utilized the S2S approach and initiated authoring of an XSL stylesheet template that would allow using XSLT processor transformation to transform the femML to its

Copyright © 2002 by ASME

AGT equivalent. The AGT2ANS module could be used to load the model into the ANSYS database.

6. An alternative way was to use a proprietary product called XMLjunction [35] that allows implementing bi-directional transformations between flat files and XML documents.

THE FUTURE OF FEMML

Issues to be resolved

The distance to be traveled for the development of a comprehensive femML standard is very long. Experience from other efforts has shown that on one hand no specification can end up becoming a standard if it does not have the support of industry, government, and academia. On the other hand experience has also shown that no standard is any good if it is not used by end-users in all of these sectors.

In addition to the economic and political factors the play a role in developing and adopting standards there are always technical reasons mostly associated with the derivable utility, that can make or brake a standard. The technical issues that we can foresee that will play an essential role in fem ML's adaptation and usage are the following:

Scale of Generality: Should we be thinking in extending femML to cover finite difference discretizations with all their idiosyncrasies, boundary element discretizations, hybrid discretization or even non-discrete models of continuous systems. For that matter should we be thinking of a femML or discrete model representation Markup Language (dmrML), or even a physical model behavior representation Markup Language (pmbrML)? It appears that the latter would be the most inclusive and general case. However, very ambitious goals may provide all kinds of reasons for not realizing these goals. This is an issue that a decision cannot be taken a priori before considering resources and support.

Separation between Appearance and Behavior: The current DTD implementation follows a strict FEM data file structural architecture. Information holding the geometry information of the 3D model is included along with loading, material and results specification. However, there are reasons for altering this situation. If we consider that there might be cases that the data that need to be exchanged are going to be based only on a particular subset of the original, then we have consider structuring the file in such a way that is easier to access and transfer data subsets of particular nature. The geometry model that is responsible for the appearance of the model, and the loading, material and results specifications that capture the behavior part of the model are two subsets of this type. The question at issue is under what conditions should we restructure the DTD to attach them under separate elements in order to facilitate transformation? In view of the existence of X3D and XSIL a direct mapping between their elements and the geometry elements of our DTD could be possible. The disadvantage of overextending such an approach may lead to a very verbose data file. However, since files like this are not intended to be human readable (although it actually is) but rather machine-readable. This issue will most likely be resolved from the need to integrate horizontally with other industries (i.e. entertainment industry) that may require CAx models.

Utilization/Leveraging of existing XML dialects: Particular element definitions (i.e. material definition) may already be defined from an already existing XML application (i.e. MatML). The namespace specification allows borrowing such constructs. The question is when we should be doing it when we should not. This issue can be resolved careful consideration of the semantic overlap and proximity between the intension of our application and the extension of the existing dialect's DTD.

Scene graph Structure for geometry: How much should we be aware and should we implement scene-graph internal representation architecture, that follows the lessons for other 3D representation methodologies such as Open Inventor [36], VRML and X3D[21]. What may determine a settlement to this issue may become a mute point when efficient ways to go in and out from the scene-graph representation become available while at the same time do not require users to spend time over steep knowledge curves. However, our group believes in leveraging existing technologies and lessons learned from their usage to make decisions about our contribution to the evolution of femML.

Composition and factoring isomorphism between data of FEM model and their femML expression: In the physical space structures can be thought as aggregates of parts and components. What allows us to synthesize a complex structure out of various parts is called composition of parts. When we decompose the aggregate structure to its parts we perform the operation of factoring. We can obviously think of both of these operations at the data representation level where femML documents can be the result of composing other ones (corresponding to part descriptions), or femML documents corresponding to part representations can be the result of factoring aggregate structures femML representations. The issue here is how much should we strive to establish and maintain a DTD/Schema architectures that preserves a on to one correspondence between the physical and data representation of the abilities to compose and decompose FEM representations.

Distribution of metadata: Should a document created as the composition of other documents that correspond to parts, components or substructures, carry the meta-data of the sources? If yes, should they also be composable or should they exist individually as a part of metadata nodes at lower levels of the DTD graph?

There are other generic issues that are applicable in the development of any XML variant and therefore valid for the case of femML as well.

The most typical of these issues is the dilemma of the choice between implementing a DTD and a Schema based strategy. There are pros and cons for both of them and for the moment we are focusing on the DTD although we have also created a schema for femML. However, we did not need to use the schema-based approach yet. The technical details of what

Copyright © 2002 by ASME

can and cannot be done with each one of them may be of no relevance here. The proliferation of particular tools and their economic impact to the users and developers combined with the usability and learn ability of associated tools, may turn out to be the major factor in the future that will determine which one is more practical for particular applications.

Another generic issue is the historical decomposition capability. Should we be able to factor the component data representations of a femML file based in previous versions of it? Effectively this would allow for a version control mechanism that is encoded inside the file. This falls well outside the particulars of our intentions regarding femML but it may very well be a seed for a useful debate on an old issue of the application development community brought into the context of the XML document management community.

Potential evolution of the femML DTD

To address all of the above-mentioned issues except the first one (scale of generality) we have already started modifying the femML DTD structure. A potential evolution of the femML DTD may start from the architecture presented in figure 7, where the top layers of the architecture have been captured as a UML class diagram.

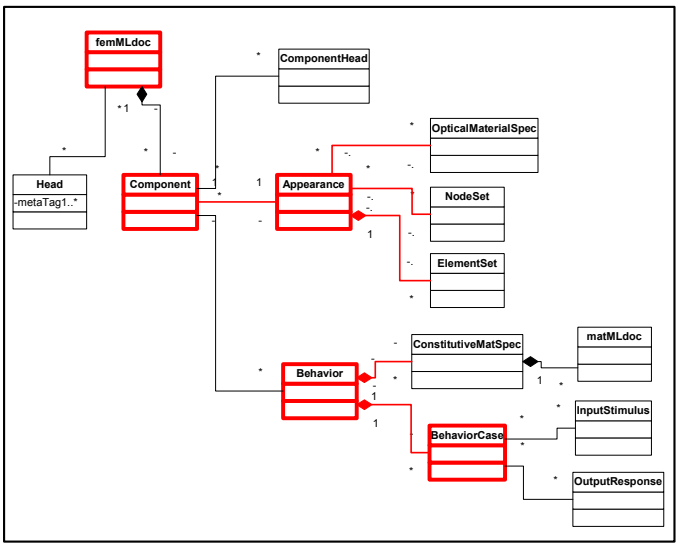

Figure 5 UML representation of the top two levels of an alternative femML DTD.

Here the issue of compositionality is addressed immediately by the implementation of the ***Component*** node. Effectively this allows the existence of many components with one or many separate domains in one file. The ***Head*** node at the same level is intended to capture the global metadata for this file with a substructure very similar to the one described before for the current incarnation of femML DTD. What is drastically different from the current architecture here is the fact that each component node has always only three children nodes. The ComponentHead node that contains the metadata associated with this particular component, and the ***Appearance*** and ***Behavior*** nodes. Effectively, these last two nodes manage to separate the geometry representation of the FEM data, from its behavior. This was done to ensure semantic independence between these two aspects of every discrete model of continuous structural system via FEM.

As expected the Appearance node has the usual children i.e. ***NodeSet*** and ***ElementSet***, but it also has the ***OpticalMaterialSpec*** node. The motivation of this node is based on the need to represent 3D geometries regardless of results data. It allows visualizing the 3D geometry of a model by assigning surface texturing, optical properties of the surface (reflectivity, emissivity etc) and in general parameters mostly known to the 3D rendering industry.

The behavior of a system in the continuous mechanics context depends on two major aspects.

The first aspect relates to the intrinsic material behavior of the structure that is usually captured by the constitutive equations applicable and the constants associated with them. To capture this aspect of the component behavior the node ***ConstitutiveMatSpec*** has been proposed as a child of the Behavior node.

The second aspect relates to the fact that the observed response of the system at the nodal or elemental levels depends on the stimulus on the system that is represented by the loading conditions. One has to have the stimulus-response representation of the system in order to claim an ability to represent behavior. For this reason ***BehaviorCase*** node has been introduced as another child of the Behavior node. Because the response of the system is specific to its stimulus and because we may have to deal with multiple cases of stimulus-response pairs this node can be repeated in the file.

To represent the stimulus for each one of the behavior cases we introduced the ***InputStimulus*** node that contains the loading and boundary conditions, and the ***OutputResponse*** node that contains the corresponding results from the FEM analysis. They are both children of the BehaviorCase node.

There is room for a lot o refinement and restructuring of these ideas and we hope that the interaction from the FEM community will help towards this end.

PROPOSED LONG TERM APPROACH AND CONCLUDING REMARKS

We feel that the in addition to the efforts on the custom development of femML by our group, a much more inclusive and robust effort would be to follow the approach similar to that used for developing MatML. The strategy of this approach has the following activities:

- Form, maintain and expand a 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
- Develop of the formal MatML document type definition or/and schema
- Development of a catalog of examples

Copyright © 2002 by ASME

- Application development and acceptance testing
- Utilize Open Source Development Network (OSDN) [37] resources like the "SourceForge" [38] development and deployment repository for DTD, SCHEMA, Examples, XSLTware, and custom format translator components.
- Iterate.
- DisseminationAt the present moment the working group is limited to NRL's CMS group representatives and to the International Science and Technology Outreach Society (ISTOS) members [39]. As soon we publish the results of the femML effort at cms.nrl.navy.mil/femML/ we plan to invite people from relevant mailing lists to participate and contribute in this effort.

REFERENCES

[01] J.G. Michopoulos, R. Badaliance, T. Chwastyk, L. Gause, P. Mast, C. Farhat and M. Lessoine, Coupled Multiphysics Simulation Of Composite Material Softening In A Virtual Wind Tunnel Environment, Proceedings of Sixth U.S. National Congress on Computational Mechanics, U.S. Association for Computational Mechanics, Dearborn MI, 1-3 August 2001, pp 521

[02] J.G. Michopoulos, P.W. Mast, R. Badaliance, I. Wolock, Health Monitoring of smart structures by the use of dissipated energy, ASME proc. 93 WAM on *Adaptive structures and material systems*, G.P. Carman/E. Garcia,eds., (ASME, AD-Vol. 35, 1993) 457-462.

[03] P. W. Mast, J.G. Michopoulos, R. Badaliance, H. Chaskelis, Dissipated energy as the means for health monitoring of smart structures, *SPIE Proc. Smart Structures and Materials 1994, Smart Sensing, Processing, and Instrumentation*, J. Sirkis, ed., (SPIE, Vol. 2191, 1994) 199-207.

[04] P. R. Factory, Health error prediction and sensor topology optimization on a smart pressure vessel, *Smart Structures and Materials 1995, Industrial and Commercial Applications of Smart Structures Technologies*, (SPIE, Vol. 2447, 1995), 155-166.

[05] V. M. Kern, Maintaining Semantics In The Integration Of Network Interoperable Product Data Models, Doctoral thesis submitted as a partial fulfillment of the requirements for a degree of 'Doutor em Engenharia de Produção' at the Universidade Federal de Santa Catarina, Brasil, Florianópolis, December of 1997.

[06] Y. Yang. "The STEP Integration Information Architecture". In: Law, K. (ed.): Engineering Data Management: Key to Success in a Global Market. Proceedings of the 1993 ASME International Computers in Engineering Conference and Exposition, San Diego-CA, pp. 39-47, 1993.

[07] P. Wilson, Information And/Or Data?, IEEE Computer Graphics & Applications 7, pp.
58-61, 1987.

[08] ISO 10303-1. Product Data Representation and Exchange - Part 1: Overview and Fundamental Principles. Committee Draft, September 15, 1992.

[09] ISO 10303-11. Product Data Representation and Exchange - Part 11: EXPRESS Language Reference Manual. Document TC184/SC4/WG5 N65(P2). ISO International Standard, November 1, 1994.

[10] D. Schenck and P. Wilson. Information Modeling: The EXPRESS Way. Oxford
University Press, New York, 388 pp., 1994.

[11] Recommended Practices for AP 209 – ME007.01.00, PDF document, can be accessed at http://pdesinc.aticorp.org/whatsnew/recprac209v1a.pdf, June 25, 1999.

[12] D. Tsichritzis and A. Klug (eds.) The ANSI/X3/SPARC DBMS Framework Report of
the Study Group on Database Management Systems, Information Systems 3, pp.173-191, 1978.

[13] T. Bray, Introduction to the Annotated XML Specification, REC-xml-19980210, http://www.xml.com/axml/testaxml.htm

[14] Router Solutions Inc., XML Strategy for CAE/CAD/CAM Integration, Press Release, can be accessed at http://www.rsi-inc.com/Company/Archive/xmlpress.html

[15] Unigraphics Solutions, Unigraphics Goes Interoperable, Press release, can be accessed at http://www.ugs.com/publications/articles/automfg_10/

[16] Unigraphics Solutions, Unigraphics Solutions Introduces Revolutionary Interoperability Standard for *c-Commerce*, Press release, can be accessed at http://www.hoops3d.com/about/press/00_07_18parasolidext.htm

[17] Autodesk, Autodesk Unveils Cross-Platform, Cross-Industry Open XML Strategy for Design Data and E-Commerce, Press release, can be accessed at, http://www3.autodesk.com/adsk/item/0,,279672-123112,00.html

[18] Bentley, Bluestone and Bentley to bring xml-driven applications to the $3.6 trillion a/e/c marketplace, Press release, can be accessed at http://www.bentley.com/news/00q1/bluestone.htm

[19] R. Williams, XSIL: Java/XML for Scientific Data, White Paper, June 27,200, Can be accesses at http://www.cacr.caltech.edu/projects/xsil/xsil_spec.pdf

[20] University of Auckland, Physiome Languages, Can be accessed at http://www.physiome.org.nz/

[21] Web 3D Consortium, X3D specification, Can be accessed at http://www.web3d.org/x3d/

[22] R. Carey and G. Bell, The Annotated VRML 2.0 Reference, Addison-Wesley Pub Co., 1997

[23] E. F. Begley, C.P. Sturrock, "MatML: XML for Material Property Data", ASM International's Advanced Materials & Processes (AM&P), November 2000, can be accessed at http://www.ceramics.nist.gov/matml/matml.htm

[24] C.P. Sturrock, E.F. Begley, J.G. Kaufman, ,MatML – Materials Markup Language Workshop Report, National Institute of Standards and Technology Report NISTIR6785, August 2001.

Copyright © 2002 by ASME

ftp://www.ceramics.nist.gov/matml/workshop.pdf

[25] P. Stern, F. Hemez, C. Farhat, RCFem: A Research Code for Finite Element Methodologies, User's Manual, Center of Aerospace Structures, University of Colorado, Boulder, September 1995.

[26] ANSYS, ANSYS Commands Reference, V 5.7 manual, can be accessed at http://www1.ansys.com/customer/content/documentation/57/Hlp_C_CH1.html

[27] HKS Inc., ABAQUS/Standard User's Manual (3 volumes),

[28] M. Fowler, K. Scott. *UML Distilled*, Second Edition. Boston: Addison-Wesley, 2000.

[29] D.Carlson. *Modeling XML Applications with UML: Practical E-Business Applications*. Boston: Addison-Wesley, 2001.

[30] NetBryx Technologies, EDITML XML editor, at http://www.editml.com.

[31] BizTalk.org can be accessed at http://www.biztalk.org

[32] ANSYS Inc., APDL Programmer's Guide, Publ. No. 001261, February 2000

[33] ANSYS Inc., Guide to ANSYS User Programmable Features, Publ. No. 001263 , February 2000.

[34] D. Harkey ,R. Orfali, Client/Server Programming With Java and CORBA , John Wiley & Sons, 1997.

[35] DataJunction Corporation, XMLjunction , can be accessed at http://www.xmljunction.net/

[36] J. Wernecke, The Inventor Mentor : Programming Object-Oriented 3d Graphics With Open Inventor, Release 2, Addison-Wesley Pub Co, (March 1994).

[37] Open Source Development Network, main site can be accessed at http://www.osdn.com/

[38] Open Source Development Network, Source Forge can be accessed at http://sourceforge.net/

[39] International Science and Technology Outreach Society (ISTOS) at http://www.istos.org.

Copyright © 2002 by ASME