TECHNOSOFT INC.

The Adaptive Modeling Language. A Technical Perspective

THE COMPANY

TechnoSoft Inc., founded in 1992, is a leading provider of object-oriented software for commercial and defense applications. TechnoSoft offers an advanced engineering framework, with an object oriented modeling paradigm called the Adaptive Modeling Language (AML), to enable the modeling and simulation of the entire product development cycle; integrating and automating product configuration and visualization, design and analysis, manufacturing and production planning, inspection and cost estimation.

TECHNICAL RATIONALE

AML provides a Knowledge Based Engineering (KBE) framework that captures knowledge from the modeled domain and creates parametric models with that knowledge. AML is "adaptive" in that it can be used to model a wide range of domains that have interacting components and constrained behavior between them. AML can be adapted to diverse engineering applications.

OBJECT-ORIENTED FRAMEWORK

AML's framework supports a single underlying object-oriented architecture. AML's object-oriented design paradigm provides two different types of relations. A class-subclass relation allows data abstraction, encapsulation, sharing of data structures and behavior, and polymorphism. Subclasses can be derived from any of the AML classes, or from user-defined classes. Multiple inheritance is available for classes. A class can be derived from an existing class, and new *properties* can be added, or *formulas* and *values* redefined for existing *properties*, as shown below:

(define-class *class-name*
  :inherit-from (*class-list*)
  :properties (*property-list)*
  :sub-objects (*subobject-list*)
)

The *class-list* is a list of classes to inherit from; the *property-list* is a list of object properties that are defined similar to the objects. The *subobject-list* is a list of objects directly located under the object being defined in the part-subpart hierarchical assembly. A part-subpart relation enables the creation of a tree structured unified part model where the children of any node of the tree represent sub-objects. Various aspects of the problem can be structured hierarchically according to the domain being modeled (Design, Analysis, Manufacturing, Costing & Inspection).

AML provides an expanded set of classes that support a wide spectrum of applications. Such classes are the basis of various modules supported within AML.

SYSTEM ARCHITECTURE

The AML modeling framework consists of several modules (sets of classes and methods), relating to the different knowledge domains, each focusing on different functionality. All the modules are written within the AML object-oriented architecture although they do communicate with external programs through the Virtual Layer Architecture. Additional modules can be defined and loaded into AML to 'adapt' the language for a specific purpose. Since AML is modular, only the necessary systems need to be loaded into AML. If a problem requires a modeling framework without graphics or geometry, only the kernel needs to be loaded for that application. Applications invoking different aspects of the system built in AML will utilize a common user interface to the system. User familiarization is required only with one interface irrespective of the applications. The Design, Analysis, Manufacturing, and Inspection modules all utilize a common AML interface.

THE AML KERNEL

The lowest level of the AML modeling paradigm provides the language constructs for defining classes, methods, and the constraint mechanism. All subsequent objects simply augment the language. The AML core system provides the ability to dynamically instantiate classes and methods and to add, edit, and delete objects and properties at runtime. The constraint mechanism, the part hierarchy, and other basic language constructs are also provided by AML.

DYNAMIC MODELS

Since design is inherently iterative and dynamic, AML is also dynamic in nature. Values and formulas of properties can be changed after the model is instantiated. AML also permits the addition and deletion of objects and properties after the model is instantiated. Methods can be defined against any AML class or one derived from an AML class. This lets the user modify behavior of classes according to the needs of the application. Polymorphism as well as `virtual functions' can be utilized.

PROPERTY OBJECTS

AML permits the definition of new property classes inheriting from existing ones such as property-object. Properties can be added to and methods written against a property class like any other class. In fact, a property-object is also derived from the AML object class. The property objects are the basis for accessing the virtual interfaces to communicate with external databases, foreign applications, geometric modelers, etc.

METHODS

Methods can be defined against any AML class or one derived from an AML class. This allows the user to modify behavior of classes according to the needs of the application. Polymorphism as well as `virtual functions' can be utilized.

UNIFIED PART MODEL

Various aspects of a problem can be detailed through a single unified model in AML. The design strategy and related engineering and production processes are captured within a single part model, represented by a hierarchy of objects. An example of such an application is the design automation and analysis of a combustion engine. A geometric design is created followed by the association of various physical attributes with the geometry. Then the attributes for a finite element model, and the strategy required for generating the mesh model along with various input files for the analysis solver, are maintained. AML enables the various aspects of the engineering processes to be stored in a single model in a structured fashion. Furthermore, knowledge for manufacturing, inspection, cost, and tooling can be incorporated into the same model for the automation of the manufacturing and inspection process plans. Feedback could be provided at various stages to different entities in the model. A complete user interface for the problem including input and output forms, menus, etc. can also be associated with the same part model that encompasses the various aspects of the application.

MODULAR, VIRTUAL LAYERS ARCHITECTURE

AML supports a modular underlying architecture consisting of a number of virtual layers that make it easy to interface to foreign applications or modules. Different modules within AML can be easily replaced or extended. A number of solid modelers are supported. An AML application can interface among different solid modelers without changes in the application code. Through the AML virtual layering interface, additional applications such as engineering analysis solvers, other solid/surface modelers, or any other engineering applications can be seamlessly integrated. The AML syntax is independent of the underlying foreign application since all communication is handled though the virtual layers. AML objects are fully portable and can seamlessly interface with other modelers or applications without changes to the user defined objects in the application source code.

Since AML supports a modular underlying architecture, additional modules required by the user can be easily integrated. Existing third party applications can be integrated with AML independent of the language or the operating system that they are developed in. Existing modules are written in C, C++, FORTRAN and LISP. The wide range of languages is proof that AML allows easy, open integration of additional modules.

The AML paradigm provides a common interface to a number of solid modelers in addition to different mesh generators and analysis solvers. This interface is implemented through a Modular and Virtual Layer Architecture, providing a common consistent interface to underlying foreign applications. Another advantage of the Virtual Layer Architecture is that a design can be created using different modeling engines. For example, the same geometric design could be created using the SHAPES™, ACIS™ or Parasolid™ solid modelers to compare accuracy and performance, or imported via STEP or IGES from various CAD systems. Similarly an analysis model could be exported to various solvers such as FLUENT™ or SPECTRUM™ without requiring changes in the analysis model.

CONSTRAINT MECHANISM

AML's underlying constraint mechanism supports demand driven and dependency backtracking behaviors. Demand driven refers to the fact that the value of a property is not calculated until it is demanded. Until a value is demanded, an internal flag refers to the property value as being unbound or the property being smashed. Several properties that effect a certain property can be modified, but the effected property does not need to be recalculated every time, only when it is finally needed. Dependency backtracking is the mechanism that actually propagates constraint changes throughout the part model. When a property is modified, all the properties in its effect list are smashed. When a property is smashed, it further smashes all properties in its effect list, propagating the change by notifying entities that they need to be recalculated when demanded next.

MODEL TREE SEARCH

AML employs a unique mechanism for associating *properties* and *objects* within *formulas* and *values* referred to as *the* referencing. The *the* mechanism provides the means for querying the model for properties and objects as well as establishing constraint relationships. Additional functionality and *querying* mechanisms are provided to enable extended definition of *constraints* and *dependencies.* The *select-object* function provides a means of selecting objects over the entire model, or for a particular branch of the tree.

PARAMETRIC DESIGN

The constraint mechanism coupled with the tree-search provides the parametric modeling environment. Several properties of the model can be changed and then the results can be computed, which may result in new geometry or different outputs. A "what-if" scenario can be achieved without the user having to manually notify entities of change. The change-value and change-formula methods assist in modification of properties.

EVENT TRIGGERS

Although a demand driven paradigm suffices for several applications, situations arise that require notification of entities that may be outside the domain of dependencies and certain actions need to occur at the moment an event takes place. The system defines classes that can be inherited into objects to define actions on creation, deletion, change and smashing of properties or objects. An example of the use of an event trigger is in the way AML handles the freeing of external pointers on notification from AML events. When a property that holds a pointer external to AML is smashed, a trigger is activated that invokes the system call to destroy the pointer and unbound the property. This cannot be done using dependencies alone since the pointer is external to the constraint network. This mechanism is unique to the AML functionality of the Virtual Interface that independently manages automatic allocation of memory to external applications.

GEOMETRY

AML supports advanced parametric solid and surfacing capabilities including “web” geometry with complete topology access and mixed Boolean operations. The advanced geometry module supports seamless virtual interfaces to Shapes™, Parasolid™ and ACIS™. Additional solid/surface modeling kernels could be easily integrated via the AML Virtual Layer Interface (VGL). AML’s VGL enables the user to preserve the AML application code while extending its compatibility to other applications and modelers.

The various applications of AML including design automation, layout and configuration, manufacturing planning, and finite element modeling and analysis, have different geometric requirements. These individual requirements are satisfied through the capability of augmenting the part model for different representations to satisfy various demands of the different applications. These different representations are manipulated through a unified part model. AML presents a number of objects/classes for modeling simple primitives in addition to complex geometrical operations. Complex Boolean operations for mixed-dimensional solid, surface, and/or wireframe representations, incorporating non-manifold topology, are also supported. Additional objects for advanced modeling of free-form surfaces (NURBS, Beziers, etc.) also exist. AML supports a STEP and IGES interface to external CAD systems.

GEOMETRIC REASONING

AML supports geometric reasoning for process planning automation to integrate the part design with the manufacturing, inspection, or analysis plans for simultaneous engineering. Various queries and methods to enhance and modify the part geometry as required by the manufacturing, analysis, and inspection processes, are also supported by AML. For example, queries about a distributed set of points on a free form surface, along with their normals and connected path could be dynamically queried and presented in a separate object. This object could also be used for an inspection plan.

PROCESS PLANNING

AML provides a suite of objects, which are fully integrated, supporting the general requirements of manufacturing and inspection process planning automation. Present manufacturing capabilities includes machining, supporting milling and hole making process planning as well as cost estimation. Also a unique tool path planner for NC is fully integrated within AML. This dynamic tool path generation supports up to five axis NC tool path planning. The automated path planner supports **geometrical reasoning** capabilities which are suitable for applications in spot welding, arc welding, spray painting, water jet cutting, Eddy Current inspection, and other applications that require the position and motion control of a tool/probe moving on or about complex shapes and surfaces.

Capabilities to control multiple path offsets, distance from, and orientation between the tool and the surface are also automated. The uniqueness of the integrated and dynamic link between the geometrical modeler and the path planner comes from AML's single underlying object-oriented architecture.

GRAPHICS

Through its virtual layer capabilities, AML supports advanced visualization techniques for manipulation and display of the geometrical objects. These capabilities include limited functionality for post processing and visualizations, such as color mapping using various grouping techniques. AML supports various objects for creating dynamic charts such as bar charts, curve fitting charts, etc.

PARAMETRIC FEATURE BASED DESIGN

AML provides a unique interactive design environment. It is parametric, constraint driven, free form feature based, and has solid and surface modeling capabilities. Geometric as well as non-geometric features can be modeled. The system enables easy referencing and parametric association for feature properties that could be linked to external processes as a part of the Virtual Layer interface.

USER INTERFACE

AML provides a complete set of interface classes including forms, buttons, radio boxes, check boxes, input forms, and pop up menus supported on Unix/Motif and Windows NT/95. Since the user interface model is represented using the same knowledge representation system as the applications, it too is dynamic in nature, and the attributes of the user interface entities, (color/size of buttons), can be altered dynamically.

ATTRIBUTE TAGGING AND PROPAGATION

AML provides a unique mechanism for facilitating association of information with entities in a geometric model. Attribute Tagging and Propagation is being utilized for facilitating association of engineering processes information with entities in a geometric model. This information typically needs to be conveyed to downstream processes such as manufacturing, inspection, meshing or analysis. Typically, when a geometric model is constructed, several stages of construction geometry are required. These construction entities are then booleaned, transformed, swept, revolved, etc., to create the final model. In a parametric modeling environment, reconfiguring the model involves the modification of parameters at the construction level and the regeneration of the geometric model. All supplementary information would need to be reconveyed to the final geometry as well as downstream processes every time the model is reconfigured. This could be a very tedious task requiring major interaction with the user and delays in the engineering cycle. AML's unique parametric modeling paradigm provides the capability to propagate attributes through geometric operations thereby automating the procedure of geometrical enhancement of the final geometry to extract the required data for complete automation of the finite element modeling and analysis processes as well as any other engineering processes.

First, using Attribute Tagging, the supplementary information is associated with construction configurable geometry. Next, every downstream operation, including final design, analysis, and manufacturing has the information passed on through Attribute Propagation. Attribute Propagation ensures that every operation on the geometry, including reads and writes, Booleans, sweeps, etc., propagate the attributes through the operation. As a result, when the model is reconfigured, i.e., upstream design entities are modified in geometry or other properties, the Attribute Propagation mechanism ensures that supplementary information is passed downstream automatically. The Attribute Tagging and Propagation mechanism is integrated with the demand driven and backward propagation mechanism using Event Property objects.

MESH GENERATION

The Automatic Mesh Generation system is an AML module that allows tight integration of various mesh generation and analysis applications. AML provides a Virtual Interface to support various third party mesh generators. The system permits the selection of geometry to mesh, tagging the vertices, edges and faces of geometry for selective refinement of the mesh, and meshing the geometry by calling the external mesh generator. It provides objects for meshing as well as a user interface. It also provides various objects and user interfaces for visualizing the mesh and querying the mesh database created by the mesh generator.

ANALYSIS MODELING

The Finite Element Analysis (FEA) Modeling module will enable the definition of an analysis problem by defining regions of interest, material models, boundary conditions, solution strategies, and other requirements for analyzing various problems utilizing a Mesh Generator and a Finite Element Solver. The various entities of interest are modeled as AML classes that can be utilized to instance a complete FEA problem model. The problem can be associated with the geometric objects as well as the mesh. The system generates several files that the solver can read and execute to generate results. This system can be extended to provide a Virtual Solver Layer that can talk to various FEA solvers.

COMPACT AND PORTABLE ARCHITECTURE

The complete AML paradigm, including the various modules, requires less than 30MB of disk space. AML requires 100MB of swap space with 64MB of RAM recommended (32MB RAM could meet the requirements of various applications). The relatively small requirements of swap and memory size reflect AML's object oriented unique architecture. AML is supported on UNIX platforms and Intel based PC's running WINDOWS NT, 98, or 95. The compact size of the AML system is proof of its innovative advanced architecture when compared to the requirements for storage, swap size, and electronic memory of other competitive products.

SUMMARY

The AML paradigm provides a versatile, parametric modeling environment supporting a unified part model for integrated design and process automation. The AML object-oriented modeling framework allows the user to develop applications using dynamic objects for composing adaptive models that can be tailored to various engineering requirements at runtime. The dynamic environment is suited to simulating “what-if” scenarios and iterative modeling environments. AML offers a flexible modeling environment that can be used for a wide spectrum of engineering problems requiring the integration of various engineering disciplines all supported within an adaptive object oriented part model. AML enables the abstraction of the modeled domain into a set of interacting entities which can be applied to problems requiring a high degree of visualization and complex geometric operations for integrated design and process automation.

AML is a revolutionary modeling framework that shortens the design-to-manufacturing cycle, resulting in rapid part production with lower cost. Complex parts and detailed process plans for manufacturing, analysis, and inspection are concurrently designed and developed in a fraction of time compared to current methods. Improved quality and efficiency is realized by producing intelligent, error-free designs.

AML PROVIDES MANY ADVANTAGES:

- A true concurrent engineering environment for integrated product-process design
- Constraint driven, parametric, feature-based design environment
- Industry standard support including IGES, STEP, CORBA (in release 4.0), and JAVA
- Platform independent, fully portable across UNIX and NT
- Configurable and customizable user interface
- Single Syntax for all operations
- A single underlying object-oriented database, with open access
- Virtual links to multiple geometry engines, mesh generators, OS systems, UI's, and DBMS
- Geometrical reasoning for process and inspection planning automation
- Provides the framework to conduct multi-disciplinary trade studies
- Automatic Mesh Generation to speed up model analysis preparation
- Feature reduction and suppression for design evaluation
- A framework to capture the design intent
- Activity Based Costing (ABC) to dynamically assess the impact of design or material change

TechnoSoft Inc. is a team of engineering software experts dedicated to the automation of the engineering and product design process, reducing the design costs and time to market. Our knowledge engineers will apply our advanced technology software to configure or customize a system to meet your requirements. We provide exceptional service, training, and support, and are focused solely on delivering the best and most productive solutions for our customers.

TECHNOSOFT INC.
11180 Reed Hartman Highway
Cincinnati, OH 45242
Phone: (513) 985-9877
Fax: (513) 985-0522
Email: info@technosoft.com