A FEDERATED INTELLIGENT PRODUCT ENVIRONMENT

Peter J. Röhl*, Raymond M. Kolonay†, Rohinton K. Irani, Michael Sobolewski, Kevin Kao
General Electric Corporate Research and Development
Schenectady, NY 12301

Michael W. Bailey†
GE Aircraft Engines
Cincinnati, OH 45215

Abstract

The concept of a federation of distributed devices on a network which enter the federation through a process of "discover" and "join", by which they register with a service request broker and publish the services which they perform is applied to engineering software tools. A highly flexible computer architecture is developed, leveraging emerging web technologies like Sun Microsystems' Jini™, RMI, JavaSpaces, in which engineering software tools like CAD, CAE, PDM, optimization, cost modeling, etc. act as distributed service providers and service requestors. The individual services communicate via so-called context models, which are abstractions of the master model data of a particular product. A human user interacts with the framework through a thin client like a web browser from anywhere in the world, where proper security measures to prevent unauthorized access to proprietary data is maintained. The paradigm of the CAD Master Model is extended with the introduction of the Intelligent Master Model (IMM), which, in addition to the *what*, captures the *why* and *how* of a design through the use of knowledge-based engineering tools. An initial example, the mechanical analysis of a turbine engine blade, is implemented.

Introduction

Turbine engine development is a highly coupled multidisciplinary process. In a market with ever increasing demands in terms of life cycle cost, environmental aspects (noise, emissions, and fuel consumption), and performance, the availability of accurate analytical tools during the design process is a given and ceases to be a discriminator between the various competitors[1,2]. It is, therefore, the application of these tools and their automated interaction in a robust computational environment, which may decide over success or failure of a specific project through reduction of design cycle time and avoidance of costly rework because of availability of high-fidelity information earlier in the design process. At the same time, especially in a multi-national company, design increasingly takes place at spatially distributed locations, potentially all over the world, where all participants in the design process need constant real-time access to all relevant up-to-date product information. In light of these challenges, GE has teamed with Engineous Software, BFGoodrich, Parker Hannifin, Ohio Aerospace Institute, and Ohio and Stanford Universites in a four-year effort to develop a "Federated Intelligent Product EnviRonment" (FIPER) under the sponsorship of the National Institute for Standards and Technology-Advanced Technology Program (NIST-ATP™), see Figure 1. FIPER strives to "drastically reduce design cycle time, and time-to-market by intelligently automating elements of the design process in a linked, associative environment, thereby providing true concurrency between design and manufacturing. This will enable distributed design of robust and optimized products within an advanced integrated web-based environment"[3].

The Intelligent Master Model

FIPER draws extensively on GE Aircraft Engines' Common Geometry Strategy, the Linked Model Environment (LME) and top-down Product Control Structure (PCS)[4] (Figure 2) using Unigraphics[5] (UG) WAVE functionality, but tries to extend these efforts with the capture of designer's knowledge in Knowledge Based Engineering (KBE) systems to create the "Intelligent Master Model" (IMM). In the following paragraphs, we will give a brief overview over the Common Geometry Strategy and the terms introduced above.

---

* Member AIAA
† Senior Member, AIAA

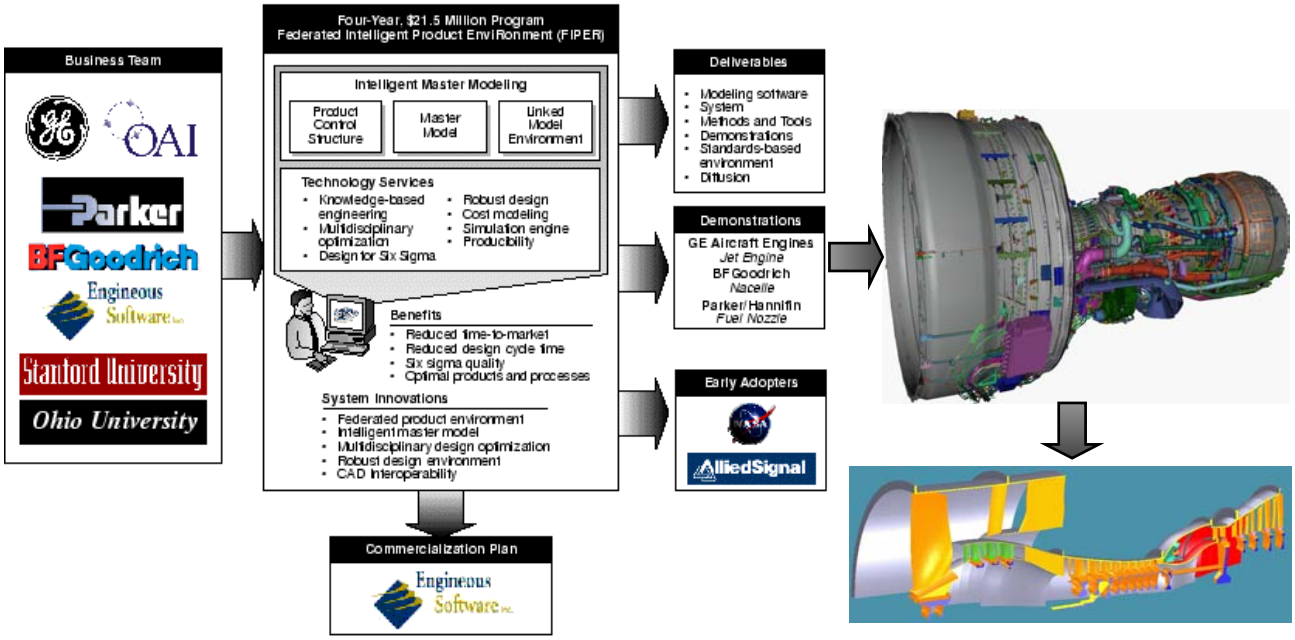

Figure 1: The FIPER Project

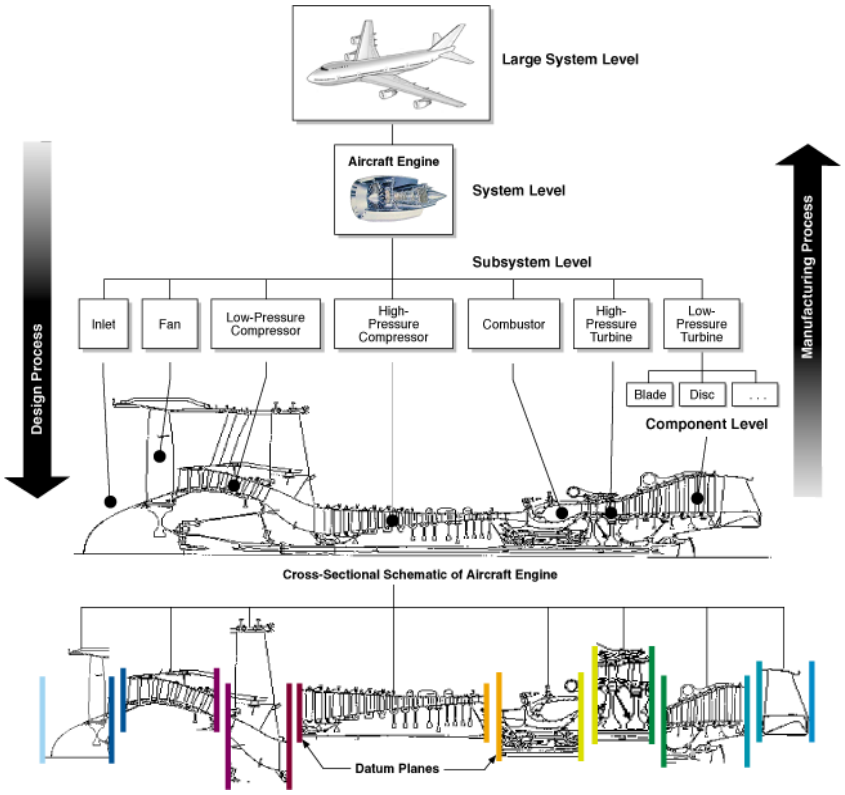

Figure 2: Product Control Structure

GEAE Common Geometry Strategy

The GEAE Common Geometry initiative started four years ago as a logical extension to Productivity Tools which had been under development since the early nineties. It was realized that merely automating what was essentially a serial process had limitations and a fundamental paradigm shift was required. Bottom-up design may be optimal from a part perspective but does not necessarily lead to optimal system design. As initially conceived, the GEAE

Common Geometry strategy objective was to make a single geometric representation common to all product creation activities from product concept through preliminary and detail design to manufacturing and services. However, to fully exploit the concept, knowledge has to be fused with feature-based parametric CAD (Figure 3), an environment linking CAD to engineering analysis, the LME (Figure 4) and a PCS to render it an Intelligent Master Model. This permits a top-down approach to design which permits system level requirements to flow down to drive the design. The IMM is a major enhancement to the master model concept, elevating the functionality of today's CAD systems to a new level.

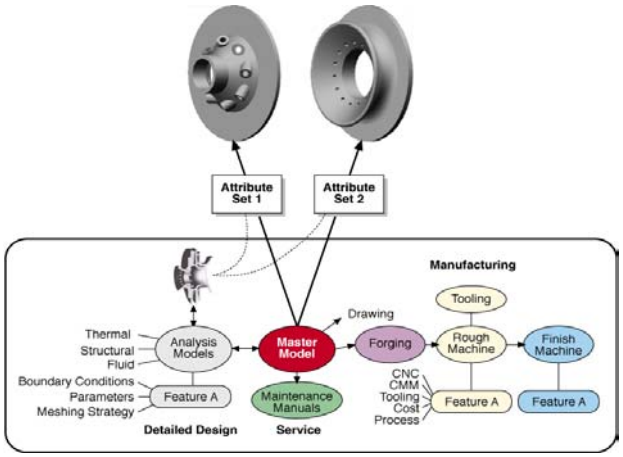

Figure 3: The Master Model Supports Feature-Based Modeling for Design and Manufacturing

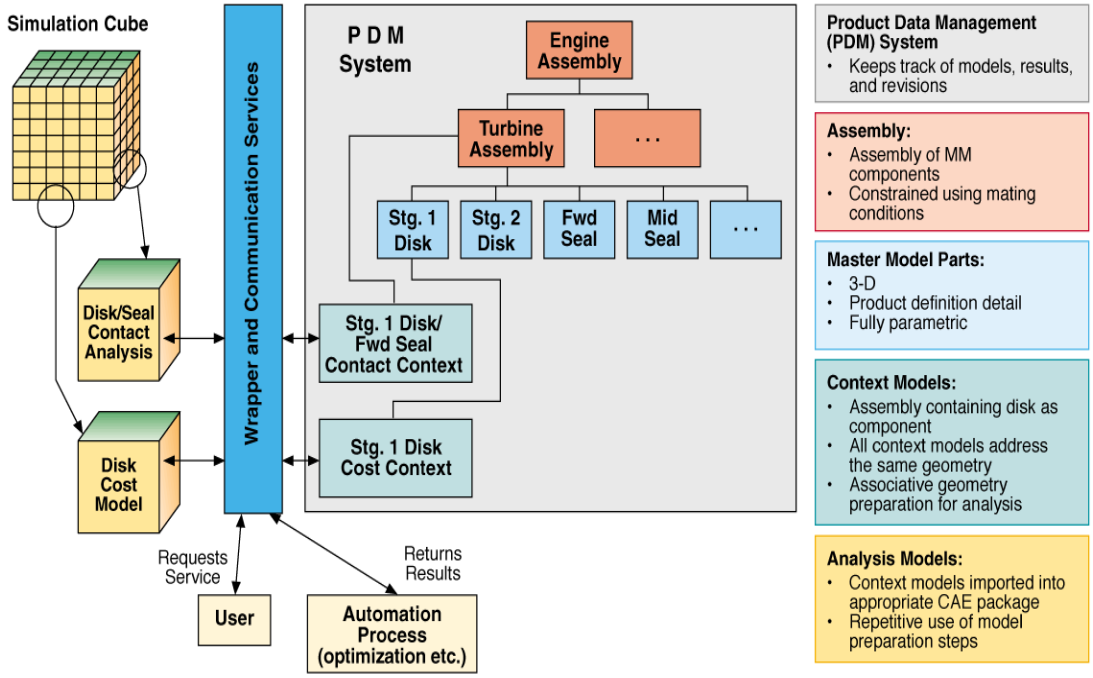

Figure 4: Linked Model Environment

The PCS allows top-down control of the design. It enables the lead engineer to lay out the overall system configuration and control changes in a top-down fashion. It facilitates what-if analysis at the conceptual, preliminary, and detailed design levels by allowing the designer to make parametric changes in the overall system layout and space allocation to evaluate one configuration versus another. Common Geometry refers to the notion that all disciplines involved in the design and manufacturing process have access to and use the same (evolving) geometric representation of the product. Realizing that different disciplinary engineering design and analysis tools require geometry at different levels of detail, the concept of a "context model" was introduced. The context model represents a disciplinary context-specific, yet fully associative, "view" of the master model geometry. Feature suppression is extensively used in context models. For example, a bolt hole, which is important for the stress analyst, may not be required for a thermal analysis and therefore be suppressed in the thermal context model. Another context or "view" of the bolt hole are the manufacturing processes and cost to produce it. These context models are then linked to the respective

disciplinary analysis tools, e.g. FEA, CFD, cost, producibility, etc, in the LME , see Figure 4.

System Level Layout with Integrated Design (SOLID)

The pilot projects to evaluate IMM functionality were described in Reference 4. To demonstrate the top-down design approach, a compressor was built using the feature based parametric CAD and WAVE functionality in Unigraphics. The following year this was extended to a full core engine comprising a compressor, combustor and high pressure turbine. The resulting model was called SOLID (System Level Layout with Integrated Design). Several lessons were learned during the pilot which were incorporated not only in the SOLID core but in Unigraphics enhanced functionality. The SOLID core is shown in Figure 5.

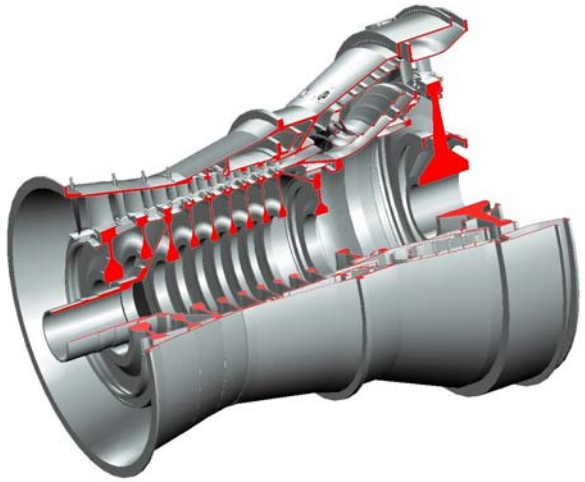

Figure 5: Complete Engine Core Model

Knowledge Based Engineering

KBE is a technology that allows an engineer to create a product model based on rules that capture the methodology used to design, configure, and assemble products. KBE facilitates the capture of the intent behind the product design by representing the *why* and *how,* in addition to the *what* of a design, see Figure 6. The knowledge captured could include everything from high-level, non-geometric engineering rules, manufacturing constraints, dependencies and relationships to parametric geometry definition. The geometric description is only one view of the information associated with the total product model. Links can also be established to standard parts catalogs, material databases, analysis tools, empirical knowledge, and design handbooks. Effectively, one can house, and ultimately archive, corporate design practices as well as design and manufacturing engineers' expertise which can then be used by non-experts in a consistent manner to produce correct-first-time designs. Once the product model has been created, it can be used to rapidly create a new instance of the design when the product specifications change. In addition, various outputs, including analysis context models, etc. would be automatically created.

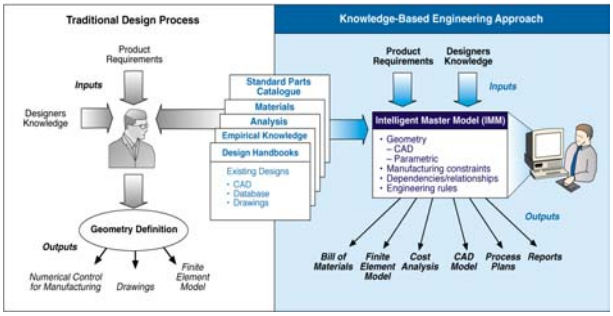

Figure 6: Knowledge-Based Engineering Extends the Master Model Concept

In the FIPER environment KBE is being used to intelligently modify the PCS, drive changes to parameters that define cross-sections and features and thereby intelligently scale a complete aircraft engine or components of the engine. To accomplish this, the approach being used within the Unigraphics system is to imbed the KBE language Intent[TM,6] to drive generative and variational design. While variational design creates a new design by intelligently scaling an existing design, generative design creates a new design based on a set of rules without the use of existing geometry. Rules management is also being addressed by incorporating them in a Product Data Management (PDM) system so that they are well documented, categorized and easily searchable.

Another use of KBE that is being pursued is for the formulation and execution of the MultiDisciplinary Optimization (MDO) problem. Here knowledge will be used to guide the decomposition of the overall optimization problem into smaller, more manageable, sub-problems, and to integrate the solutions of the sub-problems into an overall system level design.

The initial approach to KBE was the encapsulation of rules about the product in the form of the XESS spreadsheet functionality contained within Unigraphics. These spreadsheets are linked to the geometry such that design rules and practices are parameterized to drive geometry. In addition external analysis codes such as those used for engine disk design can be executed. Thus an increase in airflow through the compressor would initiate an aerodynamic resizing of blades and vanes, resulting in a blade and platform resizing combined with disk redesign. Upon initiation of the UG/WAVE update, the whole compressor would rubber band to accommodate the increased airflow.

It was realized there that were limitations to the utilization of spreadsheets to capture the knowledge required to accomplish intelligent scaling of the engine core. GEAE evaluated several KBE packages to find the desired functionality. In the meantime Unigraphics Solutions entered into an agreement with Heide Corporation to integrate their Intent™ software into Unigraphics as "UG Knowledge Fusion". The reason for this is that for complex products the number of rules gets large very quickly and consequently difficult to manage. There are two types of KBE rules, Generative and Checking. Generative rules would for example change the number of stages in the compressor from 9 to 7 stages whereas the Checking rules would check that the disk bore stress and burst margin conform to design practices and run the appropriate codes to validate this requirement.

FIPER Architecture

Fundamental to the FIPER project is its web-based distributed software architecture. FIPER federates processes, tools, methods, documents, or knowledge bases and data into a dynamic, distributed Intelligent Master Model with its underlying services. Some services are generic (for example optimization algorithms, or knowledge-based systems), and thus, are not associated with a particular IMM context but are globally available within FIPER. Members of a federation agree on basic notions of administration, identification, and policy. The resulting federation provides the simplicity of access, ease of administration and support for sharing services provided by a large monolithic system, while retaining the flexibility, and control provided by a plug-and-play environment.

FIPER supports three centricities and deploys three neutralities. FIPER's three centricities are network centricity, service centricity, and web centricity. FIPER is composed of various service providers; any of these can come and go and the system can respond to changes in its environment in a reliable way (network centricity). The services connected to FIPER discover each other and cooperate in a distributed environment (service centricity). Users can request to use multiple services and check the status of their submissions in different locations through HTTP portal with thin web clients (web centricity).

The three neutralities FIPER deploys are location neutrality, protocol neutrality, and implementation neutrality, Figure 7. Services need not be co-located; they are discovered and joined, which simplifies management of the entire software environment (location neutrality). In addition, the way clients communicate with a service provider is not essential. A service proxy can use any protocol, for example, Remote Method Invocation (RMI), IIOP or even a plain socket communication. Clients are not aware of what protocols are used and where the implementations reside (protocol neutrality). Furthermore, the clients who use the FIPER services do not need to know what languages are used and how a service is implemented (implementation neutrality). In all, FIPER provides accessibility through web centric architecture, self-manageability using federated services, scalability via network centricity, and adaptability with the power of plugging-and-playing capability.

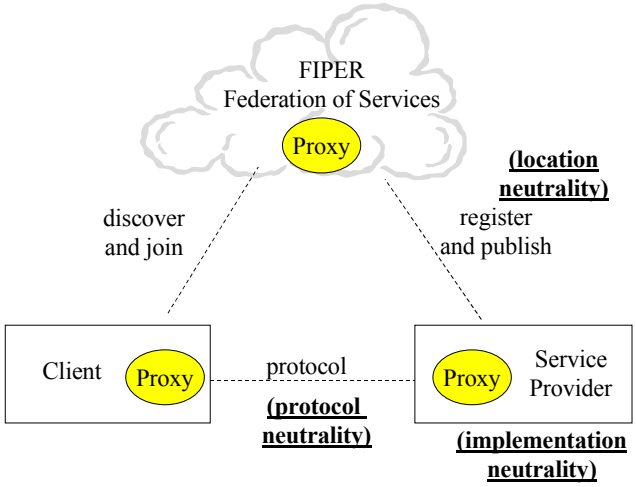

Figure 7: FIPER's Three Neutralities

FIPER's federated architecture is based on Java and Sun's emerging Jini™ software system (Figure 8). The overall goal is to turn the network into a flexible, easily administered tool on which resources can be found by humans or computational clients. The Jini™ system consists of:

1. A set of components that provides the infrastructure for federating services in a distributed environment
2. A programming model that supports the production of reliable distributed environment
3. The functionality to register services and resolve service requests

Java and the emerging Jini™ technology are at the heart of this technology. Services are found and resolved through a "lookup" service (Figure 8). New services are added to the look-up service by a process called discovery and join. When plugged into the environment, the service first uses a discovery protocol to locate an appropriate lookup service and then joins, or registers, with the lookup service. Services can communicate with any other generic service in the entire federated product space. In the case of FIPER, this is achieved by an IMM context, user, or service posting a need which is resolved by a lookup service.

The lookup service connects the requesting entity to an entity that has the functionality to supply the service. Figure 8 illustrates this in a given space with four services; CAD, KBE, Optimization and Robust Design, and the Simulation Engine. Each service provider must be Java wrapped in order to join the federation, but it can have its own framework of execution. A service could be based on RMI, CORBA, Java Native Interface (JNI), Microsoft COM/DCOM, or even simple socket connection.

Clients define and submit their jobs via web browsers. A FIPER service manager then dispatches each job into tasks. These tasks can be executed sequentially, in parallel, or combination of both in the FIPER environment, depending on their input/output data dependency. If a parallel strategy is chosen, tasks are dropped into spaces (by using JavaSpaces, for example) for distributed computation. Each service provider agent, if present, picks up appropriate tasks and generates results back to the spaces. On the other hand, FIPER provides a service catalog for direct task execution. The catalog discovers all FIPER services and maintains a list of currently active ones. Appropriate registered service providers will then be selected to perform tasks. Finally, a service manager collects all the outputs and informs the FIPER notification manager about the outcome. The results are presented to the clients when they request.

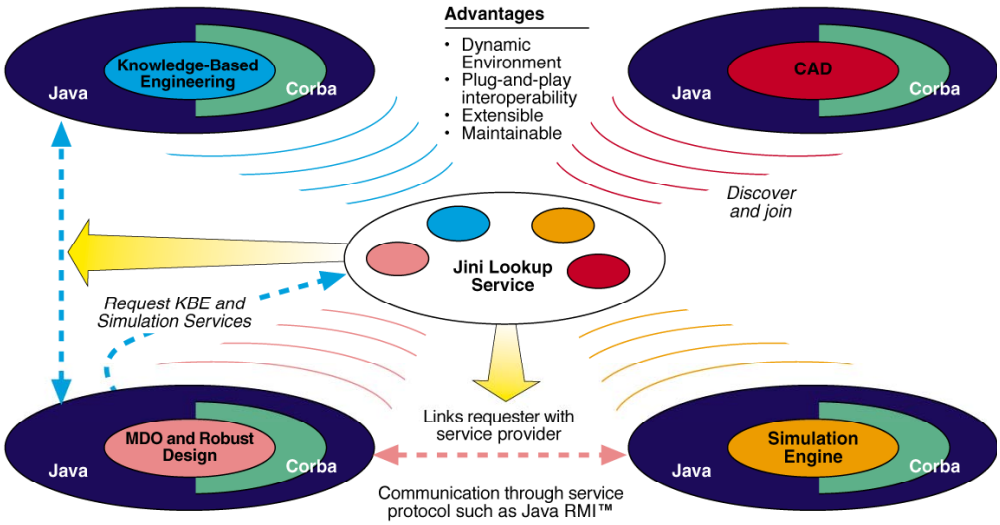

Figure 8: Web-Based FIPER Architecture

This environment promotes concurrency and ensures that current and consistent information is employed throughout the distributed system. The dynamic nature of this approach allows services to be added (for example, support for an additional CAD system) or withdrawn from a federation at any time. The federated environment enables transparent communication between the globally distributed IMM contexts and services, thus providing the means to solve distributed complex tasks such as intelligent scaling of entire systems (e.g., an aircraft engine) and MDO problems.

Engineering Services

The basic premise of FIPER is that everything is on the network and everything on the network is viewed as a service. With this in mind FIPER can contain any "service" needed to support a product throughout its life cycle. For example, services for customer requirements, design, manufacture, sales, distribution, maintenance, and disposal can all be supported by FIPER. For the purpose of the NIST project FIPER will focus on the services necessary for the design and manufacture of a product. Specifically the domains of Design for Six Sigma (DFSS)/MDO, CAD/KBE, Engineering Analysis & Sensitivities, Pre/Post processing, and Data Repositories will be addressed. This is illustrated in Figure 9. For an initial application the services required for the mechanical analysis of an aircraft turbine component will be developed. These services consist of associative parametric solid geometry modeling, meshing, boundary/initial condition application, and analysis solution. Although all of these services could potentially be provided by one monolithic system, this is rarely the case in today's design environment. Tools for these different services are selected based on many varying criteria ranging from "best in class" to corporate mandate. Figure 10 shows the relationship of these services. Although a very simple case, it can be used to demonstrate the FIPER "service" paradigm in a distributed heterogeneous computing environment.

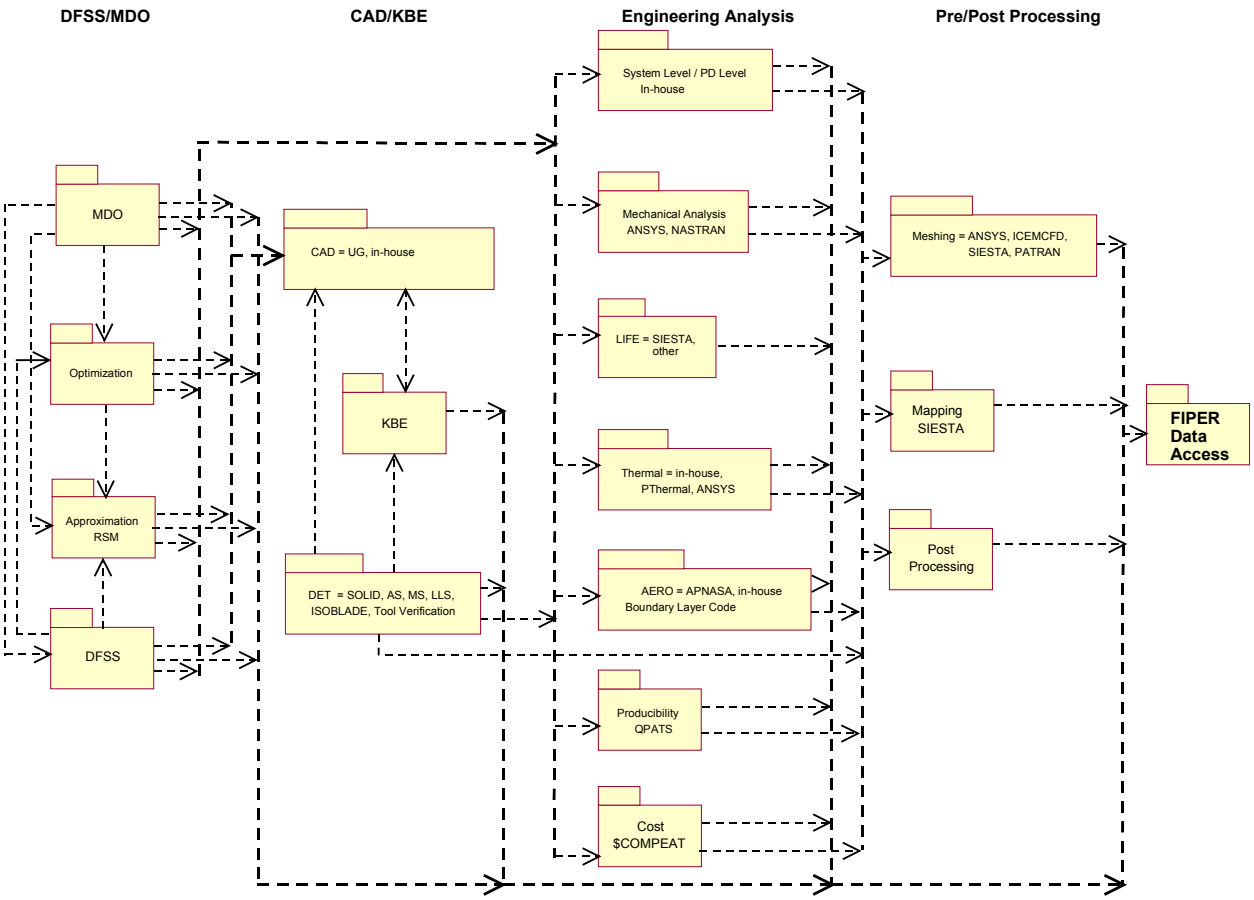

Figure 9: Services Package Diagram

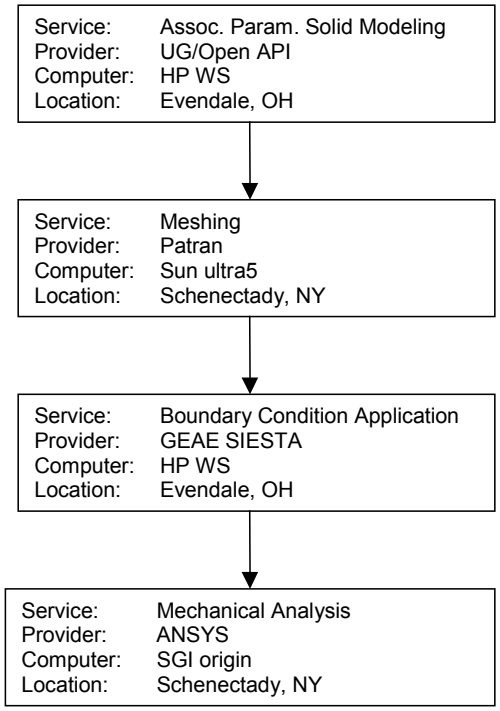

Figure 10: Analysis Flow Chart

Parametric Solid Geometry Service

This service generates the necessary solid geometry that will be meshed and analyzed. In this example the provider for this is a Unigraphics User Function (UFUNC) program that requires an initial seed part and a parametric data file as input. With these inputs the program constructs a three-dimensional solid of the component and associates attributes or "tags" with various geometric entities (surfaces, edges, etc.). These tags will be identifiable and used by other services in the system such as meshing and boundary condition application. The UG UFUNC program is "wrapped" as a FIPER service and deployed on a Unix workstation in a remote location.

Meshing Service

The meshing service discretizes a given component. As input, it requires a geometric entity and some information describing a strategy for meshing the supplied geometry. The meshing strategy contains information such as mesh seeding parameters and element types. These attributes are mapped to the geometry via the associated geometric tags. For the present case MSC PATRAN[7] is the service provider. A

wrapper is written for PATRAN that takes the meshing strategy information and generates PATRAN PCL. With the PCL the wrapper invokes PATRAN which in turn produces a mesh for the given geometry. The service exports the meshed geometry in the form of a PATRAN neutral file. As shown in Figure 10, the meshing service resides on a local Unix work station. It is worthwhile to note that all the geometric tags that were created in the solid geometry service are transferred onto the descritized geometry. Thus, the tags can continue to be used to identify particular attributes of the model. These will be available to other services.

Boundary Condition Application

The boundary condition service applies a set of boundary conditions to a given meshed geometry or group of geometries. It requires as input a descritized geometry (PATRAN neutral file is one acceptable format) and information describing the boundary conditions to be applied (specified displacements, temperatures, etc.). Here, a GEAE in-house application called SIESTA is the service provider. SIESTA is wrapped as a FIPER service and published at a remote Unix workstation. The wrapper accepts as input a PATRAN neutral file and generic boundary condition information. The wrapper produces SIESTA native commands that apply the specified boundary conditions to the meshed model. As in the case of the meshing the geometric tags are utilized to associate boundary conditions to particular geometric features. The output from this service is in a form suitable for a particular engineering analysis application such as ANSYS®.[8]

Analysis Solution

Once the model has been meshed, boundary conditions applied, and materials selected, the model is ready for solution. This service takes the specified input and invokes the appropriate solver on the model. In the current study ANSYS® is wrapped as a FIPER service. The wrapper takes as input an ANSYS® input file and simply issues a system call which executes ANSYS®. The results of the service are returned in the form of VRML (Virtual Reality Modeling Language) files that summarize the results. The ANSYS® service is located on a local high end compute server.

Use Cases

In order to determine the required functionality of FIPER, a set of use cases was developed. The use cases were divided into three major categories: System level analysis/design, sub-system analysis/design, and component analysis/design. FIPER should be flexible enough to handle requirements in all these regions. The use cases are represented by use case diagrams and sequence diagrams. Standard terminology as specified in the Unified Modeling Language (UML)[9] is used. A use case diagram and a sequence diagram for the mechanical analysis of a turbine component is shown in Figures 11 and 12.

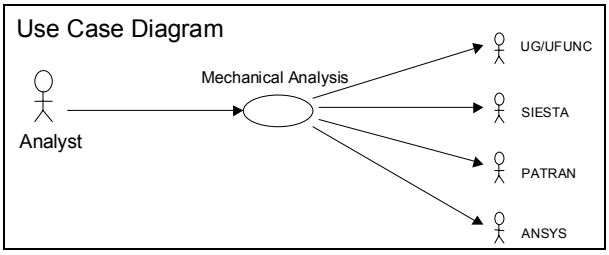

Figure 11: Use Case Diagram

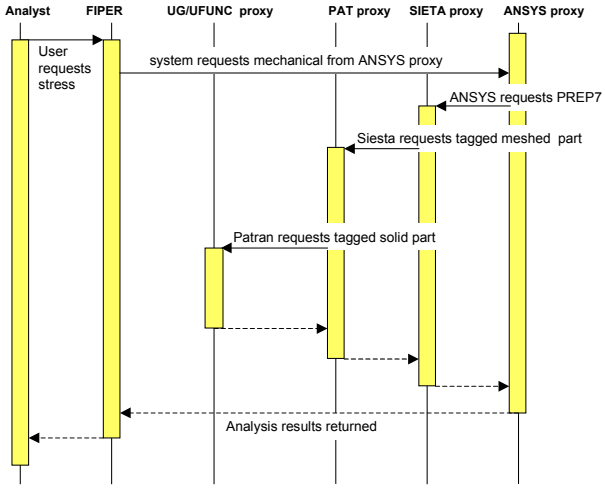

Figure 12: Sequence Diagram for Turbine Mechanical Analysis

The system level use cases focus on the "intelligent scaling" of a given system, for example an entire gas turbine engine. Intelligent scaling refers to the resizing of the system based on the use of a KBE system. The KBE contains rules ranging from standard design practices, simple empirical equations, to the invocation of high-end engineering analysis codes. Once the system level resizing is complete, FIPER should support the ability to "zoom" in on a given component or subsystem and perform a detailed analysis to verify the results produced by the intelligent scaling. The sub-system use cases address a collection of components and employ preliminary level analysis applications along with some detailed analysis level applications. These use cases also include the use of formal optimization techniques to aid in performing robust and optimal design.

The last class of use cases, component level analysis/design, addresses the requirements for

performing MDO/robust design with high fidelity analysis codes such as FEM and CFD.

Outlook

The material presented in this paper represents the first six months of work into a four-year research contract. While a lot of effort has been put into developing the architecture, defining the Intelligent Master Model, and setting up a suite of use cases representing typical problems encountered in turbine engine development, a number of technical risks still remains. Web technology is developing rapidly, but so far the engineering community has leveraged very little of these emerging tools. Sun's Jini™ technology was intended for distributed hardware devices, not software, yet conceptually there is no reason why it should not be applicable in this type of environment.

The example presented, the turbine blade mechanical analysis, is the first use case - and FIPER demonstration that - is currently being implemented. Over the next three years, these use cases will be expanded to cover the range from component to subsystem and system level design, analysis, and optimization, up to the intelligent scaling of a complete turbine engine core.

If the project is successful, it will constitute a complete paradigm shift in the use of engineering software. The authors anticipate to keep the scientific community informed about the progress of the work through subsequent publications and conference presentations.

Acknowledgments

This research is jointly funded through the National Institute for Standards and Technology-Advanced Technology Program (NIST-ATP™) and the General Electric Company. The authors would like to acknowledge this support, as well as the valuable input from the whole FIPER team.

References

[1] Röhl, P.J.; He, B.; Finnigan, P.: *A Collaborative Optimization Environment for Turbine Engine Development*, AIAA 98-4734, Proceedings, 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, St. Louis, MO, September 1998

[2] He, B.; Röhl, P.J. et al.: *CAD and CAE Integration with Application to the Forging Shape Optimization of Turbine Disks*. Proceedings, 39th AIAA/ASME/ASCE/AHS/ASC Structural Dynamics, and Materials Conference, Long Beach, CA, April 1998

[3] Federated Intelligent Product Environment, Technical Proposal, Ohio Aerospace Institute, General Electric, BFGoodrich, Parker Hannifin, Engineous Software, Ohio University, Stanford University, April, 1999

[4] Bailey, M.W.; Irani, R.K.; et al.: *Integrated Multidisciplinary Design*, Presented at the XIV ISABE conference, Florence, Italy, September, 1999

[5] Unigraphics V15 User Documentation, Unigraphics Solutions, Cypress, CA, 1999

[6] Intent User Manual, Heide Corporation, Medfield, MA, 2000

[7] PATRAN V8.0 User Documentation, MacNeal-Schwendler Corporation, Costa Mesa, CA, 1999

[8] ANSYS V5.5 User Documentation, ANSYS Inc., 1999

[9] The Unified Modeling Language User Guide, G. Booch, J. Rumbaugh, I. Jacobson, Addison Wesley Longman Inc., 1999