Internet-based Engineering Service Bureau (ESB) Technology

Andrew J. Scholand and Russell S. Peak

Technical Report EL003-1999A

Engineering Information Systems Lab
http://eislab.gatech.edu/
CALS Technology Center
Georgia Institute of Technology
Atlanta, Georgia USA

August 20, 1999

Table of Contents

Abstract

This white paper introduces the concept of an Internet-based Engineering Service Bureau (ESB), and defines both the business need for ESBs and the benefits of ESBs to distributed organizations. An ESB provides a fee-based analysis of physical behavior, ranging from 'self service', where customers interact with pre-developed analysis modules directly, to 'full service', where ESB consultants carry out all aspects of the analysis for the customer. The ESB also offers the opportunity to utilize pre-developed, validated analysis models parameterized to adjust to product specifics. These predefined analysis models are wrapped in graphical user interfaces (GUIs) which hide analysis-specific details (such as finite element mesh density) and request analysis parameters in product-specific terms. This paper further details the distinctives of an *Engineering* Service Bureau, as opposed to generic application service providers, and outlines the technical infrastructure needed by both the clients and the service bureau itself. These concepts are illustrated with examples from U-Engineer.com, a pilot commercial service bureau established by the ProAM Project currently in use by several DoD suppliers.

Document Reference:

A. J. Scholand and R. S. Peak (Aug 20, 1999) Internet-based Engineering Service Bureau (ESB) Technology, Georgia Tech Engineering Information Systems Lab Technical Report EL003-1999A.

1 Introduction

Complex engineering systems are typically produced by a primary systems integrator (the Prime) purchasing or contracting out to an extended supply chain of manufacturing corporations. Many of the companies involved in the supply chain are small to medium enterprises (SMEs) who are niche experts at producing a single component or families of components. Over 50% of the design/manufacturing effort for products such as such as automobiles, aircraft, computer systems, and energy systems are carried out by SMEs [Ramesh et al., 1995].

These highly complex systems typically stand to benefit greatly from analysis of physical behavior. For example, a lack of effective analysis in the Electronics industry results in delays that are both costly and have serious consequences. Predictive analytic capability improvements in this $300 billion dollar industry therefore have a high improvement potential.

However, the SMEs that make up the majority of the manufacturing network are typically not able to fully leverage their process and product expertise because they cannot perform sophisticated 'what-if' analyses to study the product-wide effects of cost saving measures or other product enhancements. There are a number of barriers that make it difficult in practice for the SME to carry out this level of analysis:

- **Access to *ready-to-use* analysis information**. It is typically difficult for the SME to find proven analysis models specific to their domain, and when they are found, they are usually in a paper-based form. Depending on the model complexity, it may be difficult to implement and use a paper-based description (as opposed to a computer-based description) of an analytical model. Technical articles may not fully expound details concerning underlying assumptions, boundary conditions, and material properties that are needed to effectively use the model. The other alternative for obtaining analysis models, developing the models internally and validating them experimentally, is difficult and demanding in terms of engineering resources.
- **Cost**. A considerable expense is involved in both infrastructure investments (hardware and software), and intellectual investments (training and model development). For example, a leading FEA tool with only mechanical and thermal capability has an initial purchase price of $25,000, and a yearly maintenance fee of $5,000.
- **Ease of use**. The general nature of most CAE analysis tools means that they have complex interfaces that are difficult for the non-expert to use. CAE concepts, in addition, may not be familiar to manufacturing personnel with the most intimate knowledge of the processes of concern.
- **Lack of rapid solutions**. The development of models of products or processes within the analysis tools is typically a laborious and time consuming process, even before model validation by correlation to experimental data is considered. In addition, they cannot exploit domain-specific structure in the problem being analyzed, so product descriptions must be built up from scratch.

The SME's analysis requirements are typically low volume and may not justify the resource expenditures required to overcome these barriers and develop in-house analysis capabilities. However, SMEs can bring to analysis:

- Detailed product knowledge, such as the orientation of laminates in a Printed Wiring Board (PWB)
- Detailed process setting information (which often is needed as boundary conditions or other parts of an analysis)
- Knowledge of what possible alternatives exist to address issues with the product or process and
- Detailed knowledge of the costs associated with these different options

In addition to providing these benefits of analysis in the entire supply chain, SMEs themselves can also benefit from the use of analyses directly. SMEs can use analysis to verify that desired 'Design-For-Manufacture' (DFM) product improvements will work as intended before recommending them to the Prime. In situations where the Prime or an organization higher up the supply chain has delegated design authority to the SME, analysis can enable the SME to create better designs. The SME can also use analysis to improve processes in addition to designs. Analyses can reduce the time needed to identify and troubleshoot problematic manufacturing processes, and to better tune and refine (i.e. optimize) established processes. Finally, sophisticated analysis outputs, such as stress and displacement plots, are an effective marketing tool for the SME. They enable the SME to demonstrate a "value added" service to the Prime that differentiates them from other competitors.

This paper emphasizes SME needs, how these needs can be met through an Internet-based Engineering Service Bureau, and experiences with the U-Engineer.com Electronic Packaging ESB developed in the ProAM project [EIS Lab, 1999c]. There are a similar set of issues, albeit less exacerbated, within a large conglomerates or corporations, where the expertise of a particular group needs to be deployed to a larger audience for maximum effectiveness in the manufacturing environment. In this case, the Engineering Service Bureau would probably be connected to an Intranet internally within the company's wide area network (WAN) to improve the security of the ESB, but the other aspects described below are relevant.

1.1 Business Need

Since often only the SME has this sufficiently detailed product and process knowledge needed for analysis, it is difficult for other members of the collaborative team (such as the Prime) to fulfill the same level of analysis function. A lack of analysis at the SME level diminishes the engineering knowledge of the entire product development team. An additional benefit of SME-run analyses is the employment of the results in a meaningful way. Analysis results can help the SME perform process simulation tasks to determine optimal process settings and avoid manufacturing problems, thereby improving manufacturing yields. The SME can analyze product performance to judge design alternatives. If product design changes appear warranted due to cost or performance considerations, the analyses allow the SME to quantify the benefit associated with different product configurations so that the Prime can make informed decisions on cost/benefit tradeoffs.

Other possible drivers for analysis outsourcing are one or more of the following. 1) The SME is unable to be price competitive in the market when analyzing their own products and processes. 2) The SME wants to concentrate on new product technologies in their domain of expertise, and therefore has greater analysis needs than when producing its mature technology designs. 3) The SME cannot provide enough in-house analysis to meet current demand, but the demand may be only a temporary peak, and so the capital/personnel commitments necessary for analysis 'production' expansion are risky. 4) The SME has little domain analysis experience (for instance, SMEs increasing their sophistication to expand their market share, or new company start-ups).

1.2 Conceptual Description

The Engineering Service Bureau (ESB) paradigm presents the opportunity to empower SMEs with cost-effective yet advanced analysis capabilities, dramatically improving the quality of complex engineering systems. A Service Bureau provides fee-based physical analysis capabilities, ranging from 'self service', where engineers at the SME interact with pre-developed analysis modules directly, to 'full service', where ESB consultants carry out all aspects of the analysis for the SME. Because an ESB specializes in providing analysis services, the utilization of the computer aided analysis tools is much higher than is likely in an SME focused on manufacturing. Since infrastructure (software and hardware) and intellectual property investments are averaged across the user base of an ESB, the per-use cost for any given analysis run by the SME is much less than the cost of maintaining an in-house capability. The ESB also offers the opportunity to utilize pre-developed, validated analysis models parameterized to adjust to product specifics. These predefined analysis models are wrapped in graphical user interfaces (GUIs) which hide analysis-specific details (such as finite element mesh density) and request analysis parameters in product-specific terms. This encourages frequent 'self-serve' use by SME personnel, for example by members of the technical sales department when responding to requests for quotes.

1.3 Internet Economics

The Internet is an important enabler for the Engineering Service Bureau, as it provides an ubiquitous interface which enables a single point of service to reach a large number of clients anywhere in the world. Internet-based businesses are becoming increasingly prevalent. The U.S. Internet economy generated an estimated 301.4 billion dollars of revenue in 1998, employing about 20 percent of the 5.9 million Americans who work in the high technology field [McCarty, 1999].

The Engineering Service Bureau is similar in many respects to Application Service Providers (ASPs), which have received considerable press coverage recently. [Briody, 1999, and Petreley, 1999] Application Service Providers are outsourcing companies that provide centrally managed software applications through the web. ASPs typically focus on generic application outsourcing, where similar services are offered to a large number of customers, rather than high customized, single client installations. This one-to-many delivery model allows ASPs to deliver high end applications, such as Enterprise resource planning (ERP) and customer relationship management (CRP) to market segments too small to afford stand-alone installations. Recent estimates [McCarty, 1999] place the ASP market for just these two applications at $2 billion worldwide in four years, up from $150 million in 1999. The overall application outsourcing market is expected to reach $16 billion in that time.

The market segment for ASPs is similar to that for ESBs, typically small to medium sized companies, particularly those that seek rapid growth or lack the Information Technology (IT) infrastructure to install and administer the sophisticated applications offered by ASPs. The potential ASP market also includes those Fortune 1000 companies looking to respond quickly to competitive pressures by launching electronic-business applications, hooking up with supply-chain partners, or handing off core functions during mergers and acquisitions or restructures.

However, ASPs differ from ESBs in that ASPs exist primarily to provide access to the full functionality of their software, thus earning the term rentable applications. Thus, many of the ASPs are software vendors themselves, such as Oracle, PeopleSoft, SAP, and Baan. ESBs, on the other hand, are offering a limited subset of the analysis software's full capability, having already built product-specific analysis model templates into the Internet-accessible front ends they provide. Although not as flexible, the value-added of this approach is in instant application to problems of interest.

1.4 U-Engineer.com- Exemplar ESB

The sections below detail what technologies and services must be deployed to establish an Internet-Based Engineering Service Bureau. To provide an example of each functional requirement, relevant technology descriptions are given for U-Engineer.com, an ESB developed for the Electronic Packaging and Interconnecting Industry by our research group. For more information on U-Engineer.com, please visit our web site at http://www.u-engineer.com.

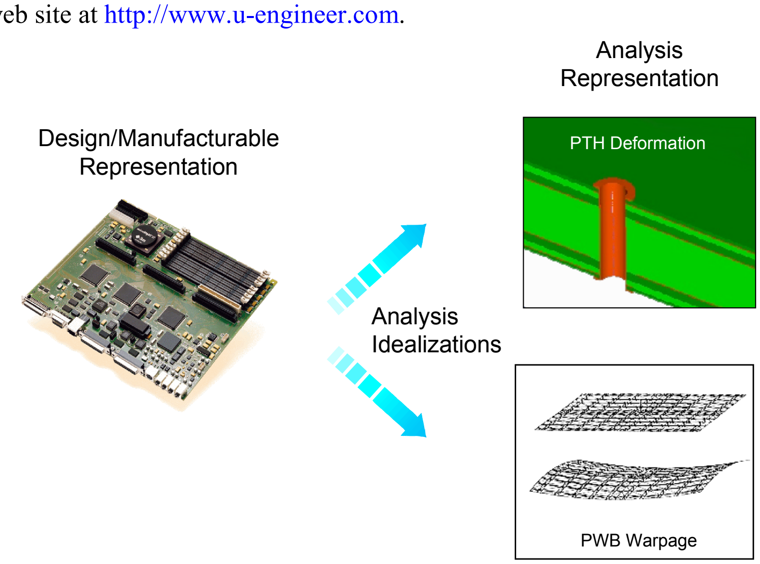

Figure 1-1: Main U-Engineer.com Analysis Families

As shown in Figure 1-1, currently available U-Engineer.com analysis modules are geared toward the Electronics product domain. These analyses fall into two groups, Plated Through Hole (PTH) deformation and fatigue, and Printed Wiring Board (PWB) warpage. Plated Through Holes, also known as Plated Through Vias, form an electrical connection between the different layers of a PWB. Typically, copper is plated into a hole that has been drilled through the dielectric material of the PWB. When the board is heated, either during manufacture or through product use, the dielectric material experiences a pronounced vertical swelling, which applies a stretching stress to the copper barrel, and tends to push the donut shaped periphery of the PTH (often called the pad) upwards. As the board cools, the copper experiences a compressive stress. If the PTH structure survives the first few applications of these loads in the manufacturing environment, use-based reversed loadings (power on/power off) will cause fatigue of the copper in the PTH, and lead to eventual failure. It is important to be able to predict if a given design will successfully survive the manufacturing process, and when it will fail under heat/cool cycles in the post-manufacture/usage environment.

There are several models in the literature that deal with the stress, strain, and fatigue life of PTH structures. U-Engineer.com has implementations of 3 different PTH models. One such model is available in paper form from the IPC in the design standard IPC-D-279 [Engelmaier, 1996]. The salient equations are illustrated in Figure 1-2. Note that the equation for fatigue life, N_f, cannot be solved explicitly. These equations are implemented in the example interfaces shown in Section 3.1.2 and Section 3.1.3.

$$\sigma_{avg} = \frac{\left[(\alpha_E - \alpha_{Cu})\Delta T + S_y \cdot \frac{E_{Cu} - E_{Cu}'}{E_{Cu} \cdot E_{Cu}'}\right] A_E \cdot E_E \cdot E_{Cu}'}{A_E \cdot E_E + A_{Cu} \cdot E_{Cu}'}$$

$$\Delta\varepsilon_{avg} = \frac{(\alpha_E - \alpha_{Cu})\Delta T \cdot A_E \cdot E_E - S_y \cdot A_{Cu} \cdot \frac{E_{Cu} - E_{Cu}'}{E_{Cu}'}}{A_E \cdot E_E + A_{Cu} \cdot E_{Cu}'}$$

$$N_f^{-0.6} D_f^{0.75} + 0.9 \frac{S_u}{E} \left[\frac{e^{D_f}}{0.36}\right]^{0.1785 \log \frac{10^5}{N_f}} - \Delta\varepsilon = 0$$

$$N_f(x\%) = N_f(50\%) \left[\frac{\ln(1 - 0.01x)}{\ln(0.5)}\right]^{\frac{1}{\beta}}$$

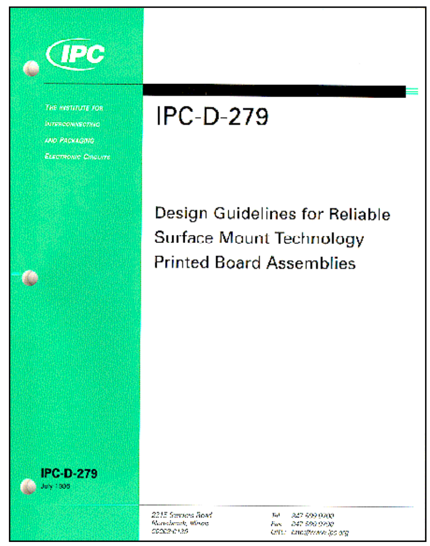

Figure 1-2: Formulas for analyzing Plated Through Holes/Vias from Appendix B of IPC-D-279

PWB warpage models are more difficult to find in the literature. Warpage arises because each of the lamina in a PWB has different stiffness and thermal expansion coefficients, depending on the material (glass fabric style, epoxy type) used and the nature of the copper wiring on the different layers. When constrained to move together by lamination into a PWB, these lamina drive warpage and residual stresses in the laminate. As shown in Figure 1-3, a given functional specification can be manufactured in several different configurations, each of which may behave very differently physically.

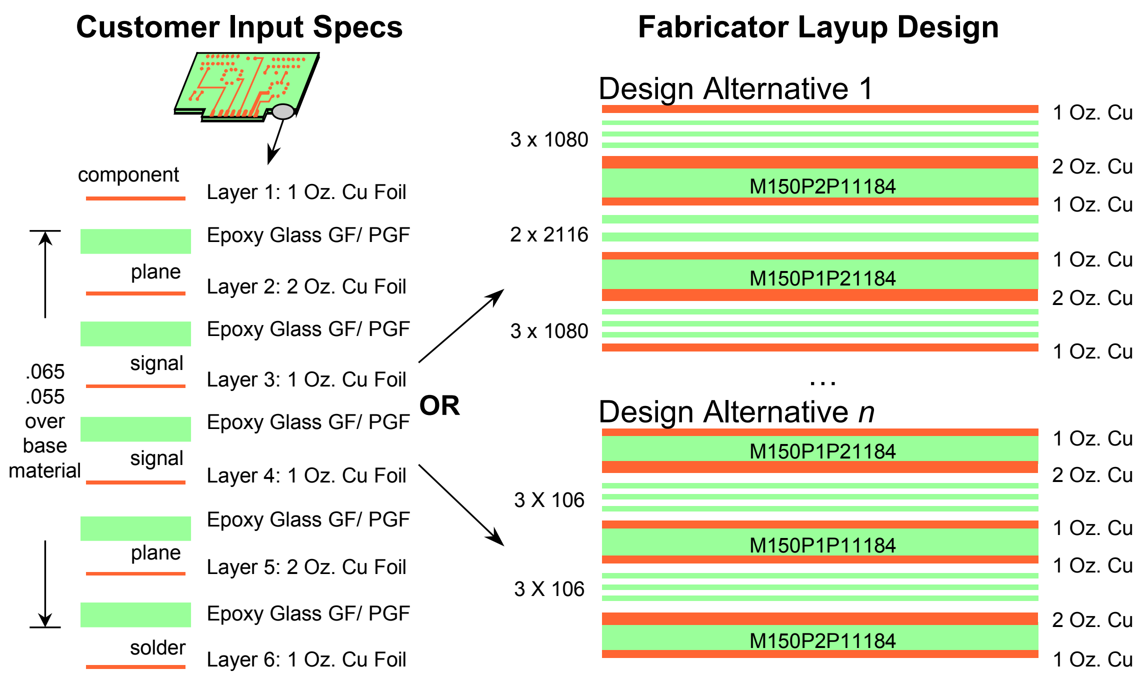

Figure 1-3: PWB Manufacturer's latitude in physically realizing PWB specification

Since warpage is so closely tied to the materials used in the PWB stackup, the chief tool for analyzing warpage at U-Engineer.com is a Java-based "thick client" [Wilson et al., 1999] tool that accounts for the actual prepregs, laminates, and copper foils used to realize a specification for a PWB (Figure 3-6).

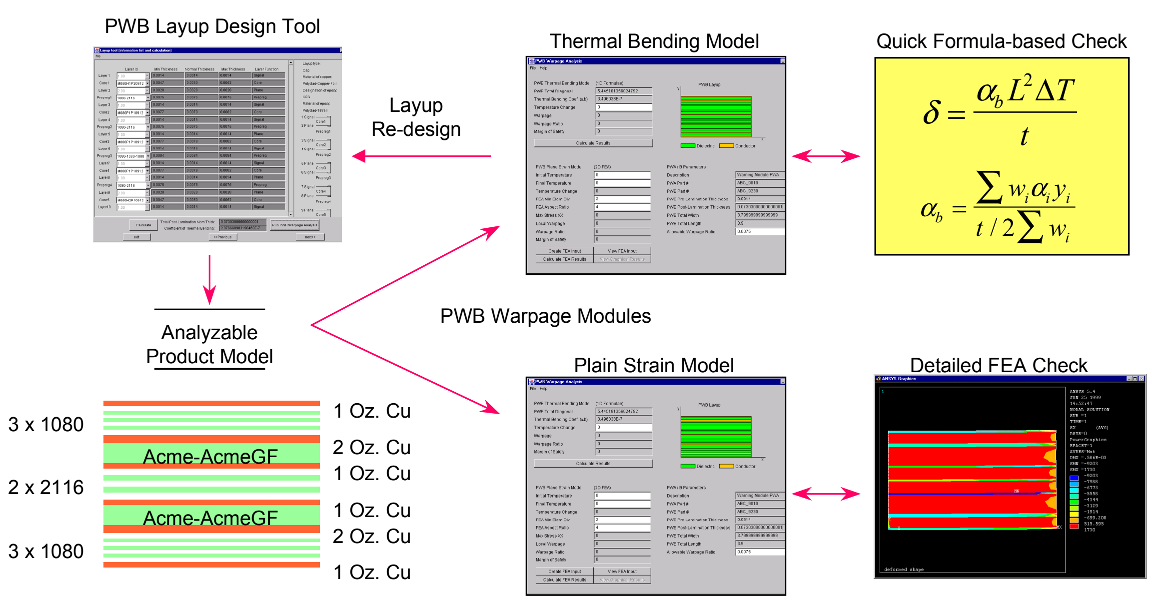

Figure 1-4: U-Engineer Warpage Models Accessible from the Thick Client Interface

As illustrated in Figure 1-4 above, U-Engineer.com currently provides two formulations to derive an estimate of PWB warpage from the information gathered from the thick client interfaces. The first is a formula based model developed by Tamburini et al. (1996b) that considers isotropic linear material properties in determining the tendency of the PWB to deform under temperature loads. The second is a plane strain Finite Element model that also uses only isotropic material properties. The remaining sections give examples of how analysis models like these can be packaged and delivered via ESB technology.

2 Generic Guidelines For Establishing An ESB

The following guidelines are based on our experience setting up U-Engineer.com, first as a fully server-based application with X Windows interfaces [Scholand, et. al., 1997], and later in incarnations as a client-server application utilizing CORBA and a web-wrapped application server. We consider the following to be important areas for attention when implementing an ESB, regardless of the toolset or functionality provided.

2.1 Security

SMEs must have confidence in the ESBs they deal with in order to enter a partnership of trust with them. In some cases, the Prime contractor will mandate a certain level of security. Even in the absence of external decrees, once a SME has agreed to upload their product and process information to run analysis tools at an ESB, that information needs to be secured from unauthorized access. Furthermore, in the "virtual" environment of the Internet, the ESB must establish its identity and credibility to protect itself and its customers from impostors. Finally, the exchange of data to and from the ESB must be encrypted to prevent it being intercepted by third parties. Secure Sockets Layer (SSL) transmissions and Digital IDs [Verisign, 1997] accomplish the latter two goals, guaranteeing an authentic identity and ensuring data security during transmission.

A Digital ID provides an electronic means of verifying that the individual or organization is who they claim to be. The Server Digital ID, for example, provides third-party evidence of an Internet Web server's authenticity, establishing that the server is operated by an organization with the right to use the name associated with the server's Digital ID. Web browsers generally perform server authentication automatically. The user is only notified if authentication fails due to an expired certificate, mismatched URL, or other problem.

The Secure Sockets Layer (SSL) is a technology developed by Netscape and adopted by many vendors producing web-related software (Figure 2-1). It negotiates and employs three fundamental security services:

1) *Mutual Authentication*. SSL 3.0 allows the identities of both the server and client to be authenticated through exchange and verification of their Digital IDs. (Figure 2-1, however, shows only server authentication.)
2) *Message Privacy*. All traffic between an SSL server and SSL client is encrypted using a unique session key. The server's key-pair is used to encrypt the session key itself when it is passed to the client.
3) *Message Integrity*. SSL also protects the contents of messages exchanged between client and server from being altered enroute.

The diagram below illustrates the process that establishes SSL protected communications between a Web server and a client, although it is also possible to secure other Internet protocols such as Internet Inter Orb Protocol (IIOP) for secure CORBA access. All exchanges occur within seconds, and require no action by the person connecting.

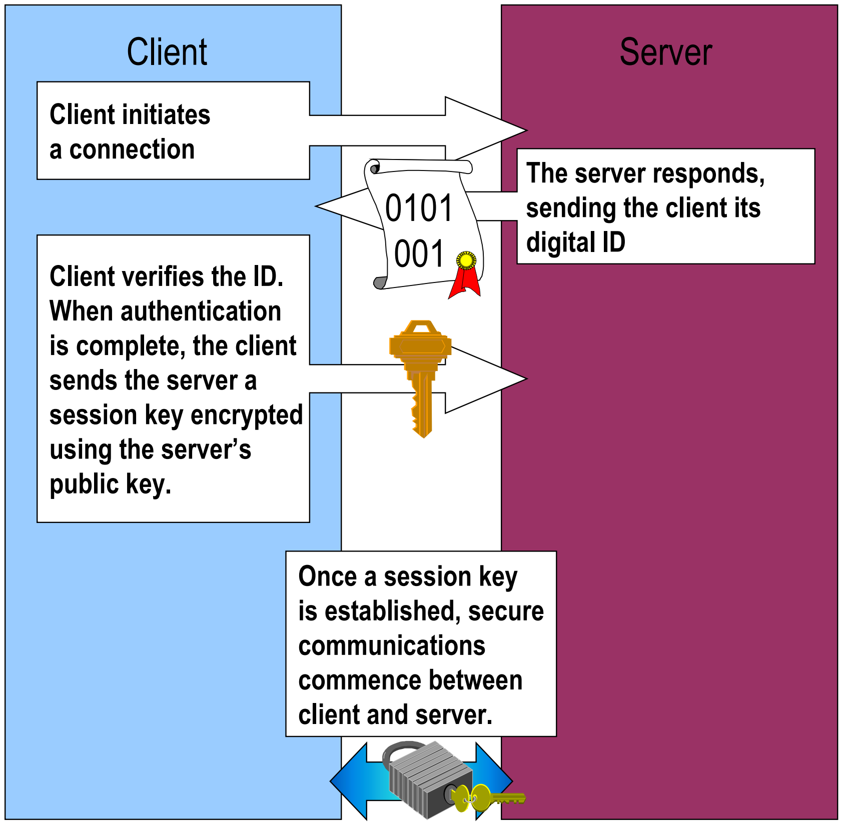

Figure 2-1: Establishment of a Secure Sockets Connection

Security also carries implications of both protection from outside malicious agents who may attempt to mount attacks from outside to either the client or server accounts, and confidentiality, in which customer data is completely compartmentalized among customers of the ESB. Levels of security vary somewhat depending on the type of technology the ESB employs, so further in-depth discussions of security are discussed for each of the technology sections below. A table in the conclusion section summarizes the relative strengths and weaknesses of various ESB technical approaches with respect to security.

2.2 Data Storage

Each ESB needs to make a decision regarding data storage for the user accounts, i.e. the ESB must decide if data is persistent between sequential accesses to an ESB. The methodology used to implement persistent data storage will of course be access method dependent. Options include client side data storage, such as "magic cookies" (a limited ASCII flat file database) in client side web browsers, or more sophisticated programmed databases, such as STEP databases embedded in thick client code. Client side data storage requires either restrictions on browser versions permissible to access the ESB services or installation of executable code on the client. Server-side storage is also possible, although persistent objects in a CAE environment would then dictate each customer (or aggregate customer such as a corporate account) has his or her own instantiation of the object environment. This is potentially costly for the ESB because additional capacity must be purchased as the number of clients increases. The first version of the U-Engineer Toolset (DaiTools) used server-side storage, but this architecture was changed to client-side data storage in the ProAM project.

Non-persistent data is potentially more inconvenient for the clients. If data files need to be uploaded to the server for processing (server-side storage), this may be moderately expensive and time-consuming. Since many small enterprises obtain their Internet access via relatively slow dial up links, and pay for connect time hourly, uploading large product data files can be problematic. Even when processing client side data, re-entering or reloading product data between sequential uses of the ESB toolset can be inconvenient. This may be less of a factor depending on the rapidity of iterations in the design environment, however. If each time the ESB is accessed, a new design or designs are analyzed, the advantages of keeping the last upload in the SME account are much reduced.

3 ESB Server Technologies

The first step in setting up an Internet Service Bureau is construction of the server side, including both the hardware and software to serve web documents and the hardware and software to handle the computational needs of the ESB clients. In the sections below we discuss important technological design decisions which must be made before committing any resources to establishing the server-side infrastructure. We then get into specific hardware and software options currently available for implementing these decisions.

3.1 Conceptual Approach to Servicing Clients

The first technological decision to make is how to allow clients to connect to the server-based services over the Internet. There are two main classes of approaches, "thin client" and "thick client" [Korzeniowski, 1996]. These names derive from the computational demands on the client machine- a thin client requires little overhead, relying chiefly on the server for its processing needs. Consequently, there is only a small amount of code that must be run on the client machine. "Thick clients" use a much greater library of code on the client side to off-load some of the processing tasks from the server to the client.

We label interfaces utilizing Internet browsers as 'thin clients' as the amount of additional code to create a usable control mechanism for CAE tools is minimal. In some sense, 'thin client' is somewhat of a misnomer, since the client machine must be running the Internet browser software itself, which can be substantial in size.[1] However, ideally, analysis services need to be accessible from any computer, running any OS, connecting from anywhere, and currently the closest software platform approaching this ideal state is the Internet browser.

3.1.1 X Windows Based Interfaces – "Thin Client"

X Windows is a standard for remotely displaying graphical windows which evolved for Unix-to-Unix communication. To display X Interfaces on IBM-compatible Personal Computers (PCs) running the Microsoft Windows operating system, X Windows emulation software is required. This software is available as a full suite of Unix connectivity software, such as Hummingbird's Exceed [Hummingbird, 1999], or as an Internet Browser "Plug-In". The use of these tools in the first generation of U-Engineer.com [Scholand et al., 1997, Peak et al. 1997] is illustrated below in Figure 3-1.

The SME client browses a web server to review analysis documentation, and uses an HTTP form to upload the product data to the ESB server. The relevant web pages should have links to automatically launch telnet sessions, from which the graphical interfaces to relevant Unix tools can be launched. If the X-Windows emulator has been initiated on the client, and the ESB hosts are allowed to send displays to the client, the X-Windows GUI will be rendered on the PC. Further user-tool communication is conducted over the X Windows protocol.

[1] The full Internet Explorer^{tm} 5 installation, for example, is close to 89 Megabytes. Even the smallest version of Internet Explorer^{tm} 4.0 takes approximately 2 hours to download over a 28.8 modem.

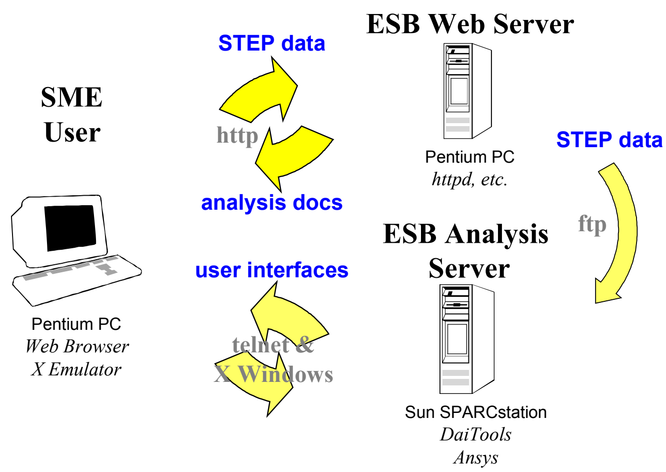

Figure 3-1: Data Exchange with X-Windows ESB Access

X Windows interfaces provide several advantages. Since many CAE applications have Unix Graphical Interfaces, little to no work is required by the service bureau to provide direct access to the functionality of the tools. This is particularly helpful to advanced analysis users, since after an analysis integration approach (see Section 3.3.5) has created an initial model, they may wish to tweak the model further or consider 'what if' perturbations to the model.[2] In addition, the use of X Emulators allows the ESB to develop a single product and deploy it to both Unix and PC environments.

There are several disadvantages to this approach however. X Windows requires bandwidth slightly higher than the typical dial-up Internet access line to display screens. The visual "style" of the interface is based on the server's operating system, and so may not be immediately familiar to a PC user. Finally, different vendors' implementations of the X Windows interface differs somewhat, with not all X Windows emulation products supporting the full suite of possible X behaviors. If a server executes a behavior that is not supported, the client X Windows software may crash. An X Windows approach is a purely server-based approach, and so depending on the analysis integration methods employed, the client may have to upload and entrust the server with a copy of his product data. This is costly to the client in that connect time charges may apply to the upload itself, and the security of that product data is subsequently in the hands of the Engineering Service Bureau.

We believe these disadvantages make it infeasible for the ESB to deploy this technology when service of Small to Medium Enterprises is their prime customer market segment. However, if supporting a mix of Prime design engineers, first tier manufacturing engineers, and SMEs is possible, the advantages of a Unix-based system may outweigh the disadvantages.

3.1.2 Web Based Interfaces - "Thin Client"

A web-based interface to analysis tools presents several obvious advantages important from an ESB point of view. The web interface and its associated navigational conventions are increasingly ubiquitous. Future 'information appliance' trends, such as web surfing through television and portable phone devices, will probably further expand the exposure of this data presentation metaphor. The familiarity of this method increases the prospective audience for using the tools behind the web interface. Web interfaces can be rendered by a wide variety of Web browsers, available on virtually all computer operating systems. Some of these web browsers, such as Opera, have been optimized for execution on computationally lightweight clients. Both the familiarities of the web interface and its widespread availability provide an opportunity for ESBs to market their services to the widest possible customer base. An example of a web-based interface for analysis of PWB plated through holes is illustrated in Figure 3-2. Note that this graphical interface is a front-end to the IPC D 279 equations shown in Figure 1-2 for modeling PTH deformation and fatigue.

[2] Note however, that the software license of the CAE software may preclude providing direct access to the solution tools in this manner.

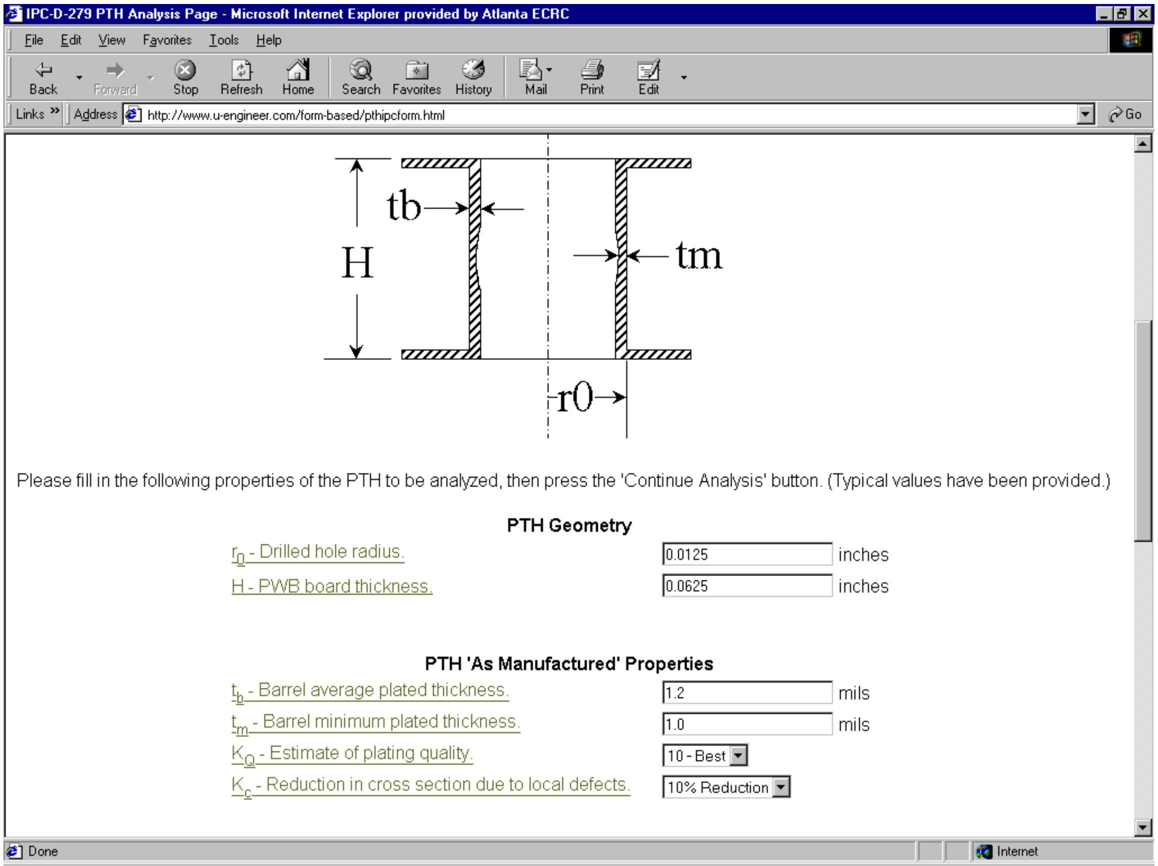

Figure 3-2: Example "Thin Client" Web Interface for IPC D 279 PTH Analysis Module

Web interfaces are much less bandwidth-intensive than other technologies for presenting graphical interfaces remotely (such as X Windows). This maximizes the perceived performance of ESB tools over slower dial-up Internet connections. These slower connection methods predominate in the Small Enterprise Enterprises (SMEs) that ESBs serve.

Web browsers are relatively mature client applications, and employing them to render and support user interfaces represent a good reuse of software components. In addition, most current browsers support the Secure Sockets Layer (SSL) transmissions and Digital IDs described in Section 2.1 above. By using current generation web servers and browsers to protect SME intellectual property during transit across the Internet, these security measures are easy to implement. Of course, the ESB must not be negligent on other security issues such as secure data storage [DoD 5200.28-STD].

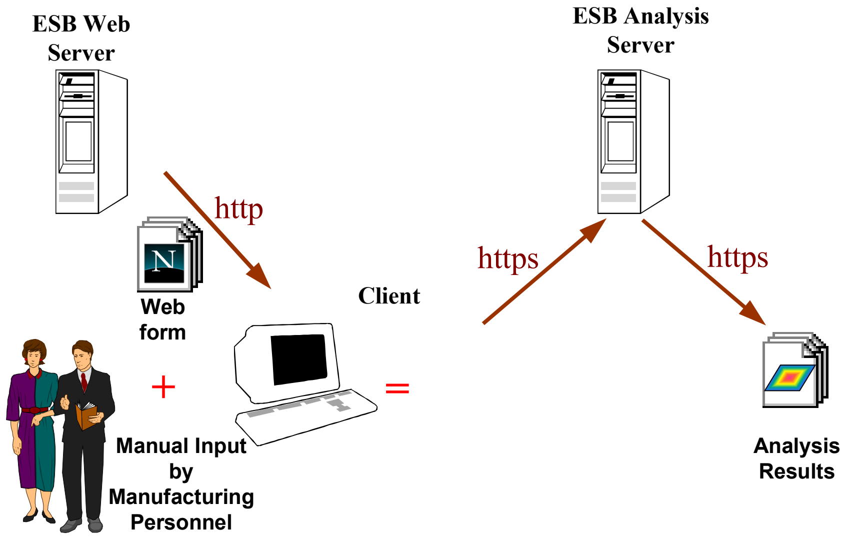

Figure 3-3: Web "Thin Client" Data Interface Diagram

From a computer-aided analysis viewpoint, however, there are a few disadvantages to using a pure Web Interface for direct data gathering (Figure 3-3). The main issue is a reliance on human data processing. In the case of manufacturing process information this manual data entry is appropriate, as the information is readily known to experts in the field and is typically low in quantity. For example, in the specific case of plated through hole analysis, the loads which are applied to the structure are determined from the temperatures of various manufacturing processes on the factory floor, such as the temperature of the solder in a Hot Air Solder Leveling (HASL) machine. Operators are familiar with this temperature, since it is a controlled process variable presented to them by instruments on this machine. In contrast, however, the product data required by the analyses may not be so readily known. For example, the plated through holes which are most at risk of thermo-mechanical failure are the smallest holes. When manually populating fields in a web interface, the operator must determine the minimum hole diameter among the thousands of holes in the PWB. Even in cases where the operator doesn't have to manually determine a maximum or minimum among a large data set, simple re-entering of data is inconvenient. Re-entry of data is also a prime opportunity for the introduction of errors.

So, while a traditional web-based form interface is easy to use, it lacks automated integration with product data and potentially process data. The emerging standard eXtensible Markup Language (XML) presents an opportunity to represent product data in a web-enabled format, and thereby address the data integration issues mentioned above.

3.1.3 Hybrid "Thin Client" – Browser Processed Interfaces

XML (eXtensible Markup Language) is a web-enabled data interchange language derived from the Standard Generalized Markup Language (SGML) [Megginson, 1998]. The World Wide Web Consortium (W3C) approved XML as a W3C recommendation in February 1998. As a platform and vendor-independent technology, XML has been publicly endorsed by almost all of the leaders in the computer industry, including Microsoft.

XML is not itself a set of ready-to-use tags the way another SGML subset, HyperText Markup Language (HTML), is. Rather, XML is used to define application-specific markup languages to represent structured data. XML defines the rules for these markup languages to ensure that they are easy to parse and validate by machine. XML-defined markup languages enable the separation of semantics from the way the data are used by an application or rendered for presentation. This is analogous to the practice of model and view separation in good object-oriented design.

Validation in XML-defined grammars allows an application to ensure that a given instance of data is complete, correctly hierarchical, and with acceptable content values. Comparing a data instance to a

Document Type Declaration (DTD) enforces validity. A DTD is a computer processable description specifying which tags and attributes, and in what order, are allowed in the data instance file. A DTD can be included in line with a data file, or the instance file can point to a DTD using the Universal Resource Identifier (URI) mechanism.

GenX[3] is a specific XML markup language mapped from the Institute for Interconnecting and Packaging Electronic Circuits (IPC) standard, IPC-2510, GenCAM^{sm}. The GenCAM^{sm} standard has been developed to facilitate the communication of PWA/B manufacturing data from the designer to the fabricator. It integrates functional descriptions, test data, and administrative information into a single file format. It is sufficiently detailed for tooling, manufacturing, assembly, inspection and testing requirements. The GenX DTD is identical to the contents of the IPC-251x series of documents; both describe what is legally allowable in the file.

The inherent structure of XML files makes them easy to parse. This, in turn, means that XML parsers can be relatively small and compact. XML parsers have been written in less than 26K of Java [Microstar, 1999] and less than 5K of JavaScript [Miller, 1999]. These low footprints mean that a GenX parser[4] can easily be embedded in a web page (or pages) that acts as the interface to sophisticated product analysis capabilities. The combination of web interfaces and standard-driven product analysis yields several unique benefits.

The most obvious benefit is the improved accuracy of the data input to analysis. Since the same data source used to drive manufacturing also drives the analysis, there is no potential for errors arising from needless (and typically manual) data duplication. Automation of data extraction also enables tasks that are not feasible when humans provide analysis inputs. For example, exhaustive searches over all the input combinations are typically trivial programming exercises, but are infeasible for a human operator. Depending on how the parsing is carried out, it is also possible to increase the intellectual property protection available to the ESB customer.[5] The processing of the GenX file in preparation for computer aided analysis can be performed on either the remote machine (server) or on the local machine (client), depending on where the parser is located. If the parser is loaded onto the local machine, the parser operates on the GenX file in local memory. Only those abstract parameters needed for analysis (what we describe as the *idealized* attributes of the product) are entrusted to the remote machine. Since it is typically almost impossible to reconstitute a detailed description of the product from the idealized attributes, the ESB customers enjoy relative immunity from intellectual property theft in the event the ESB server is compromised. Finally, there is a natural data compression benefit which can be realized if the parser is located on the local (client) machine. The idealization process (abstracting the detailed manufacturable description into a simpler form more amenable to analysis) typically reduces the amount of pertinent data by two orders of magnitude. Thus, under typical usage situations[6], it is much less expensive to send the parser to the client than to send the data file to the host. This is a particularly important benefit to the target market for ESB usage, since most Small to Medium sized Enterprises will access the Internet through dial up links with limited bandwidth.

---

[3] GenX is an abbreviation of the term GenCAM^{sm} XML file.

[4] We use GenX as a representative example of an XML-based standard. Similar results can be expected with other XML based standards.

[5] Note that our use of the term "greater intellectual property protection" here strictly refers to greater protection from outside malicious parties. We assume the ESB to be a trustworthy partner.

[6] Assuming that the typical GenX file will be ~700K and that only a limited family of related analyses will be repetitively performed on a given board.

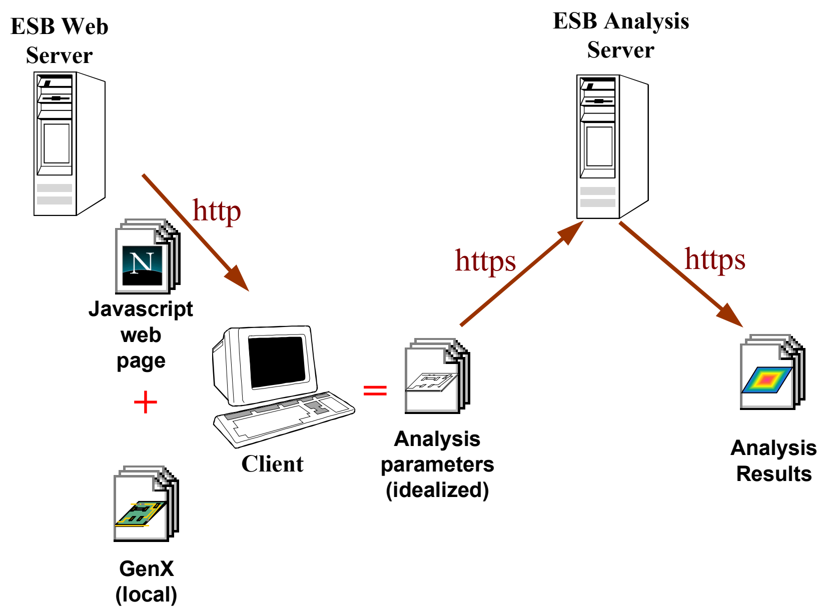

Figure 3-4: Hybrid "Thin Client" XML and Web-enabled Analysis Data Diagram

Thus, the scenario in which an XML-enabled web analysis interface is accessed is illustrated in Figure 3-4. The end user accesses a GenX-capable web page from an Internet Engineering Service Bureau using a standard web browser. The GenX parser downloads to the local machine along with an HTML form or Java applet interface. Using this interface, the end user loads the GenX file into the browser, and the parser operates on the file on the local machine. The idealized attributes are now ready to be sent to the Engineering Service Bureau for sophisticated CAE processing. The example below illustrates this scenario, using an HTML form and Common Gateway Interface (CGI) script to send the data to the ESB server, where it is handed off to an analysis server on the back end.

3.1.3.1 Example U-Engineer.com Application- PTH Analysis

As mentioned in Section 1.4 above, and illustrated in Figure 1-2, one formula-based model for predicting how a given PTH will respond to a given load is documented in the IPC Standard IPC-D-279 [Engelmaier, 1996]. We have implemented these equations as a template in *Mathematica*, a commercially available software package, and provided a web-based pure thin client front end to this analysis model (Figure 3-2).

Illustrated in Figure 3-5 is the GenX-enabled hybrid thin client version of the same analysis model. The frame on the left hand side contains the GenX file. The 'Process' button in that frame runs a JavaScript parser (*Xparse*, [Miller, 1999]) and additional domain-specific routines for processing the GenX file. The output from the local parsing is put into the fields of the IPC-D-279 analysis form, which is displayed on the right hand side frame. Currently, the GenX parsing engine determines the maximum permissible board thickness and the minimum plated through hole diameter, since these are the 'worst case' parameters for this type of analysis. The analysis form can then be edited as usual, in case the operator wants to override any of the automatically generated values. Pressing the 'Continue Analysis' button (not visible in Figure 3-5) submits the form for CGI processing on the web server in the manner described above. The total combined downloadable size of this web interface is 29K, making it convenient to use even over slower Internet connections.

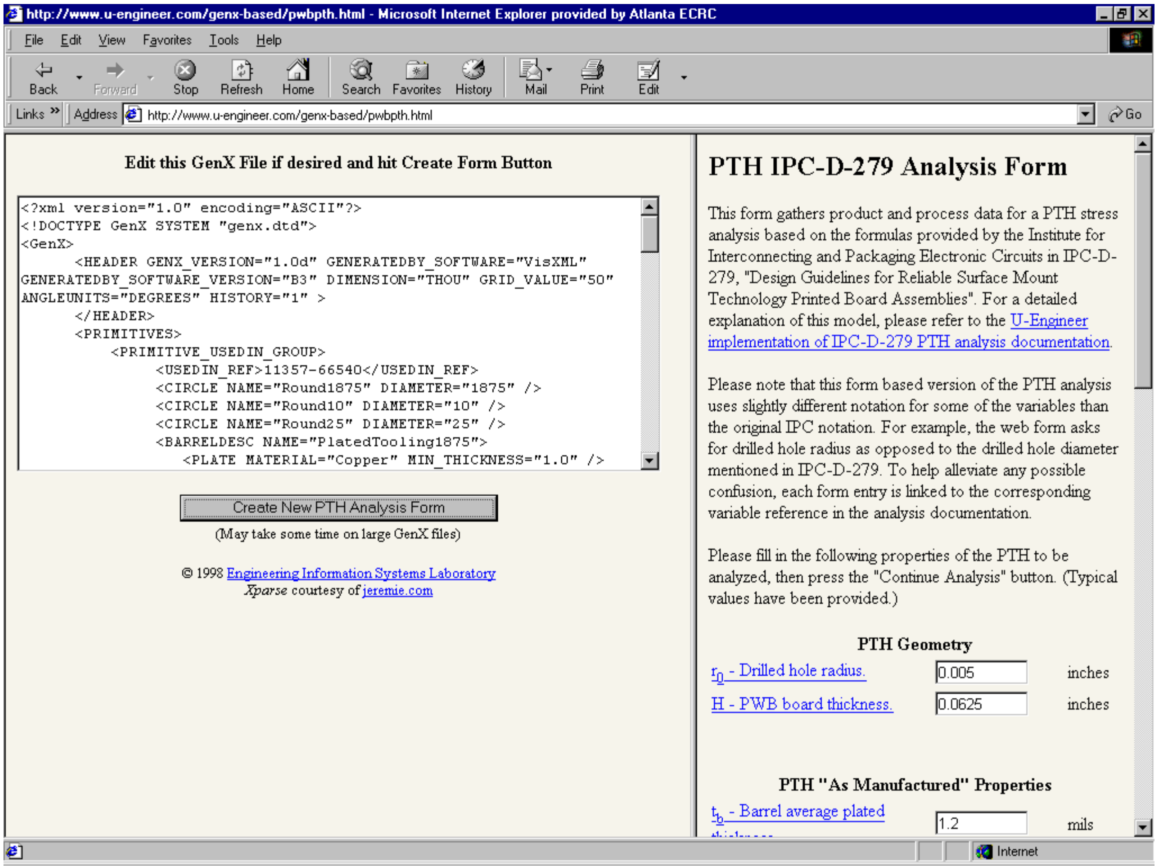

Figure 3-5: Hybrid Thin Client Web Interface Populated with Values from XML Product Data File

3.1.4 "Thick Client" - CORBA Based Interfaces

"Thick Clients" must provide code for three main functions: (1) Graphical User Interfaces; (2) Local Data Management; and (3) Connectivity to the Server.

Graphical User Interfaces need to be developed from whatever fundamental building blocks the programming language offers. The Thick Client GUIs developed for the U-Engineer.com ESB have been written in Java, which offers a number of graphical capabilities in an easy-to-use format. The advantage to developing custom interface code is that arbitrarily complex GUIs can be created, and, assuming the code is distributed as a downloadable application with its own installation procedure, the security restrictions of browser-based active components are avoided. An example Java-based thick client GUI from U-Engineer.com is shown in Figure 3-6 [EIS Lab, 1999b].

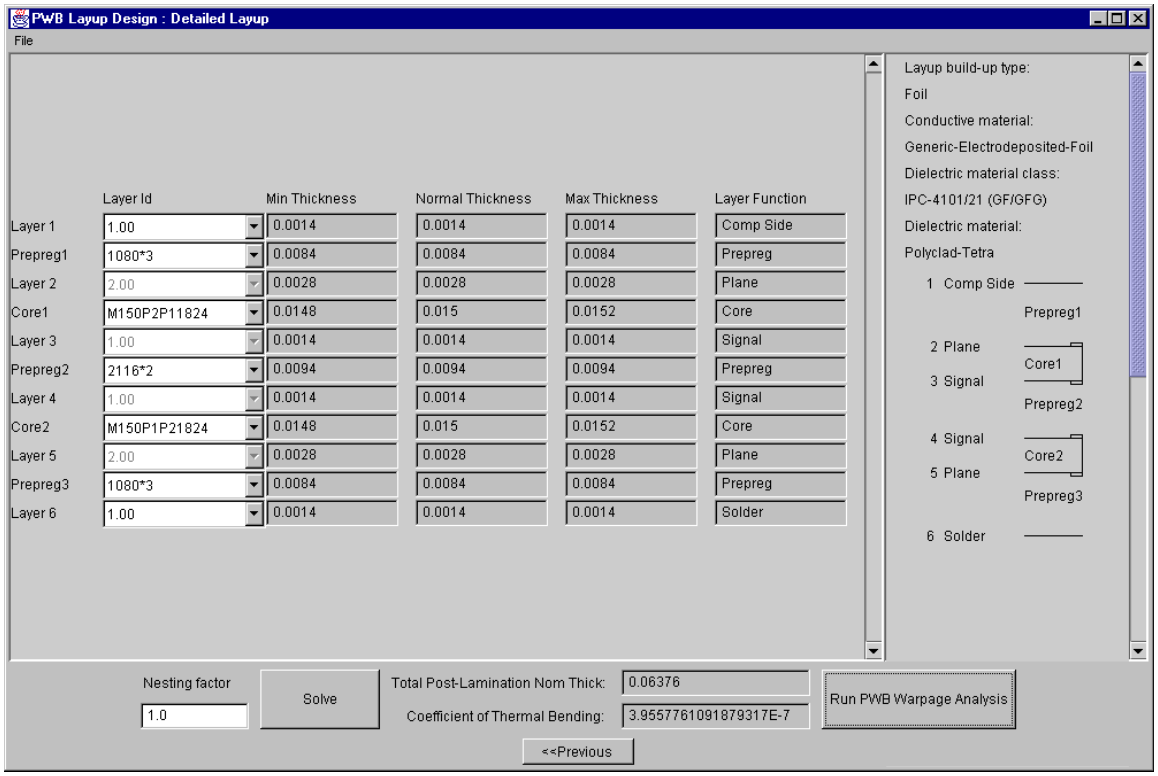

Figure 3-6: Example Java "Thick Client" GUI Showing Customized Display of PWB Stackup

Another task that "Thick Clients" must perform is local data storage. Analysis applications can be most efficiently and effectively implemented if a single, integrated repository is used to provide data entities and idealizations that are shared among multiple analyses. In addition to the data model facilitating design-analysis integration, data mappings are required to retrieve and integrate design information from several sources to populate the repository. Data transformations are typically required to idealize this information (concerning geometry, materials, environment, etc.) into forms amenable to analysis.

The U-Engineer.com ESB fills this requirement with a formal engineering representation, specifically tailored for analysis, called Analyzable Product Model (APM) [Tamburini, 1999]. From this APM, analysis applications can extract the information they need, including product idealizations. Besides providing a single source of analysis information, the APM bridges the semantic and syntactic gap between design and analysis. The APM specification includes auxiliary graphical models to aid in the development, understanding and human communication of complex APMs. The APM approach also takes advantage of the product data exchange standard STEP AP210 to simplify the data mappings used to extract analysis parameters from the design information sources.

Finally, "Thick Clients" must be able to communicate back to the server to obtain access to the sophisticated computer-aided analysis tools provided by the ESB. The Common Object Request Broker Architecture (CORBA) is one possible approach[7] to provide interoperability among the wide array of hardware and software products an ESB would have to contend with today. CORBA is particularly useful for ESB applications because it greatly simplifies client applications by using an Object Request Broker (ORB) to establish the client-server relationships between objects. Using an ORB, a client can transparently invoke a method on a server object, which can be on the same machine or across a network. The ORB intercepts the call and is responsible for finding an object that can implement the request, pass it the parameters, invoke its method, and return the results. The client does not have to be aware of where the

[7] At the time of this writing there are really only two other potential technology choices for interoperability and both are tied strongly to particular vendors. DCOM is a Microsoft-centric technology that mimics CORBA relatively closely. Enterprise Java Beans, evangelized by Sun, are not an exact replacement for CORBA type applications, but do provide a subset of their functionality.

object is located, its programming language, its operating system, or any other system aspects that are not part of an object's interface. In so doing, the ORB provides interoperability between applications on different machines in heterogeneous distributed environments, allowing the ESB to service the widest array of clients.

These components are illustrated in Figure 3-7 below. The executable code includes a GUI that is used by the Manufacturing Personnel to control the analysis and input boundary conditions and other manufacturing-specific information. The APM included in the thick client houses the neutral product data, whether it is in STEP, GenCAM, or a vendor-specific format. The executable code includes CORBA classes that enable the client to send the abstracted analysis information over the Internet to the ESB Servers, and receive back the analysis results.

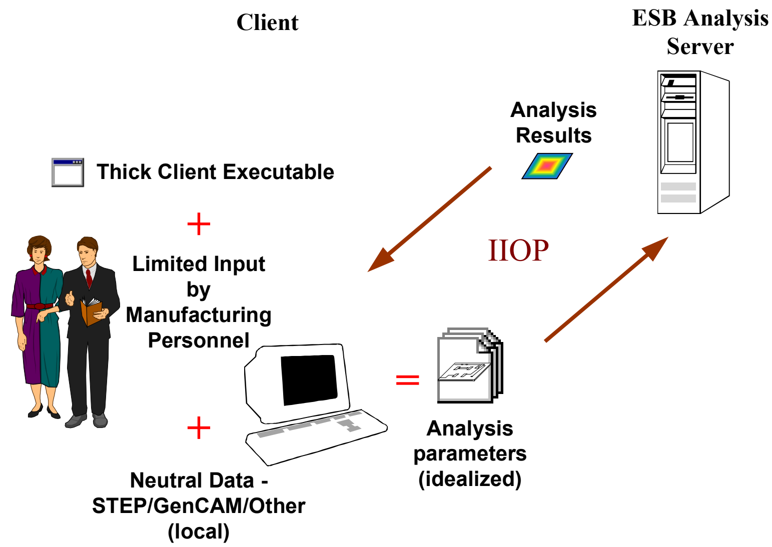

Figure 3-7: "Thick Client" Data Flow

Note also that ORBs also provide flexibility for the server-side development. They let programmers choose the most appropriate operating system, execution environment and even programming language to use for each component of a CAE system under construction. More importantly, they simplify the integration of existing components, such as Finite Element programs. In an ORB-based solution, developers simply model the legacy component, then write "wrapper" code that translates between the standardized ORB bus and the legacy tool interfaces.

3.2 Summary Overview of ESB Technologies

The following table lays out the main tradeoffs inherent in the technologies described above. One tradeoff is greater security versus greater cost, which balances server and local based processing. A second trade off is greater flexibility versus increased complexity, which is seen in the choice of thin/hybrid client web based tools compared to thick client tools.

Toolkit technology	Client code originates on	Client side data storage	Client side processing	Transfer protocol	Data transferred	Server side data storage	Server side processing	Client cost/ supply chain use
Web Form- Pure "Thin Client"	ESB Server	None or "cookie"- based	None	HTTP or HTTP/ SSL	Small number of idealized attributes and subset of design variables to host. Analysis results to client.	Analysis results	Common Gateway Interface (cgi) and Analysis tool operation	Low/ excellent
XML Based Web Form- Hybrid "Thin Client"	ESB Server	Full design information copied into HTML form (stays on client)	Javascript extraction of idealized parameters and critical design variables	HTTP or HTTP/ SSL	Small number of idealized attributes and subset of design variables to host. Analysis results to client.	Analysis results	Common Gateway Interface (cgi) and Analysis tool operation	Low/ Very good (requires 3.x or later model browser)
Web Augmented X Windows (TIGER)- "Thin Client"	ESB Server	None	None	HTTP/ SSL upload and X display	Full design information transferred to host over SSL upload. Analysis screens displayed to client.	Full design information	Common Gateway Interface (cgi), X Windows display, and Analysis tool operation	Medium / Good (requires X Windows client program)
Java CORBA client- "Thick Client"	Local client (installaton required) Server code at ESB	Full APM	Extensive processing in Java	CORBA IIOP	Idealized attributes and subset of design variables to host. Analysis results to client.	None	CORBA services and Analysis tool operation	Low / Very Good (requires Java installation)
Java CORBA client/server standalone	Local (installaton required)	Full APM	Java	N/A	N/A	N/A	N/A	High /Poor (expensive HW and SW)

Table 3-1: Comparison of Various ESB Toolkit Technologies from data storage, security, processing requirements, and cost perspectives.

3.3 ESB Technical Infrastructure Requirements

Several servers are required to build a robust, large scale ESB. For example, a high volume analysis bureau might include a Web server, an E-commerce server, analysis solution server(s), and possibly a materials database server. Smaller ESBs may combine these functions on to fewer servers.

Because the interaction between the ESB and its clients is essentially an interactive, design-oriented one (with a large number of files and graphics being exchanged) the ESB will have to have a high speed connection to the Internet, preferably a T-1 connection (bandwidth up to 1.5 Mbps) or greater.

Table 1 below is a rough indication of the computing infrastructure (1999 figures) necessary to set up and run a basic commercial ESB (low volume, single server).

Table 3-2: Computing Infrastructure for a Basic ESB

Item	Description	Approx. Cost
Computing Hardware	Workstation, OS, and Router • 256 MB of memory • 50 GB hard drive space	$25,000
Internet Connection	T-1 Connection (1.5 Mbps)	$18,000/year
Internet Software	Apache WWW Server or similar, Security Certificate • HTML 3.0 and up • SSL encryption	$300/year
Electronic Commerce Software	Accept on-line electronic payments	$1,000
Analysis Integration Tools	Dependent upon client needs	$10,000
Solution tools (e.g. ANSYS, *Mathematica*)	Dependent upon client needs	$60,000 plus $12,000/year
	TOTAL:	**$96,000 plus $30,300/year**

A full business case which includes personnel and other costs was beyond the scope of ProAM but will be useful for potential ESB providers. The self-service aspects of the ESB paradigm have a strong potential and could be a natural outgrowth of existing engineering consulting businesses.

In the sections below we detail some of the technical infrastructure required to provide the services on offer through the example U-Engineer.com ESB.

3.3.1 Hardware

U-Engineer.com currently runs on three servers. (See Figure B-1and Figure B-2 in Appendix B: U-Engineer Current Network Design.) We have a dedicated web server, a 266 MHz Pentium II PC with 128 Megabytes of RAM, which serves all the html pages and cgi programs linking to the back end. The web server also handles email addressed to u-engineer.com. We run a Sun Ultra 1 SPARC as our production analysis server, and we have a third 266 MHz Pentium II PC as a development analysis server.

3.3.2 Software

U-Engineer.com currently offers analysis modules supported by back-end processing from two main general purpose CAE Packages, Wolfram Research's *Mathematica*, and ANSYS. Both of these packages are installed on the both analysis servers, one running Solaris 2.6, and the other Windows NT 4.0. We also run the CORBA server software, Iona's OrbixWeb 3.0, from the analysis servers. (See Appendix A – CORBA Access Metrics.)

The Pentium II PC we use as a web server is running Red Hat Linux 5.2 with the Apache 1.3.3 web server.[8] We have found this combination to be fast and stable. Apache offers SSL functionality and basic authentication (via .htaccess and .htpasswd files), which we use to control access to the web-based analyses.

The Apache web server (which like Linux is open source) is growing increasingly popular as the web server of choice (see Figure 3-8 below).

Server Market Share

Source/Server	Sept, 1996	Feb, 1997
NETCRAFT SURVEY		
Apache	37%	41%
Netscape	14%	12%
Microsoft	8%	10%
NCSA	16%	10%
Other	25%	27%

source: eStats; Zona Research; Netcraft

Figure 3-8: Market Share of Web Servers in 1996, 1997

3.3.3 ESB Security

Securing the ESB servers is no different than the normal security routine for publicly accessible machines. This may include a firewall, filtering of particular protocols at the router level, and other security hardening steps. Since the web server has access and execution privileges on the analysis servers, particular attention should be paid to its security.

The publicly accessible ESB web server should not allow Server Side Includes (SSI) to enhance static HTML pages. Hackers can insert malicious SSI instructions into a field that will be evaluated as a HTML field by the web server, allowing the execution of commands locally on the web server.

Hidden HTML tags should not be used to transfer any e-commerce information, such as the cost of the analysis, to the server. Backend processing (such as a database lookup) should handle the e-commerce aspects, since the HMTL can easily be edited and the modified page submitted to the server.

Connections to the ESB server needs to be encrypted since the traffic traverses the Internet. Both web server traffic and connections from CORBA should be protected with a tunneled solution such as SSL. SSL, with its reliance on a digital ID for the server, offers the additional benefit of assuring the client that it is connecting to the actual ESB server, not a hacker masquerading as the host. In addition to tunneling, further authentication/authorization of the client to the server may be required, particularly if the ESB chooses to differentiate between different membership levels, or charge on a per-use basis (rather than a time based subscription).

Most security-conscious firms are particularly interested to learn that CORBA connections are uni-directional, i.e. the client always makes the establishing connection and not the server. Another common client requirement is for a TCP application proxy for any outbound (from the client to the server) TCP connections and a TCP application proxy with strong authentication for any inbound (from the server to the client) connections. However, because CORBA client-server communications are initiated on the TCP port for the ORB daemon, and then moved to a new TCP port for object level interactions, it is

[8] One note of caution: Linux is a variant of Unix, and so the file system is case-sensitive. This can cause missing graphics and hyperlink problems if you are migrating HTML files from a Windows NT server, since NT is not case-sensitive. Attention must also be paid to the format of cgi files. We learned this the hard way when several Perl scripts were ported from NT without converting the text format to the Unix format. Apache reported that it could not find the requested script when we attempted to run it from a web page, although the script could be run through a Perl interpreter manually!

difficult for most general firewalls to provide application proxies for CORBA. If filtering of IIOP packets must be enforced, many CORBA vendors offer IIOP-aware firewalls, such as Iona's 'WonderWall' product.

3.3.4 Analysis Applications

The Engineering Service Bureau is differentiated from pure Application Service Providers in that the ESB is directly providing information rather than providing an application that can be used to derive information. An ESB customer need not interact with a standard, off-the-shelf analysis package directly while using analysis modules. Thus, unlike Application Service Providers, ESBs have a great deal of latitude in selecting the analysis programs used to provide the functionality customers want. Selection of analysis programs may be made on a multitude of criteria, including purchase cost, ease of programmatic access to features, computational efficiency, output quality, and others. Use of an object-oriented legacy wrapper technology, such as CORBA or Enterprise Java Beans, will allow a wide variety of tools, from Finite Element programs (such as ANSYS and Abaqus) to general purpose mathematical tools (such as MAPLE, MATLAB, and *Mathematica*) to be integrated in a cohesive system where each component is used where it is most effective.

3.3.5 Neutral Data and Analysis Integration Techniques

Building accurate analysis models takes time, and in the past was typically manually performed for each product to be analyzed. This raises both the response time and the ultimate cost of the analysis, preventing the early and frequent use of analysis through the product development cycle. To overcome this obstacle an ability to intelligently "routinize" analysis [Peak, et. al. 1998] is needed. This process parameterizes analysis to the extent that data read from neutral product and process files (such as International (ISO 10303 STEP), National (ANSI), Trade Organization (IPC), or Vendor-specific (Accel) neutral files), combined with Artificial Intelligence (AI) techniques, can provide highly detailed analysis modules. The design-analysis integration (DAI) techniques underlying this methodology are covered in [Peak et. al., 1999], [Tamburini et. al., 1996], [Tamburini et. al., 1997] and [Zhou et. al., 1997]. An ESB using such analysis integration techniques can thus minimize the need for specialist intervention and provide rapid response, high accuracy, and lower cost analysis by reading necessary information from neutral product and process files (e.g. STEP and GenCAM standard files).

U-Engineer.com has focused the design analysis integration techniques on the class of problems termed *routine analysis* - the regular use of established analysis models in product design. Based on the multi-representation architecture (MRA) design-analysis integration strategy, this process creates catalogs of product model-based analysis models (PBAMs) - analysis modules that associate design data with analysis models to obtain results in a highly automated manner. The creation and use of these analysis catalogs is illustrated in Figure 3-9 below.

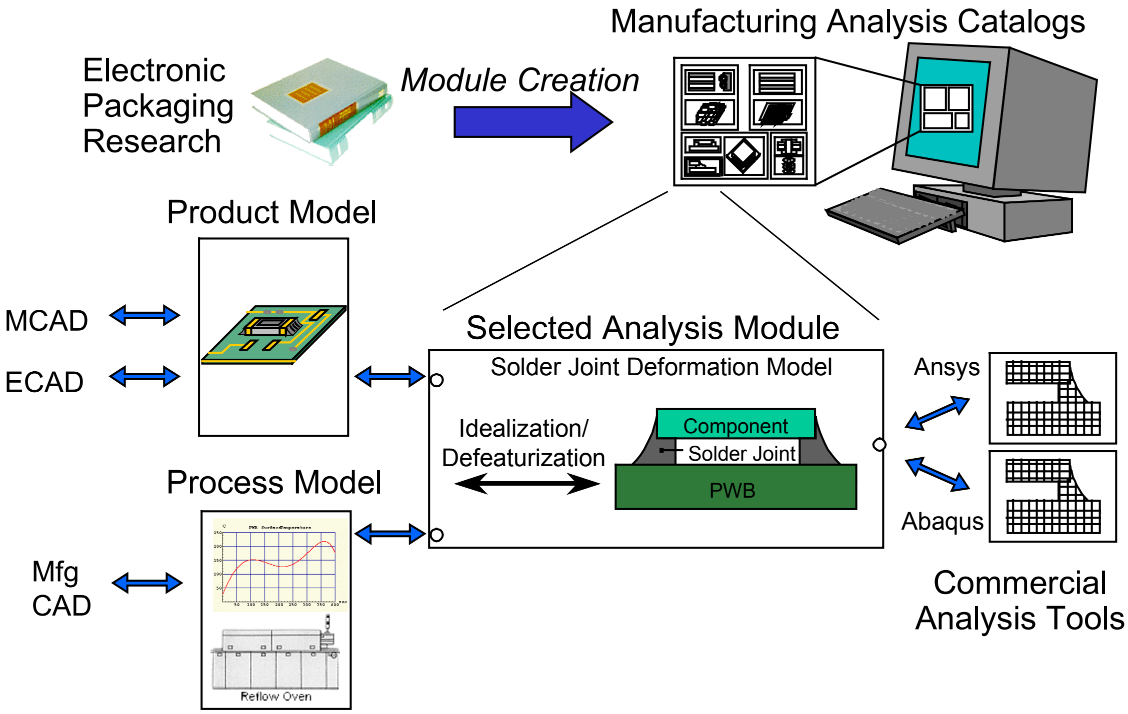

Figure 3-9: Neutral Data Sources in Analysis Integration Framework

From an infrastructure point of view, this approach means that the ESB must be able to accept and analyze product and process files which can be large (e.g. 25 megabytes for some STEP files) from an Internet point-of-presence. The TIGER and ProAM projects applied these techniques to drive self-serve thermo-mechanical PWB analyses from STEP AP210 data files exported from Mentor Graphics ECAD tools. [Peak et. al., 1997, EIS Lab, 1999c]

Increasing pressure to bring products to market faster means that the traditional bottleneck in the contract consulting/analysis business- the transfer of product information from the designers to the analysis provider- is fast becoming the areas where 'Standards savvy' ESB companies can differentiate themselves from the global consultancy pack.

3.4 ESB Software and Service Delivery Mechanisms

ESBs are commercial entities, and in the end they must have both a distribution channel and a method for processing a revenue stream. Since ESBs are Internet-based from their inception, it is sensible to consider how the Internet may be used to both distribute software and collect revenue. Assuming that most ESBs will include a range of complementary client-side technologies, from "thin-client" web-based services to "thick client" access, this task consists of a number of inter-related sub-tasks:

a) Methodology to charge for the services and thick clients

b) Delivery mechanism for thick clients (install from the web)

c) Procedures to maintain thick clients (live update)

Charging for web-based services is becoming relatively straightforward – access to the cgi-generated pages can be restricted, and the appropriate password or digital ID given to customers who submit payment information. Thick client solutions require an integration of payment and delivery mechanism, such as the secure Internet distribution offered by InstallShield's InstallFromTheWeb product [InstallShield, 1999].

InstallFromTheWeb works by the inclusion of an e-commerce digital ID with the software it distributes over the Web, which in addition to verifying payment, creates a digital signature that assures customers that they're receiving software from the ESB, and that it hasn't been tampered with. InstallFromTheWeb 3.0 provides end users with a combined download and installation process in one simple step [Cybersource, 1999]. As illustrated in Figure 3-10, an on-line software purchase enables a customer to launch the InstallFromTheWeb client. This client retrieves a "ticket" (digital certificate) for the product purchased. This certificate is presented to the CyberSource download server as proof the customer is entitled to receive the product. If the authentication is valid, the InstallFromTheWeb client begins a download.

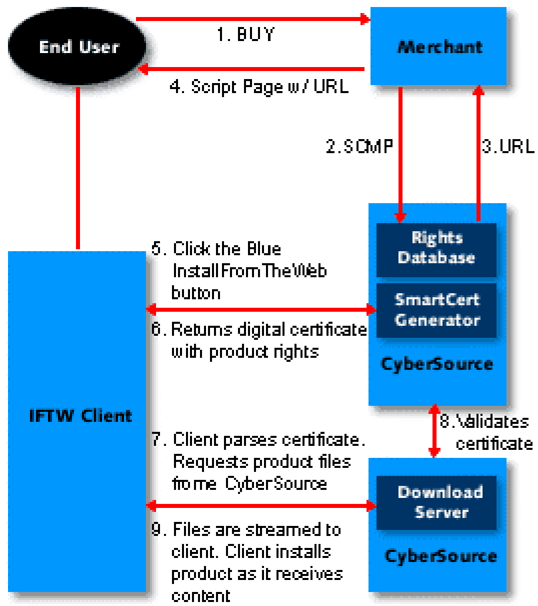

Figure 3-10: InstallFromTheWeb Payment Process

Maintenance of thick client code is also possible in a similar manner, although special care is needed to update code that is currently executing[9]. If, however, an ESB will charge for the upgrade, not all customers can be expected to upgrade immediately after a new release. The ESB will therefore have to remain "backwards compatible" on the server side for several client releases.

4 ESB Client Technologies

An Engineering Service Bureau client must be able to connect to the Internet for access to the graphical interfaces controlling the analysis tools and ease of electronic data transfer. This requirement is of substantial benefit to the SME, since by acquiring the technology to access a single service (the Internet) they gain access to a worldwide marketplace of competing Engineering Service Bureaus. This is in contrast to other approaches which champion separate networks for each service, such as the Value-Added Networks (VANs) espoused by some Electronic Data Interchange (EDI) software vendors, or the proprietary networks for video conferencing. While these approaches have merit for highly capitalized companies, SMEs must carefully examine the value of each dollar spent. Internet access is also attractive to cost-conscious companies because it provides many other benefits to the SME as well, and is increasingly being viewed as a standard method for customers to get in contact with a company.

Internet browser-based 'Thin Clients' [Korzeniowski, 1996] require the least bandwidth and provide the interactivity and graphical displays required for effective analysis. In TIGER the SME accessed the U-Engineer.com ESB from his PC via a web browser and an X Windows emulator. The

[9] For example, Microsoft operating systems (currently approximately 90% or better of the client OS market) lock files which are currently in use, yielding a "sharing violation" when attempting to directly overwrite them. The Microsoft utility 'Inuse.exe' can be used to replace files that are currently in use by the operating system. See http://www.microsoft.com/SYSPRO/inside/6-7-99.htm#inuse.

ProAM version of this toolkit dispensed with this need by utilizing client side interfaces based on technologies such as HTML, JavaScript, and Java.

A recent World Wide Web survey [GVU 1999] of over 5,000 individuals revealed that 66% of users are surfing the Web using a 56kbps modem or slower. The breakdown was as follows:

33k modem or slower	35%
56k modem	31%
128k ISDN line	6%
1 Mbps (T1 line) or faster	28%

Table 4-1: GVU Survey of World Wide Web Access Speeds [GVU, 1999]

Our experience has been that a 28.8 kbps dial-up Internet connection is workable for utilizing the GUIs to manipulate the product and process information and display the analysis results. As a greater percentage of the market adopts 56 kbps modems or other high bandwidth technologies such as cable modems and DSL, remote tool use will become comparable to local use, since response times will be dominated by calculations running on the ESB analysis servers. Hence we argue that the level of bandwidth currently purchased by SMEs for business purposes will be perfectly adequate for accessing ESB-type services.

5 Value-Added ESB Services

ESBs may provide a wide array of services, ranging from AI-based heuristic evaluations such as process-specific design checks or design-for-manufacturability (DFM), to discipline-specific numeric computations, such as thermo-mechanical analysis. Rather than having one ESB offer all these services, we envision multiple ESBs, each operating as a Center of expertise, with a healthy competition between bureaus. While there will be several differentiating factors between centers (including user assistance and documentation, ease of use of the tools, etc.) ultimately their service will only be as good as the analysis modules they provide.

5.1 Analysis Module Catalogs

One of the services an analysis-focused ESB should provide is a catalog of diverse analysis models with differing levels of complexity for the product domain of interest (Figure 3-9). These might include: 1) models "widely accepted" in the industry (i.e. Electronic Packaging models documented by the Institute for Interconnecting and Packaging Electronic Circuits [IPC-D-279]) 2) models which have been published in respected journals or other literature 3) internally developed, proprietary models which are based on the above and experimental data. Normally, the higher complexity models such as FEM produce higher accuracy results but are more expensive to run and develop. We envision the high accuracy models being used as a verification of designs after iterative improvement using lower cost, simpler analysis modules. To this end, the ESB should provide comparisons of their various models, and guides to assist users in making these cost/performance decisions. These comparisons should include sensitivity studies, so that ESB customers can assess the impact uncertainties in the input parameters will have on the analysis result.

5.2 Ergonomic Domain-Specific Analysis Tool Wrappers

It is important to realize that many of the ESB clients will be only occasional users, connecting to study the unusual or cutting edge product configurations they are asked to manufacture. An important feature of the provided services therefore is that the ESB needs to wrap solution tools in such a way as to shield the clients from the gory details of the tools. For the sake of users with a greater understanding of the tools involved, it is probably still useful to make available suitable intermediate analysis results (screens showing analysis variables of interest, output files from tools such as ANSYS, graphical representations of analysis results, etc.). However, the mapping from the product domain (PWB attributes such as length, width, and height) to the analysis domain (mesh size, density, and element type in the case of FEA) must be handled automatically. Similarly, the reverse mapping, where for example the dimensionless nodal displacement of element 843 needs to be converted back into a PWB board warpage measurement, must also be handled transparently. In addition to wrapping the inputs and outputs of analysis tools, the ESB must also provide a means (hopefully a GUI) to enable SME clients to manipulate the product description in some fashion as they "do design".

5.3 Analysis Data Repositories

There is a wide arena of information needed to support analysis, and another value-added service of the ESB is to provide this data. In the structural and thermal analysis of electronic packaging, this requires catalogs of mechanical and thermal properties. Again, varying levels of complexity are needed, from constant, elastic, temperature-independent properties, to nonlinear, strain and time dependent, temperature dependent properties. In general, "typical values" for many of the analysis parameters of interest are useful. For example, a (Prime) design engineer performing a quick verification of how a particular design will fare in the manufacturing environment will need 'typical' values for various stages of the manufacturing process. A (SME) manufacturing engineer making process design decisions concerning how to manufacture a specific part will probably want to use vendor-, company-, or factory-specific values for these same parameters however. Thus, all properties that the ESB provides must have the provision of being over-ridden by user input, and where possible should be capable of being extracted from standard files, to avoid error-prone manual re-entry.

6 Customer Support Requirements

One of the most important principles for the beginning ESB to understand is that proper education is a fundamental part of analysis. By making analysis easier to use, there is a real danger that the ESB will also make analysis easier to misuse! This danger can only be mitigated by appropriate education of the ESB's customers. This training may take many forms. Appropriate on-line documentation, whether in the form of web pages or context sensitive help, must be provided. A Frequently Asked Questions (FAQ) document should also be available, since it can be invaluable in guiding a new user through complexity of interacting with analysis modules while illustrating their boundaries and limitations. Finally, ESBs specializing in particular problem areas, such as PWB warpage, may give short training courses or seminars on how to correctly apply the analysis modules they make available.

6.1 Analysis Documentation

One very important value-added service the ESB can provide is that of an information clearinghouse. ESB personnel can monitor the diverse channels for the distribution of analysis-related information, such as professional conferences, refereed journals, industry consortia, and experimental studies. The ESB collects, filters, and packages this material to make it available for regular use by industry. An important service, then, is the analysis documentation, which explains the origins of the model, particularly the fundamental physical behavioral models it relies on, typical application examples, and limits of accuracy. The extent of experimental validation and correlation to physical product behavior are also important information of interest to the end-user who may decide to purchase the result of this particular analysis on his product. Some indication of the complexity of the analysis model is probably also useful; although the free-market per run price will also provide a relatively accurate gauge of the computational cost of the analysis.

We have found it useful to divide the analysis documentation into a hierarchy of related "families", based on the type of behavior the analysis models describe. Thus, for example, we have one family of models describing Plated Through Hole thermo-mechanical behavior, and another describing PWB/PWA warpage. At the family level of aggregation we document the type of problem the analyses represent, typical causes and curative measures, and compare and contrast the various individual analysis modules available.

6.2 Result Interpretation Guidelines

We believe that analysis results should be relatively rich in terms of the information they return to the end-user, since this ancillary information can be important in the event some aspect of the idealization performed to carry out the analysis does not strictly hold. For example, the ultimate strength of copper may be idealized as having a single value exactly equal to 260 MPa, which would provide a passing grade to a stress analysis that returned a result less than this value. Providing the value of this result, such as 245 MPa, rather just than a "go-no go" answer, is useful in cases such as where a batch of copper is purchased with a slightly lower ultimate strength, or where the maximum applied load is at an elevated temperature. To augment and elucidate the analysis result data set, the ESB needs to provide interpretation guidelines that assist in relating analysis results back to the physical product in question. Thus, they will include broad coordinated observations and insights that explain the limits and qualifications of various aspects of the analysis that cannot be easily processed by computer. (Relatively simple if-then type rules, such as checking a linear analysis module's output for violation of the assumption of linearity, should be implemented in an automated fashion. See Section 6.4, Further Work, below.)

The next logical question is how to best provide these interpretation guidelines to ESB consumers. The following two subsections detail the methods used by us in supporting the customers of U-Engineer.com.

6.2.1 Phone/On Site Consultation

The first time ESB customers begin to use the automated analyses, they will typically have questions concerning the information that the analyses return. Phone consultations provide an opportunity for ESB personnel to train customers how to interpret and filter analysis results by example. Such expert advice could include:

- Providing experienced-based correlation, by which we mean that although a particular model may not have a quantitative body of statistically valid data to support it, the model has been used with a high degree of success in a range of similar problems in the area of interest.
- Analysis input assistance, i.e. aiding the selection of proper inputs to a particular analysis or group of analyses. This assistance includes describing how and where analysis boundary conditions can be derived from aspects of the product's environment, either use environment or the manufacturing processes used to create the product.
- Guidance as to when certain material properties should be reduced or replaced with "effective" properties under certain conditions (such as microscopic quantities).
- Recommendations for choosing the right idealizations depending on the purpose and intent of the analysis.

It is helpful to provide pointers to web-based documentation (see below) to enable clients to further study and clarify judgements made by ESB analysts. Such phone contact can also be beneficial to the ESB, as they learn of new features and capabilities their customers are interested in having access to.

6.2.2 Web Documentation

We argue that web-based documentation is extremely important for the Internet-based Engineering Service Bureau. In addition to providing necessary background informing clients of the basis of the analyses provided, they effectively perform advertising functions. As the documentation pages are indexed by the various search engines on the web, they perform as focused advertisements for the specific functionality described. Someone searching the web for information concerning "PWB warpage" for example, will find the documentation pages for various modules dealing with warpage returned with a high degree of relevance.

6.3 Custom Consulting

In addition to delivering these capabilities to SMEs on a self-service basis for highly automated routine analyses, a commercial ESB should also offer consultation-based full-service analyses for challenging new problems. Potentially, if appropriate intellectual property arrangements have been made, these new analyses eventually become repackaged for self-serve analysis as the industry as a whole evolves. For example, a custom analysis of an ambitious new application using Flip Chips could eventually become a routine analysis as this technology becomes more of a standard packaging option. This potential inter-relation between custom consulting work and self-serve analysis means that commercial ESB operations will probably most successfully evolve from existing engineering consulting businesses over time. The section below details some of the infrastructure that has to be developed to allow this to happen.

Offering the more traditional custom consulting services may also help the ESB when dealing with lenders and creditors, as the aspects of self-serve Internet-based analysis are novel, and may not be as palatable as lines of business which have existed for much longer periods of time.

6.4 Further Work

It is possible, and indeed probable, that some ESB users will have only a trade school education. To help ensure valid analysis results and guard against unintentional analysis misuse, some form of knowledge capture and subsequent execution is required, documenting that this analysis module works for these products under these conditions. Although we recognize the need for such automated results and analysis assumptions checking, unfortunately the realization of this capability is still largely a research issue.

Until automated analysis toolkits such as *XaiTools-PWA/B* [EIS Lab, 1999b] have animated context-sensitive assistants similar to those found in Microsoft's Office 97 business software suite, good documentation and good interface design are the minimum requirements for safe, reasonable automated analysis.

7 Summary

We have introduced the Internet-based Engineering Service Bureau (ESB) concept as a multifaceted means of empowering SMEs with advanced self-serve analysis capabilities. An ESB provides catalogs of highly automated plug-and-play analysis modules in a useable form along with necessary supporting information such as material properties. By wrapping the tools that carry out the analysis in interactive product-specific Graphical User Interfaces (GUIs), ESBs remove the need for the end user to know how to operate complex analysis software. The use of rich product models contained in neutral standard files such as STEP ISO-10303 files reduces tedious manual data entry and enables analysis automation to a greater extent than is currently practiced. XML-compliant product standards, such as GenCAM^{sm} XML, enable the benefits of the web interface (ubiquity, portability, and comprehensibility) to be realized without sacrificing data integrity or automated data processing. By employing such design-analysis integration techniques and specializing in the volume utilization of expensive analysis tools and associated intellectual investments, ESBs are able to lower the cost of sophisticated analysis and substantially expand the roster of potential users.

The use of the client-server model of computing allows product data to remain distributed across the web, rather requiring it to be concentrated on ESB servers. Uploading only idealized product parameters offers a significant reduction in bandwidth requirements and improves the intellectual property protection for ESB customers.

When analyses such as these are performed throughout a supply chain by the organizations with the most knowledge of the product and process in question, we hold that a substantial improvement in product quality can be achieved. The greater level of detail known about the product and process mean that SME-run analyses often are more precise than those which can be performed by the Prime.

The Engineering Service Bureau paradigm is beneficial to all involved. By creating an Application Service Provider market, ESBs open the door to a relatively untapped CAE user market

(SMEs), increasing CAE tool sales. For SMEs, accurate analysis capabilities provide an important value-added service that can improve manufacturing yields and product costs, providing a competitive advantage. Primes and their customers gain by having potential problems caught earlier in the product realization process, resulting in higher quality products. Primes may also be interested in an Intranet-based Engineering Service Bureau as a supplement to locally installed tools for occasional users or users with highly repetitive analysis needs, freeing up specialized analysts to work on more challenging problems.

8 Acknowledgements

Development of U-Engineer.com and its underlying technologies has been funded by various organizations including the following:

- While under Defense Advanced Research Projects Agency, the National Electronic Commerce Resource Center (ECRC) program funded the SCRA-led TIGER project, in which Georgia Tech pioneered the Internet-based self-serve engineering service bureau (ESB) concept with its product data-driven analysis distinctive.
- Under U. S. Department of Defense Joint Electronic Commerce Program Office (JECPO), the National ECRC program funded Georgia Tech to take the TIGER demonstration capabilities and transform U-Engineer into a pilot commercial ESB. This effort, the ProAM project, was overseen by the Atlanta ECRC as a subcontract under Concurrent Technologies Corporation, with the Army Aviation and Missile Command (AMCOM) as its primary stakeholder.

In addition to full-service and development efforts, current support and funding is provided by the following:

- The Atlanta ECRC continues to support U-Engineer.com with technical support as a service to its DoD and SME clientele.
- With an emphasis on warpage behavior, Electronic Packaging Services, Ltd. Co. (EPS) is donating funds to further research and applications of ESB technology to electronic packaging thermomechanical behavior.
- The Georgia Tech CALS Technology Center continues to provide management, computing, and infrastructure support to U-Engineer.com.

9 References

Arar, Yardena (1999). “Web-Based Utilities and Productivity Apps Spell Lower Prices and Easier Upgrades,” PC World Online April 1999, Internet WWW Page, at URL:
http://www.pcworld.com/current_issue/article/0,1212,9792,00.html (version current at April 8, 1999).

Briody, Dan (1999). “Software licensing in limbo,” InfoWorld, Vol. 21, Issue 28, July 12, 1999. Available online at URL:
http://www.infoworld.com/cgi-bin/displayArchive.pl?/99/28/t01-28.1.htm (version current at June 25, 1999).

CyberSource (1999). “InstallShield Adopts CyberSource Technology For Secure Digital Distribution,” Internet WWW Page, at URL:
http://www.cybersource.com/iftw/ (version current at June 25, 1999).

DoD 5200.28-STD, or the "Orange Book". Department of Defense Trusted Computer System Evaluation Criteria

Engelmaier, Werner (1996). "IPC-D-279, Design Guidelines for Reliable Surface Mount Technology Printed Board Assemblies," Institute for Interconnecting and Packaging Electronic Circuits, Northbrook, IL.

Engineering Information Systems (EIS) Lab, (1999a) "XaiTools Users' Guide," available on the web at URL:
http://eislab.gatech.edu/tools/xaitools/ (version current at June 30, 1999).

Engineering Information Systems (EIS) Lab, (1999b) "XaiTools PWA/B Users' Guide," available on the web at URL:
http://eislab.gatech.edu/tools/xaitools/ (version current at June 30, 1999).

Engineering Information Systems (EIS) Lab, (1999c) "ProAM - Product Data-Driven Analysis in a Missile Supply Chain," available on the web at URL:
http://eislab.gatech.edu/projects/proam/ (version current at June 30, 1999).

Fulton, R.E.; Peak, R.S.; Tamburini, D.R.; Scholand, A.J.; Zhou, W.X.; Cimtalay, S. (1997). "Enabling Advanced Prime-Supplier Collaboration in TIGER Using STEP Product Model-Driven Analysis," CALS Expo USA, Orlando, Track 2, Session 4.

Graphic, Visualization, & Usability Center's (GVU) 10th WWW User Survey. (1999). GVU Center, College of Computing, Georgia Institute of Technology. Internet WWW Page, at URL:
http://www.cc.gatech.edu/gvu/user_surveys/survey-1998-10/ (version current at June 25, 1999).

Hummingbird Communications Ltd. (1999). "The Gold Standard in PC and Thin X Connectivity," Internet WWW Product Page, at URL:
http://www.hummingbird.com/products/nc/exceed/index.html (version current at June 25, 1999).

InstallShield, (1999a) "Secure Electronic Software Distribution," Internet WWW Page, at URL:
http://www.installshield.com/products/verisign/default.asp (version current at June 25, 1999).

McCarty, Meredith, (1999) "Worldwide Application Service Provider Review and Forecast, 1998-2003," International Data Corporation, IDC #W18664, March 1999.

Megginson, David, (1998). *Structuring XML Documents*, Prentice Hall PTR, Upper Saddle River, NJ.

Junnarkar, Sandeep (1999). "Pay-per-use app market blooming, benefits unclear," C|Net News Story, at URL:
http://www.news.com/News/Item/0%2C4%2C37410%2C00.html?sas.mail (version current at June 4, 1999).

Korzeniowski, Paul. (1996) "NCs: Coming soon to a desktop near you?" PC Week. Vol. 13, Oct 7,1996, pp.18-21.

McLay, Michael (1999). "Introduction to the GenCAM Standard," Internet WWW Page, at URL:
http://www.gencam.org/intro/ (version current at January 25, 1999).

Microstar (1999). "Ælfred," XML Parser provided on Internet WWW Page at URL:
http://www.microstar.com/aelfred.html (version current at June 30, 1999)

Miller, Jeremie (1999). "Xparse," Internet WWW Page, at URL:
http://www.jeremie.com/Dev/XML/index.jer (version current at January 25, 1999).

Petreley, Nicholas (1999). "Everything changes when you add a middleman to the software equation," Infoworld, July 26, 1999. Available online at URL:
http://www.infoworld.com/cgi-bin/displayNew.pl?/petrel/990726np.htm (version current at June 25, 1999).

Peak, R.S.; Scholand, A.J.; Tamburini, D.R.; Fulton, R.E. (1999), Invited Paper for Special Issue: Advanced Product Data Management Supporting Product Life-Cycle Activities, *Intl. J. Computer Applications in Technology*, Vol. 12, No. 1, 1-15.

Peak, R.S.; Fulton, R.E.; Nishigaki, I.; Okamoto, N. (1998). "Integrating Engineering Design and Analysis Using a Multi-Representation Approach," *Engineering with Computers*, Volume 14, Number 2., 93-114.

Peak, R. S.; Fulton, R. E.; Sitaraman, S. K., (1997) "Thermomechanical CAD/CAE Integration in the Tiger PWA Toolset." InterPACK '97, ASME, Kohala Coast, Hawaii.

Peak, R. S.; Scholand, A. J; Fulton, R. E.,(1996) "On the Routinization of Analysis for Physical Design." Application of CAE/CAD to Electronic Systems, EEP-Vol.18, Agonafer, D., et al., eds., 1996 ASME Intl. Mech. Engr. Congress & Expo., Atlanta, 73-82.

Pfeiffer, Ralf I. (1999). "XML Tutorials for Programmers," Internet WWW Page, at URL: http://www.software.ibm.com/xml/education/tutorial-prog/abstract.html (version current at January 25, 1999).

Ramesh, M.C., Fulton, R.E., and Duhig, J.J., (1995) "Approaches to Bringing Information Technology to Small Business Applications." Proceedings of the ASME Engineering Database Conference, San Francisco, December 1995.

Scholand, A.J.; Peak, R.S.; Fulton, R.E. (1997). "The Engineering Service Bureau - Empowering SMEs to Improve Collaboratively Developed Products," CALS Expo USA, Orlando, Track 2, Session 4.

Scholand, A.J.; Peak, R.S.; Fulton, R.E.(1999), ASME Design Engineering Technical Conference (DETC99), Las Vegas.

Tamburini, D. R.; Peak, R. S.; Fulton, R. E. (1996a) "Populating Product Data for Engineering Analysis with Applications to Printed Wiring Assemblies." Application of CAE/CAD to Electronic Systems, EEP-Vol.18, Agonafer, D., et al., eds., 1996 ASME Intl. Mech. Engr.Congress & Expo., Atlanta, 33-46.

Tamburini, D. R.; Peak, R. S.; Scholand, A. J., Fulton, R. E. (1996b) "Deriving a PWB Thermal Coefficient of Bending from Isotropic Material Properties of Constituent Components" Unpublished internal memo.

Tamburini, D. R.; Peak, R. S.; Fulton, R. E. (1997) Driving PWA Thermomechanical Analysis from STEP AP210 Product Models. 1997 ASME Intl. Mech. Engr. Congress & Expo., Dallas.

Tamburini, Diego R. (1999). "The Analyzable Product Model Representation to Support Design-Analysis Integration." Doctoral Thesis, Georgia Institute of Technology, Atlanta.

U-Engineer.com. Pilot Commercial and Demonstration World Wide Web Pages, Georgia Institute of Technology, 1999.
http://www.u-engineer.com/ (version current at June 25, 1999).

Verisign, Inc., Verisign World Wide Web Pages, 1997.
http://digitalid.verisign.com/

Zhou W. X.; Hsiung, C. H.; Fulton, R. E.; Yin, X. F.; Yeh, C. P.; Wyatt, K. (1997) "CAD-Based Analysis Tools for Electronic Packaging Design – A New Modeling Methodology for a Virtual Development Environment." InterPACK '97, ASME, Kohala Coast, Hawaii.

10 Glossary

AP210 - ISO STEP standard for PWA/PWB descriptions [ISO 10303-210]
APM - Analyzable Product Model
ASP - Application Service Provider
CAE - Computer Aided Engineering
CGI - Common Gateway Interface
CORBA - Common Object Request Broker Architecture
CRP - Customer Relationship Management
DAI - Design Analysis Integration
DARPA – Defense Advanced Research Projects Agency
DFM - Design for Manufacture
DTD- Document Type Declaration
ECRC – Electronic Commerce Resource Center
ERP - Enterprise Resource Planning
ESB – Engineering Service Bureau
GenCAM - IPC PWA/B Data Standard
GenX - GenCAM in XML
GUI - Graphical User Interface
HASL - Hot Air Solder Leveling
HTML - HyperText Markup Language
IIOP - Internet Inter Orb Protocol
IPC- formerly Institute for Interconnecting and Packaging Electronic Circuits
MRA- Multi Representation Architecture
ORB - Object Request Broker
PBAMs - Product model Based Analysis Models
PC - Personal Computer
ProAM - Product Data-Driven Analysis in a Missile Supply Chain
PTH - Plated Through Hole
PWA - Printed Wiring Assembly (circuit board and components)
PWB - Printed Wiring Board (circuit board only)
SGML - Standard Generalized Markup Language
SME - Small-to-Medium-sized Enterprise (supplier)
SSL - Secure Sockets Layer
TIGER - Team InteGrated Electronic Response
URI - Universal Resource Identifier
W3C - World Wide Web Consortium (W3C)
WAN - Wide Area Network
XML - eXtensible Markup Language

Appendix A – CORBA Access Metrics

1. The following tests were performed with XaiTools v0.3.0.1 (May 3, 1999 version). Results for the basic test suite (now called Basic2 test suite in *XaiTools* v0.3.2) were as follows:

	Total Processing	Processing Time per Call	CORBA/Connection Overhead per Call
a. With local Mathematica	36 seconds	1.24 seconds	N/A
b. With CORBA-based, via LAN (single user)	105 seconds	3.62 seconds	2.38 seconds
c. With CORBA-based, via 28.8K modem (single user)	210 seconds	7.24 seconds	3.62 seconds

- The number of calls to Mathematica (via CORBA) = 29
- These were run with the DOS Transcript window open/on top (makes things slower - but still useful for relative comparison)
- These are single user times (with no delay in processing, no queue at the server, and assumed typical network load)

2. Typical Mathematica input/output file size:
< 100K (most < 50K)

3. Typical FEA file sizes (for Ansys - based on 2D PWB warpage model):

input:	< 100K (most < 20K)	- preprocessor model
outputs:	< 10K	- extrema only
	100K-500K	- for each postscript graphics screen

Appendix B: U-Engineer Current Network Design

U-Engineer.com is currently[10] operating with the network setup illustrated schematically in Figure B-1 and Figure B-2 below. The Thin Web Client (Figure 3-2) uses an HTML browser to review documentation and information on U-Engineer.com's web server. When the client wishes to execute an analysis, the web browser uploads a response to an HTML form. The Web Server processes the form submission with a CGI (Common Gateway Interface) program written in Perl. This program contains templates coded for direct manipulation of the CAE tools themselves. Currently U-Engineer.com has scripts interfacing to *Mathematica* and ANSYS via Unix remote shells. Although not illustrated in the Figures below for the sake of clarity, the CAE tools return the desired results as HTML files to the web server for the client to access.

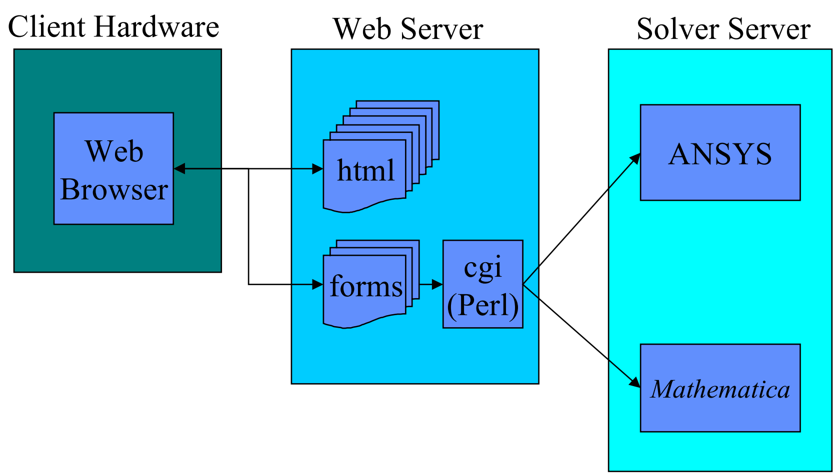

Figure B-1: Network Components for U-Engineer.com's Thin Web Client

U-Engineer.com's Thick Clients still rely on HTML based documentation and support information, so a web browser is still needed in this scenario. However, the thick client uses a layer of abstraction, namely CORBA, between the client code and the server CAE tools. This layer of abstraction, although increasing complexity somewhat, allows much greater freedom in supporting clients, since many implementation details are hidden from the clients. In Figure B-2 below both the thick client executable and the *Mathematica* executable are shown with a wide frame around their respective boxes to illustrate they are interfacing with CORBA wrapper classes rather than directly with each other.

[10] As of 7/30/99.

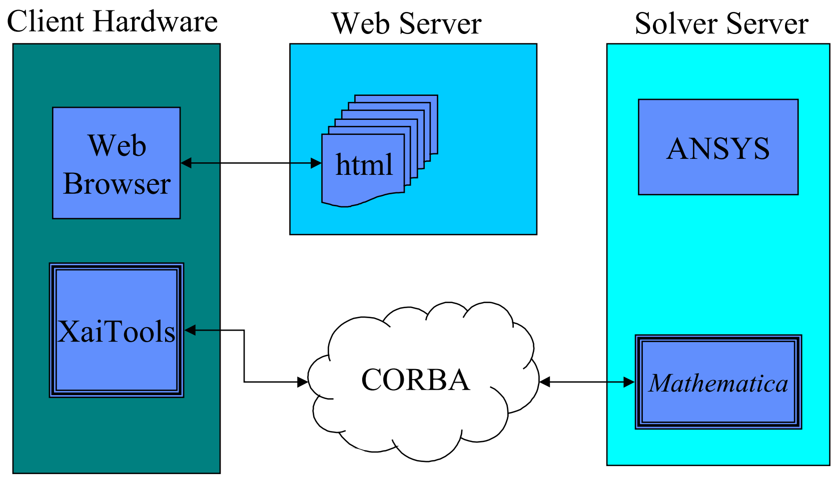

Figure B-2: Network Components for U-Engineer.com's Thick Client