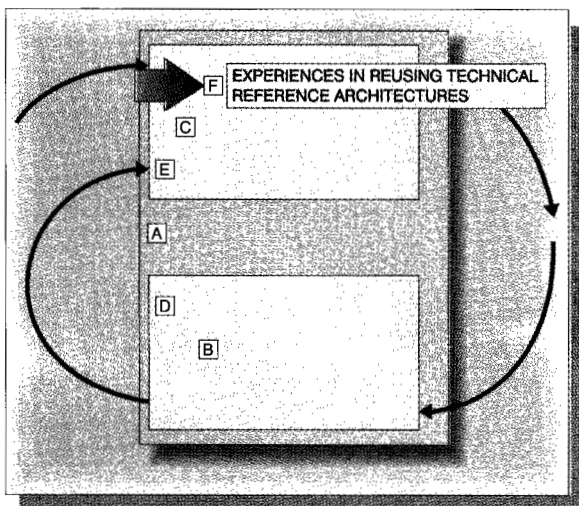


# Experiences in reusing technical reference architectures

by T. Harris  
J. W. Rothwell  
P. T. L. Lloyd



*Despite significant investment, the promise of reuse in IT (information technology) solution development—shorter delivery times, cost reductions, risk mitigation—has seldom been realized in practice. The difficulties of reuse at the front end of the projects have overwhelmed the potential benefits downstream. Enterprise Solutions Structure (ESS) is designed to provide a rigorous and powerful structure to facilitate the sharing of experience and IT architectural assets across IBM's solutions development community worldwide. The key question this paper seeks to answer is, can the ESS approach address this very real barrier and deliver the promised value? Described here by the consultants involved are a number of customer engagements in which the early ESS work was exploited. We show how relevant assets could readily be identified and the value they contributed to both our clients and IBM at different stages of the full engagement life cycle—at bid/proposal*

*time, in contract definition and scoping, and during the engagement—and the continuing benefit after the engagement. Finally, we outline some of the challenges yet to be addressed.*

Few people would argue that reuse of experience and “piece parts” is not a desirable objective. It is a standard engineering discipline in many areas. The promise of reduced time frames and lower costs and risks is an attractive one—indeed, with the growing complexity of today's IT (information technology) systems and the increasing business pressures to deliver them in shorter times, it could be regarded as an imperative. But it has proved an elusive goal. The significant costs of investing in standard, complete, and unambiguous ways to describe IT solutions are justified only if the intended experience sharing and reuse is achieved in subsequent projects and engagements. Many practitioners who have struggled with this will recognize that the key challenge is in representing the information in such a way that later projects can quickly identify relevant material and easily adapt it to their needs, without a detailed knowledge of the earlier project and without risking the integrity of the later ones.

©Copyright 1999 by International Business Machines Corporation. Copying in printed form for private use is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the *Journal* reference and IBM copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free without further permission by computer-based and other information-service systems. Permission to *republish* any other portion of this paper must be obtained from the Editor.

In an effort to overcome some of these problems, which were particularly acute when developing large-scale solutions for its largest clients (which we call *enterprises*), IBM in mid-1996 set up the *Enterprise Solutions Structure* (ESS) initiative as a way to bring an improved, asset-based approach to its solutions development. The paper by Plachy and Hausler<sup>1</sup> in this issue gives further background on the establishment of ESS. Two key objectives were: earlier availability to clients of solutions that address current business needs, and better integration within one client enterprise of multiple IBM solutions.

In order to provide a reusable solution, we must capture its architecture. By architecture, we mean the structure or structures that make up the software and hardware components, the externally visible properties of the components, and the relationships among them. ESS tackles the challenge of architecture reuse in two ways: first, in establishing a standard structure using standard semantics for the description of architectures and, second, by identifying and abstracting a constrained set of high-level architectural patterns, from successful recent solutions development projects, in problem domains with high priority for IBM solutions and services businesses.

The initial domains (subject areas) were identified as transactional, business intelligence, and collaboration, with important subdomains covering call center and mobile computing styles. In order to avoid the development of architectural "silos,"<sup>2</sup> all these high-level architectures (which we call *technical reference architectures*) themselves reuse standard architectures "under the covers," for example the Internet subdomain. The paper in this issue by Lloyd and Galambos<sup>3</sup> provides more information on how these technical reference architectures are structured and provides examples of content, while the paper by Youngs et al.<sup>4</sup> gives more detail on the semantics used.

Although neither of these two techniques—defining structures and defining standard content—represents innovations, and our library of asset content is at this stage far from complete, we will nevertheless show in this paper how bringing them together can provide real benefit to our clients and to IBM in saving time and money, in reduced elapsed times, and in a higher quality result.

In addition to using a standard structure and nomenclature, ESS uses a technique that provides a context for the reference architectures that form the base of

the reusable assets, allowing users to quickly locate and qualify relevant assets. Associated "design guidance" documents help to ensure the correct interpretation and use. Another technique is to hold design information at separate, but linked, levels of

---

**ESS provides real benefit to our clients and to IBM: saved time and money, reduced elapsed time, and higher quality results.**

---

abstraction that we call *elaboration points*. The ability to "drill down" from a contextual view, through logical, to physical architectural constructs is a powerful approach that supports both usability and the widest possible reuse. Currently we are using the terms *initial* to describe the architectural elaboration point with minimal constraints, *logical* for the set of designs that include network topology, including the differentiation of clients and servers, and *physical* for the stage where hardware architectures and operating systems are defined (though not the numbers of hardware engines required, for example).

While the original investment in this project was in support of IBM solution development, it was clear that early value could be demonstrated in IT architecture engagements carried out for our clients by IBM services consultants and professionals. In this context, ESS is not a product for sale or license to our clients. Rather, it provides reusable, proven architectural patterns available to IBM service professionals.

The first test of whether the "usability barrier" had been overcome and real value obtained was in 20 or so architecture consulting engagements, carried out in 1997 and the first half of 1998, using early versions of the ESS assets. We have selected three of these, which we believe to be representative and with which the authors were closely associated, to illustrate the use of the assets and draw some conclusions about the value—to both the client and IBM. ESS is most likely to make a real difference in large systems. The three selected engagements were with major organizations in North America—one is a bank and the other two are insurance companies.

Two of the organizations have agreed to be identified by name: State Farm Insurance Companies and CT Financial Services Inc. The third organization will remain anonymous. We refer to the organizations as Company A, State Farm, and Canada Trust in the remainder of this paper.

It is important to note that all these engagements delivered architectural designs, not code, and that they were related to infrastructure architecture designs, rather than specifically industry-dependent architecture designs (such as core banking). However, call center architectures were included in these engagements.

We anticipated that the assets, as they were designed to be used in the execution of engagements, would deliver significant value in the "solution build" phase. More slowly, we recognized that in fact the higher level (more conceptual) representations of the assets were powerful tools at the earlier stages of the engagement cycle—during the proposal phase and in clarifying the scope of the engagement. For that reason, in the main body of this paper we review each of these stages for each of our chosen engagements to illustrate our assertion that these technical assets can deliver value at each stage. We will show that results were delivered quicker, at lower cost, and that clients were enthusiastic about the ESS approach, in some cases adopting the techniques for use in their own organizations.

For the assets to have continued value, they must grow and adapt as experience and technologies mature, so this is a process of continuous improvement. Thus we should expect to increase the value as ESS becomes more pervasive, more experience has been harvested, and our solution development methods more explicitly invoke the assets. It should be noted that as engagements utilize the ESS assets as a "starter" set, the material they create tends naturally to be in the ESS structural format, and so it becomes much easier to recapture new improvements than with the more unstructured work of the past. In our concluding remarks, we will return to this vital topic.

### Engagement experiences

*For years we didn't know what we wanted. When we saw ESS, we knew what we wanted.*

—United States client, 1997

Our experience to date shows that the ESS asset-based approach has value both to our clients and to

IBM at multiple points in the life cycle of architecture-oriented client engagements:

- At bid or proposal time: The client selects a supplier, based on the supplier's response to the perceived client requirement. At this stage, the client is looking for the supplier to demonstrate understanding and capability, as well as value for money.
- In contract definition and scoping: It is vital that the client and supplier be able to articulate a clear common understanding of what exactly is to be delivered. In practice, this is often difficult. Client and supplier may have a different understanding of the context of the requirement, may well use different terminology to express it, and may unknowingly be working under different assumptions.
- During the engagement: The supplier uses staff skills, plus appropriate assets and techniques, to develop and deliver the contracted solution (in this case, an architectural definition).
- After the engagement: This can be viewed from two perspectives. From the point of view of the client, if the engagement has gone well, benefit will continue over and above the contracted work product. From the point of view of the supplier, valuable experience will have been gained, and this experience should be captured in a form that can provide benefits in future engagements.

Different benefits were obtained at each of these stages, and we review them in turn, based upon our experience in three specific engagements. These engagements span a period of almost two years, from late 1996 to early 1998, and during this time the ESS materials matured and were enhanced by harvesting concepts from successful work and putting them back into the asset base. The engagements are representative of many cases where the approach of using ESS assets for architecture and design engagements has been successful. As time progressed and the starter set of ESS assets improved, we were encouraged to observe that increased benefit was derived by the clients.

We show some of the actual design assets that were used in the engagements. At the time of use, these assets did not fully conform to the new IBM Architecture Description Standard,<sup>4</sup> and both the semantics and notation were of variable quality. For this paper, we have made minimal changes to the terminology in these charts (for example, we now use the term "node" to denote a logical or physical hardware platform), rather than bring them up to date with our latest standards. We feel this approach is

more honest, and at the same time supports our optimism about the potential benefits of the ESS approach. As we can demonstrate benefits with the assets in this immature state, we feel very confident that future improvements will yield even better results.

### At bid/proposal time

When an enterprise embarks on building a new IT architecture, which may be the precursor to an IT solution, this is not done lightly. Typically, the architecture represents a large investment and frequently is of strategic importance to the growth and profitability of the business. The enterprise must be confident that its chosen development partner brings the right breadth and depth of expertise to the project, understands very well the subject area represented, and is able to apply it in a way that markedly improves the chances of a successful outcome.

We found that the very tangible nature of the ESS technology templates was highly attractive to our enterprise prospects. It gave them confidence in IBM's experience in this area, and they could see how the assets would contribute to a fast and successful solution to their requirement.

**Company A.** In describing their business needs, Company A representatives stated that:

- They had adopted a new retail customer business strategy centered on knowing the customer as a client, rather than as a series of separate accounts (life insurance, pension, group health, mortgage, etc.)
- They needed to develop a new IT enterprise architecture to match the new business direction (but were not sure what was included in such an architecture, or how to structure it)
- They also needed immediate design and product selection advice to support an ongoing call center project. They needed advice on how to ensure that the call center would fit the new overall architecture.

By showing examples from ESS we were able to describe succinctly the various aspects that in IBM's view should be part of an enterprise architecture. By understanding the ESS structure and (at that time) partially completed architecture patterns, the client was able to see precisely how we could help meet the business need, and was better able to specify what that need really was.

Company A had just finished a business-related consulting project with another consulting firm. This had determined the business view (how they wanted to do business as a firm), and part of their view of applications (what key applications they should implement). They now needed to decide how to structure and deliver those applications.

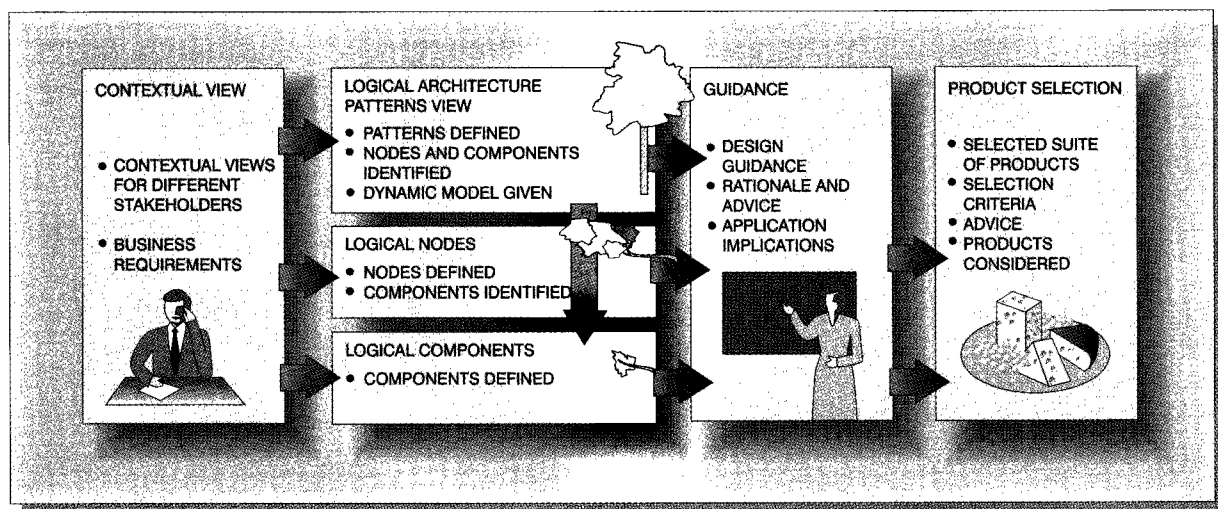
ESS assets were proposed to Company A as being able to answer questions such as: "How do I build . . . ?" "What should my technology infrastructure look like?"

To show what might be delivered for such an engagement, the ESS diagram in Figure 1 has been used successfully several times, first with this client. Although a simplified view of the real asset structure, it describes the basic structure in terms that are easily explained. It gives an idea of how the assets relate to the IT architecture method that was to be used in the engagement. Following are informal definitions of some of the terms used in the figure:

- Contextual views are a visual expression, in terms the user management can understand, of the business need (including the business of IT), or other aspects of the needs of the envisaged system.
- Architecture patterns (logical or physical) are visually expressed, with detailed walkthroughs (formal and informal use cases<sup>5</sup>), and indicate the topologies and technologies common to a class of solutions.
- Nodes and components are elements that make up the architecture patterns and are visible at a lower level of detail.
- Guidance can be applied at any level and is based upon good advice from lead designers and architects of successful projects.
- Products are finally selected based upon the physical architecture that has been elaborated from the logical architecture.

The logical architecture pattern consists of many parts. Two of the key parts used in this engagement were the high-level diagram showing the building blocks (nodes and components) that compose the basic infrastructure (the operational aspect of the architecture), and a textual walkthrough of a use case showing the flow of an interaction using that diagram. Figure 2 is a high-level diagram of the infrastructure for a call center, and Figure 3 shows the first part of a multipage textual walkthrough of what happens when a call comes in.

**Figure 1 Structure of ESS assets: Requirements to products with guidance at every level**



**Figure 2 The call center pattern: Phone call support to conduct business**

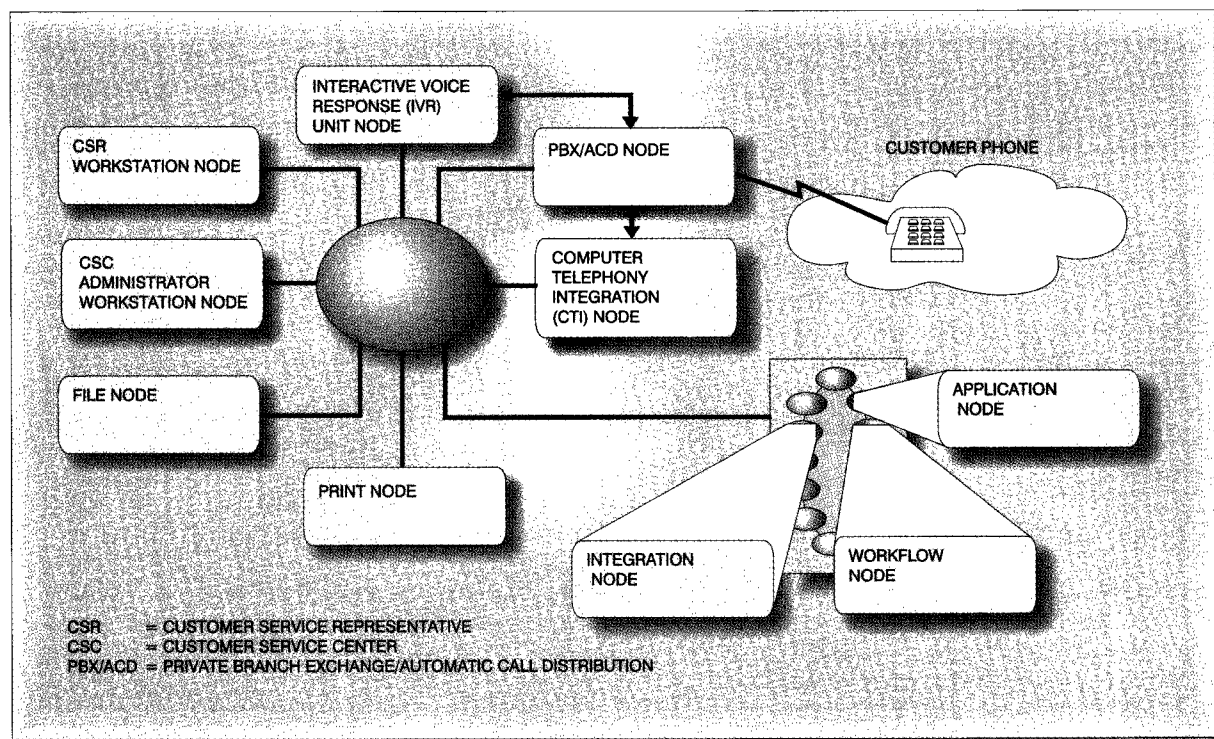
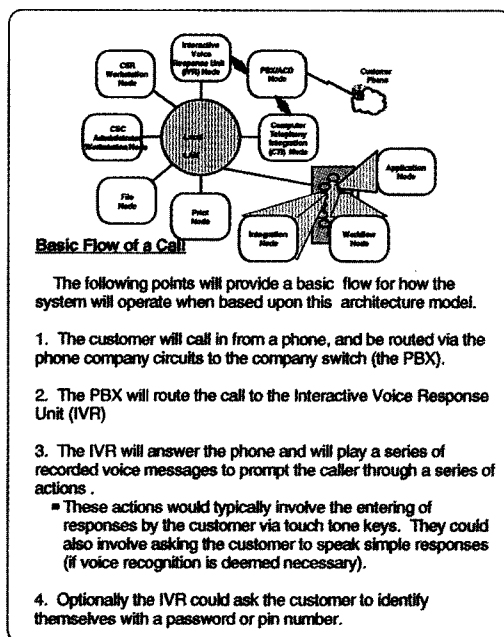


Figure 3 The overall flow of a telephone call traced through the nodes on the logical pattern

## To see how the Customer Service Centre is intended to work the overall flow of a telephone call has been traced through the Nodes on the Logical Pattern . . .



■ This will depend on whether the caller provides general information, at this time, or whether the caller is a customer wanting to request specific action on an owned product (such as asking for a Pension balance). Possibly a different phone number would be used to separate the types of calls.

— The user interaction required will depend on how the operation of the Business is defined

5. If a PIN number is required, then the IVR will need to send a message to wherever the PIN number validation is performed to ensure authentication of the caller. The architecture recommends that this validation be performed on a centralized authentication service (See Guidance Section)

6. Once the caller is authenticated then the IVR will play a series of messages prompting the caller to specify what action or information he/she would like.

■ The architecture indicates that these scripts should have a common sound across all product lines, and should have well thought out and designed interfaces to the customer (only a small number of choices to listen to at any time, a way to zero out and get help if needed, etc...)

7. As a result of specifying actions via the touch tone, the IVR may be instructed to execute transactions requiring interaction with Server functions on other System Building Blocks (Nodes).

■ Examples of such actions could involve

- the sending of product information (by mail or by fax)
- the IVR response giving current balance of a pension, principal left on a mortgage, etc...

The important aspect of this example is that the pattern includes a diagram of the major building blocks in the infrastructure (static view), and a walkthrough of how the flow of a call is handled (dynamic behavior).

The dynamic behavior can be shown as a textual walkthrough (for use in describing the behavior to users) as in Figure 3, or as a more formal component sequence diagram between the components running on these nodes (for use in describing the behavior to programmers).

Further detail was shown for one of the logical nodes and the components it contains to show the kind of information provided by the architecture in those areas. Samples are shown in Figures 4 and 5, which show the ESS descriptions (at the time of the engagement) of an interactive voice response (IVR) unit and some of the components placed on it.

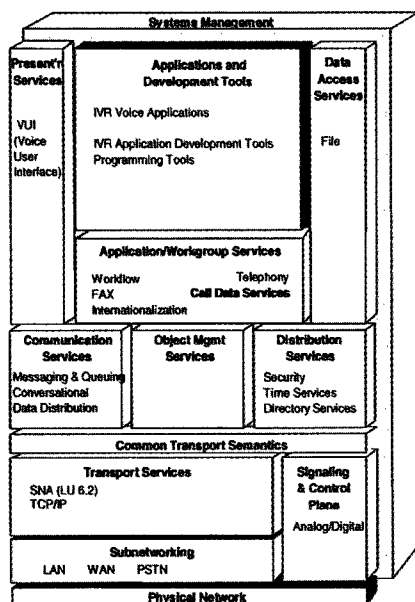
Note that the format of the diagram for the IVR node adopts the structure of the IBM Open Blueprint\*.<sup>6</sup> This structure had been used prior to ESS to provide a framework for defining technical components (services) that are needed.

The IVR is one of the nodes in the call center diagram shown in Figure 2, and the call data services represent components that reside on the IVR node. This example is demonstrating a “drill down” to more detailed levels—from the logical operational architecture pattern, to one of the nodes in that pattern, to one of the components on that node (see Figure 1). Obviously it would be useful if such assets were stored in a repository that supports links between levels of abstraction. The ESS assets are stored in a Lotus Notes\*\* database that provides this capability.

Company A representatives were shown examples like these at the time the proposal was made. Note

Figure 4 Example of part of a node description

## IVR Node



Components of interest are shown in red.

Call Data Services shown in blue are detailed in Figure 5.

### Interactive Voice Response (IVR) Node Overview

An Interactive Voice Response system allows businesses to provide callers with a number of services through voice interaction. The Components placed on the node enable the following services amongst others:

- providing information as selected by callers (e.g. on policies)
- completing business transactions (e.g. move money between pension funds)
- obtaining information from callers so that calls can be routed to the appropriate Customer Service Reps (CSRs)
  - (e.g. language preference, product to be discussed)

The IVR's components communicate with the end user through playing digital recording of voice segments, through text to speech conversion and listening through touch-tone processing and through voice recognition. In some cases an IVR may be connected to enhanced devices such as screen phones which allows for text displays (menu option displayed as text).

The IVR's capability to provide up-to-date business information, customer specific information and transaction type services depends on its components having real-time access to enterprise data and on it being able to initiate a transaction or workflow.

The IVR can be connected to the voice network (PSTN) directly or indirectly via an on-premise PBX. The IVR's components use the services of the CTI node for coherent routing of its call and call data, to the CSR taking the call, resulting in a more seamless and responsive service. (e.g. CSR taking the call will already have the customer identity and other information and will be able to anticipate the product or service to be discussed)

The IVR node may be implemented in conjunction with a CTI node or by itself. It can reside within a call centre environment or remotely such as at a branch, third party location, or home office.

that the details, such as the exact flow of a phone call in Figure 3, were not as important at this stage as the form of examples—the client could see the kind of deliverable material that could be expected.

Through the use of these tangible examples of the asset base, Company A representatives formed a good understanding of what the architecture would contain after an engagement with IBM consultants, and this contributed significantly to their confidence level and to the acceptance of the IBM proposal for consulting services.

**State Farm.** State Farm was embarking on a major system change to manage customer information at the enterprise level. The company was asking for help from consulting firms to determine changes needed in their architecture and system design.

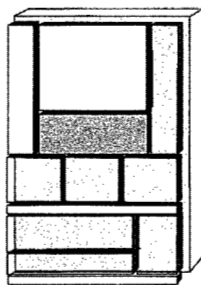
In early discussions, State Farm representatives expressed a need for the following:

- An “enterprise architecture” (although they had not defined what that really meant to them)
- A detailed vision of how to document an Internet and network computing architecture
- A unifying structure to bring together the many diverse individuals in the very large (several thousand person) IT department.

During the proposal, we showed a selection of the ESS assets as an example of what should be in an enterprise architecture. We followed the same basic approach as with Company A, except that the assets shown were specific aspects of what we were calling at that time the Network Computing (NC) Transactional Reference Architecture. It encompasses the

Figure 5 Call Data Services: Example of a component description

## Call Data Services



### Call Data Services

The Call Data Services component provides the application (CSR or IVR component) access to read and modify the call data associated with a call. The Call Data Services are enabled through a connection to a middleware component that provides an interface between the application component and lower level telephony. They are closely related to the Telephony Services component.

The call data may include

- Customer identification (e.g. Name, address, etc.)
- Customer attributes (e.g. gold service, new, etc.)
- Call history (e.g. this call has been handled by IVR and another CSR)
- Purpose of the call (e.g. customer selected eligibility inquiry on the IVR menu)
- CSR notation (e.g. reason for transferring a call to another CSR)
- Index to more customer information (such as a customer id)

Call Data Services provide the data associated with the call in a timely manner so that this data can be displayed to the CSR when a call arrives to the phone set. It can also provide the means for the CSR application to update the call data so that when the call is transferred or a conference is initiated, the second CSR has up-to-date call data.

If two or more CSRs collaborate in providing a service to the customer, the call data services can provide ongoing synchronisation of updates among the CSRs.

In order to support the virtual call centre, the Call Data Services must be available to the distributed CSR resources, whether located in a single site (single PBX) or multiple site (inter-PBX) across the WAN.

Call Data Services run on the following Nodes:

- CTI Server
- IVR
- CSR Workstation
- CSC Administrator Workstation

use of the Internet, intranet, and extranet to access corporate systems. One of these architecture diagrams represented a basic structure for Web-based computing (Figure 6). We show later that during the engagement this architecture was elaborated.

The accompanying walkthrough (Figure 7) was used to give insight into the NC transactional pattern that was being proposed. While the call center walkthrough shown earlier was directed toward an understanding of how external business events affect the system, this walkthrough focuses more on some of the "under-the-covers" behavior of the system, and was of interest mainly to the State Farm IT community.

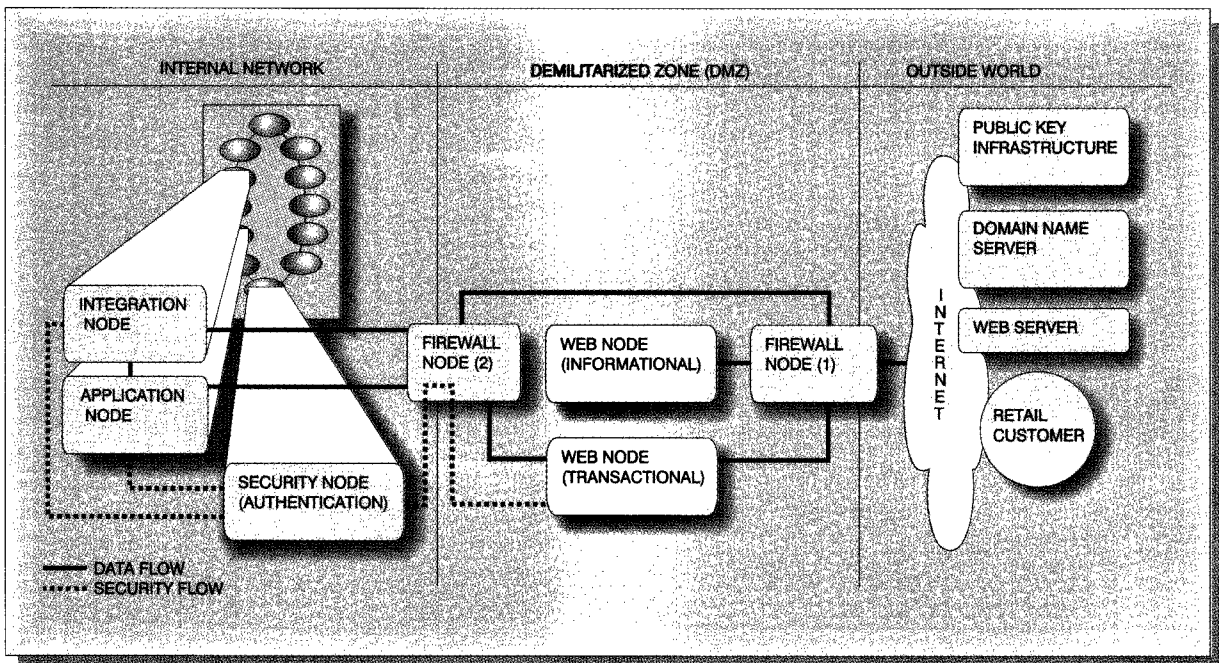
Although State Farm representatives were very interested in the quick start that ESS assets could give

them, they were equally interested in the fact that ESS provided a structure that all of the parties within their very large IT organization could agree upon and to which they could relate.

Showing the ESS assets during the proposal for consulting work helped to clarify for the client: what an enterprise architecture would look like, how it would be structured, and that existing material could give the company a quick start toward creating their own architecture. As in the previous example, they decided to work with IBM consultants.

**Canada Trust.** The needs of the bank were a little different. This client was interested in how to build systems using technology of the late 1990s, but *not* interested in starting work on all aspects of a full-blown enterprise architecture at the same time. In-

**Figure 6** The NC transactional network diagram. This basic pattern depicts access to business applications via the Internet using secure firewall nodes that provide access control to the internal systems.



stead, once representatives had seen some of the ESS architectural assets, they decided to choose only one small portion as a starting point. The approach we used to scope the work is described in the next section. Again, however, the asset-based approach, and in particular a demonstrated ability to understand the "big picture," provided a level of confidence in IBM's expertise that contributed significantly to their decision to work with us.

### In contract definition and scoping

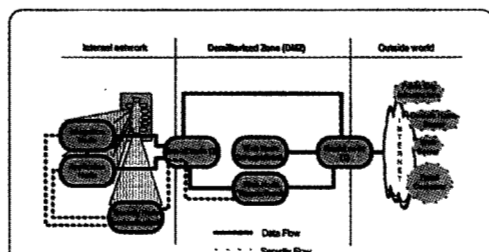
In our experience (and our view is widely shared in the IT industry), too many projects disappoint or fail to deliver, not through any technical failure, but from an inability to completely and unambiguously define the requirement. Client and vendor each believe they have understood the other, but misinterpretations frequently arise. The use of architectural templates as "strawmen" (i.e., examples) for discussion at this stage contributes to a common understanding of the requirement and helps to identify dangers (e.g., falling into the silo trap) and risks. Thus, the asset-based approach to scoping greatly improves the ability of the project to deliver the right solution, on time, and within budget.

**Canada Trust.** Canada Trust was interested in an overall enterprise architecture, but wanted to commit to only one aspect at a time. After reviewing various examples of the ESS material with the IBM consulting team, representatives chose to start with a small subset. The most pressing problem for their organization was coherence; they were purchasing packages and developing many new application systems where data and function were dispersed across different technologies and distributed over different locations. Some of these data were considered to be "data of record" for the company. They were becoming uncomfortable with the relatively uncontrolled placement of data and function across the corporate locations.

The immediate need was to develop a very pragmatic process, so that the architecture group could help the development project teams decide where to place function and data in a distributed application, using technology of the late 1990s instead of either the "fat client" or centralized, mainframe-based approaches of the past. Management and lead architects saw in ESS assets a number of prebuilt architectural patterns, and in addition, guidance on several important top-

Figure 7 The overall flow of a transaction traced through the nodes on the logical pattern

## To illustrate external access to business applications, the overall flow of a transaction has been traced through the Nodes on the Logical Operational Architecture diagram . . .



This logical infrastructure diagram introduces the concept of Demilitarized Zone (DMZ), which is discussed in more detail in the Security Guidance section. A DMZ is the area between the two firewall Nodes, where the publicly accessible systems reside (e.g. Web Node). Its purpose is to protect the internal network from external access.

### Basic Operation from an External Internet Client

The following points illustrate the basic flow for an external Internet client accessing the company's business applications.

1. From an external Internet client (browser), a user will choose to link to a destination (URL) by typing it in, or clicking on a saved "favorite link", or clicking an imbedded link on the page that is currently displayed.

- The link provides a name that must be resolved into a specific network address.

2. The address of that link will be found by the client, or from a name server on the Internet (System). The function of the name server is to resolve the address of a URL (or resource server).

- If the default name server cannot find the address, then it typically would search other name servers until the address is resolved (for more details see the level 3 services description of the Domain Name System.)
- This is an area of uncertainty when it comes to performance since for the public Internet there may be an unknown number of searches performed to resolve the address (for details see the performance guidelines on Name Resolution in the Internet Guidance section).

3. Once the address of the specified link has been found, a message will be sent to firewall (1) to gain access to the Web Node, which is where the destination of that URL resides. As discussed in more detail in the Security Guidance section, the Web Node in the DMZ is separated into two for security reasons:

- An informational Web Node which serves Web pages to the general public, does not require authentication of users, and does not have access to back-end systems.
- A transactional Web Node which serves a subset of the outside world, requires user authentication, and has access to and knows how to get to back-end systems in the trusted network. In this scenario, authentication is provided by the components of the Security Node.

ics—including "where to place data and function" within the patterns.

This guidance contained not only examples of data and function placement in patterns such as the call center, but also a general process for deciding placement, with many guidelines supporting that process. In making the placement decision there are a number of conflicting factors that can lead toward either distribution or consolidation. Some of these factors are technical issues, and some are nontechnical (we called them "project-related issues"). Figures 8 and 9 show the factors that are considered. These are examples of some of the assets related to placement.

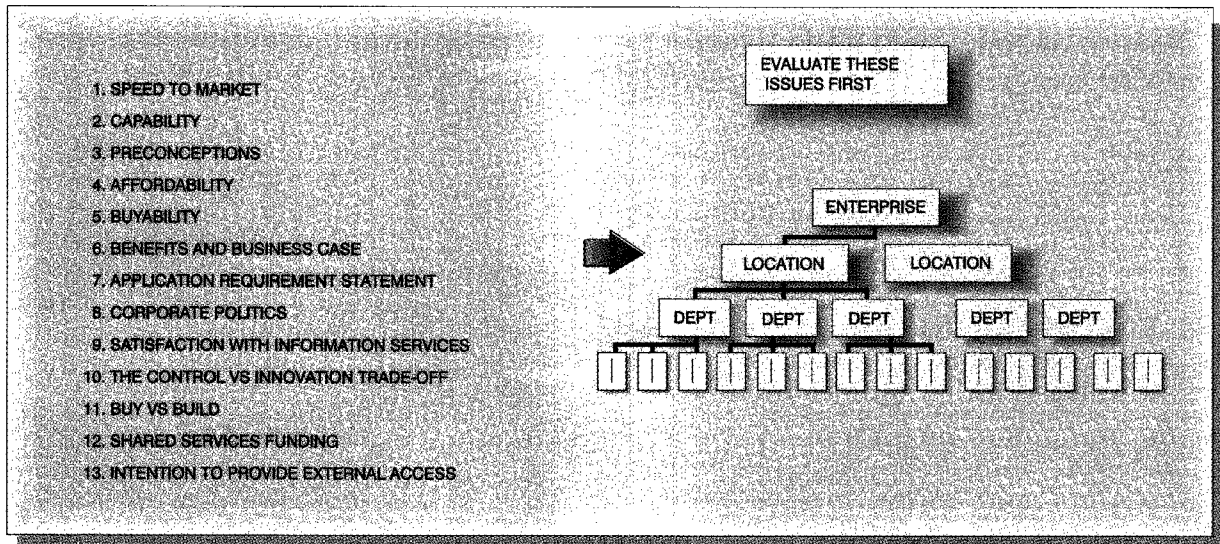
We found that one benefit of showing ESS assets at this point in a consulting engagement is that the client sees samples of an overall architecture and can more clearly define the scope of what is needed. The

old adage, "I'm not sure what I want, but I'll know it when I see it," applies here. With samples, the risk of scope misunderstandings is greatly minimized.

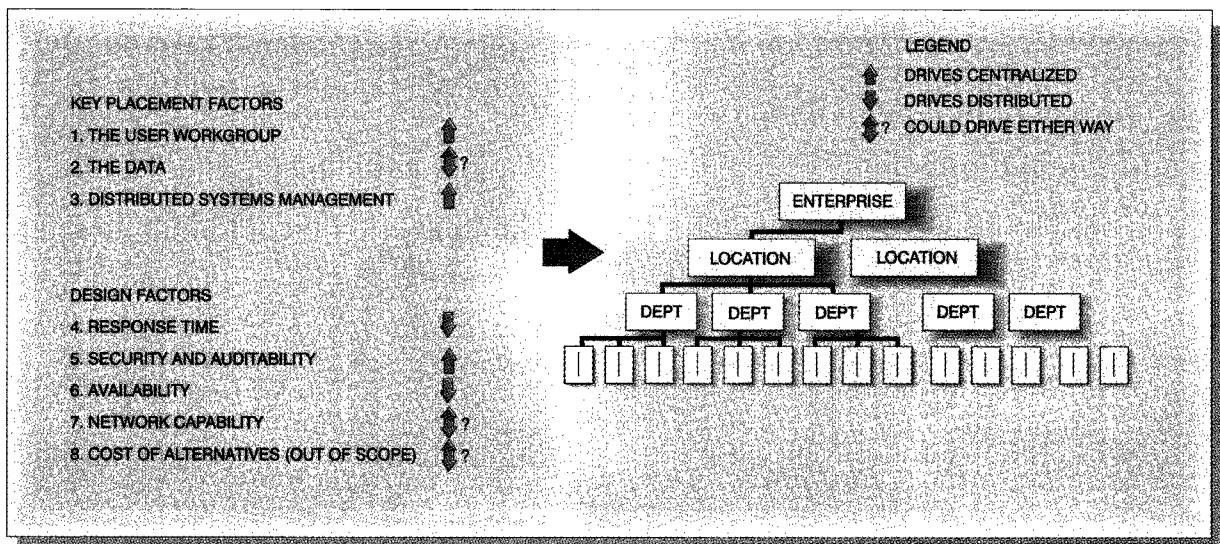
In the case of Canada Trust, the assistant vice-president of architecture and information management was able to say, "No I don't want all that yet, I only want the placement material now." It then became easier to agree on a very specific set of deliverable material for the first engagement, and for both IBM and the client to agree upon scope and time for the project.

As this initial project progressed, it was decided that the next area for attention was the architecture for the intranet infrastructure. Again the client was able to specifically indicate which parts of the ESS material were of interest—effectively saying, "Now I want to add the intranet part, and I can see in gen-

**Figure 8** Project issues affecting placement. Depending on corporate culture and other factors, these will drive placement toward either a centralized or decentralized model.



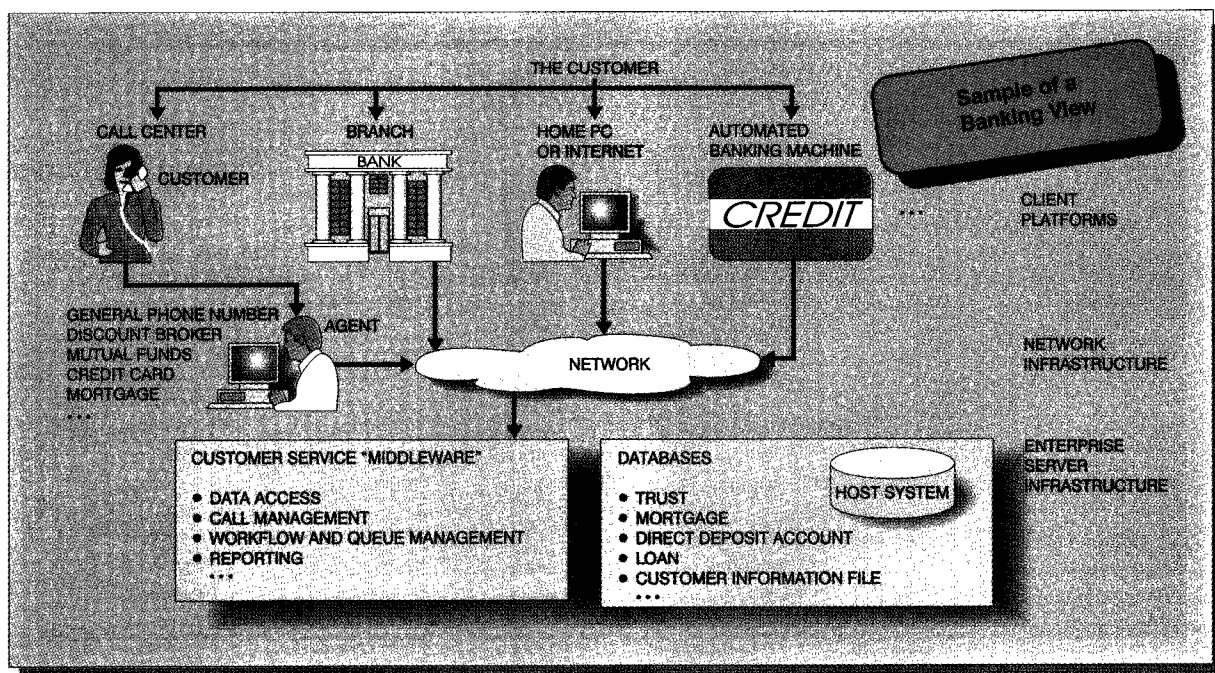
**Figure 9** Technical factors affecting placement. These also drive placement toward either a centralized or decentralized model.



eral what it will look like.” A second engagement was scoped and begun. Specifically, in the second project the intranet pattern was used, plus the building block definitions, walkthroughs, etc., that are associated with the pattern. Again, it was clear what was to be included and what was not.

To summarize, our experience has shown that the effect of asset-based work in contract definition and scoping is a much clearer definition of scope, effort, and time needed. This should lower the risk of misunderstandings and improve the overall chances for successful projects.

**Figure 10** ESS contextual view of customer access points. The customer may choose from a variety of different ways to access the financial aspects of the bank, but the supporting infrastructure should be common whenever possible.



### During the engagement

We expected to see that the use of populated templates or patterns would provide a fast start for the project during the engagement, often allowing a higher quality work product in a shorter time. This is of increasing importance as IT departments come under greater pressure to provide new systems in support of a window of business opportunity, where the profits are proportional to the speed at which the system reaches the market.

**Company A.** The first pilot use of an asset-based approach and the ESS materials was executed at the large insurance company referred to as Company A. The materials were in an early stage, harvested from a number of successful engagements but still only partially complete. Even so, it was found that significant time savings could be obtained by reusing material from previous experience in an organized fashion.

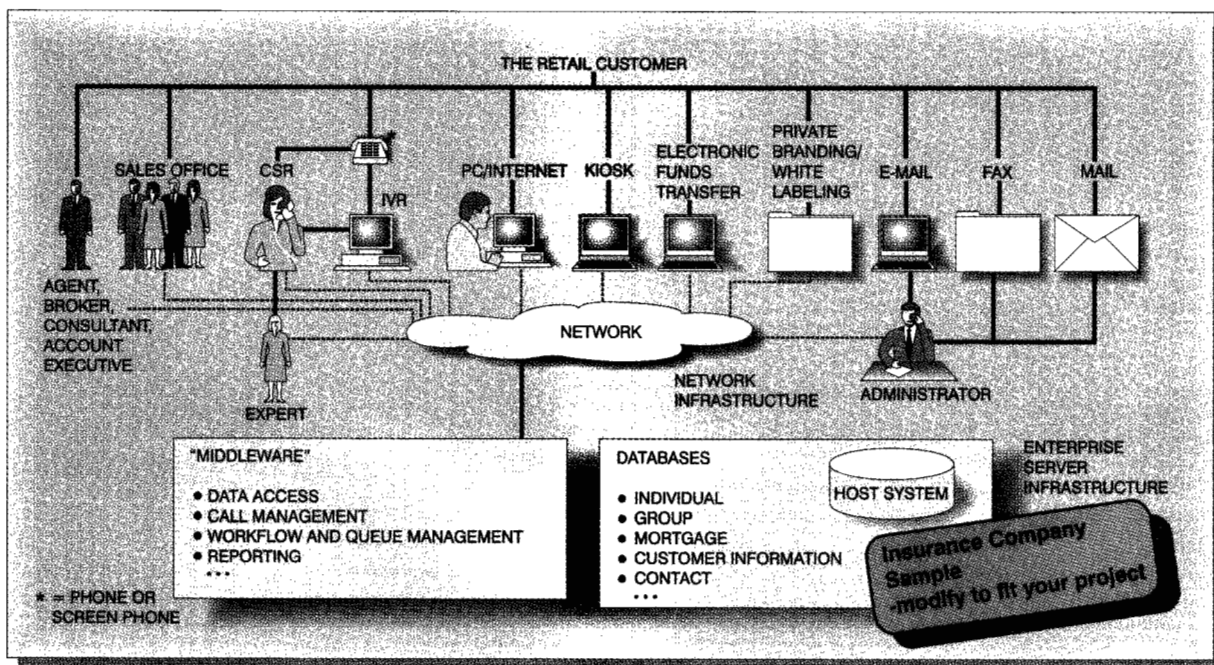
One of the very important requirements of the architecture for this client was to be able to support any channel that customers wished to use in doing

business. A contextual view was used to graphically depict the support needed to operate across the silo-based legacy systems. Figure 10 shows the initial contextual view, which was part of the ESS material at that time, and Figure 11 shows the way that it was enhanced and modified to suit the needs of this insurance company and to capture the additional access points needed.

This contextual view in Figure 11 shows very clearly the need to share data and applications across multiple access points (or delivery channels). It is the kind of visual display that user executives can relate to. When working on a project for any of these channels, the implication is clear—the project must be designed so that function and data can be easily shared across all channels and not be specific to only one.

We think of contextual views as the kind of picture that the users would themselves draw to show what they want. We tell our architects and designers to ask themselves: “If the user draws this picture of the requirements, what does it mean to my infrastructure?”

**Figure 11** An insurance company's modified version of an ESS contextual view, showing customer access points for the company's business



One specific example resulting in time saved at Company A involved the ESS call center material. In an earlier engagement, work had been done to specify the data needed to support a world-class call center for a banking client. The original specification work for that engagement had been done by a team of 15 IBM and client experts, along with users and operations and networking personnel, in three weeks of intensive joint workshops. Some of this work was subsequently generalized and abstracted for the ESS asset base. During the engagement at Company A the material was used as a starting point. This workshop completed in two and one-half days—most of the previous work was adopted, and a few new insurance-specific data types were defined.

We found that we were also able to adopt most of the earlier recommendations on where to place the data. The contextual view of channels needing access to shared data (described in Figure 11) was used as input to the placement decisions.

Two examples show a small subset of some of the material that was modified by Company A. Figure 12 is one of the tables describing the data, and Fig-

ure 13 is an example showing placement of data in the call center and at an enterprise-wide level of sharing.

Figure 12 is representative of a few similar pages that hold the "data groups" typically needed by a successful call center. While early in the design the client might not know the details of the database design, we know there will be a need for customer data, calling line data, product data, etc., and we know who will need to share the data, how often the data will likely change, etc. Figure 13 shows where the data have resided in other successful call centers. The arrows show the potential movement of a copy of the data from one level of sharing to another.

The benefit for Company A was that, by reusing these assets, they were able to define the data needed and make an initial decision on where to place the data in two and one-half days, saving two and one-half weeks over the time required to get to the same high-level design for the data on a previous project.

The approach during the engagement was to make use of material already structured into

**Figure 12 Data for call center (one of several tables)**

| DATA NAME                                                                                                              | USED BY                                                              | CREATED/<br>UPDATED BY                                                              | CURRENCY               | SIZE<br>ESTIMATED,<br>TO BE VALIDATED     |
|------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------|-------------------------------------------------------------------------------------|------------------------|-------------------------------------------|
| <b>CUSTOMER DATA</b><br>• CUSTOMER SYSTEM (CIF)<br>• NAME, ADDRESS, POINTERS TO LEGACY<br>SYSTEMS, SIGNATURE? ...      | • ALL CSRs<br>• BRANCHES<br>• AGENTS<br>• HEAD OFFICE<br>• CUSTOMERS | • ALL CALL CENTER<br>AGENTS<br>• BRANCHES, IVR<br>• AGENTS, HQ                      | FREQUENT<br>CHANGES    | LARGE<br>• 3 MILLION<br>• MULTIGIGS       |
| <b>PRODUCT INFORMATION</b><br>• ALL PRODUCTS, EVERYTHING YOU SELL<br>e.g., MORTGAGE, MUTUAL FUNDS, LIFE,<br>HEALTH ... | • ALL CSRs<br>• BRANCHES<br>• AGENTS<br>• HEAD OFFICE<br>• CUSTOMERS | • CENTRAL PRODUCT<br>MANAGEMENT<br>ORGANIZATIONS                                    | RELATIVELY<br>STATIC   | 20 MEG?                                   |
| <b>RATES</b><br>• CURRENT RATES AND PAST RATES FOR 5 YEARS<br>• FEES, SERVICE CHARGE, FOREIGN CURRENCY<br>...          | • ALL CSRs<br>• BRANCHES<br>• AGENTS<br>• HEAD OFFICE<br>• CUSTOMERS | • ALL CALL CENTER<br>AGENTS<br>• BRANCHES, HQ<br>• BATCH JOBS,<br>CUSTOMER          | FREQUENT<br>CHANGE     | CURRENT<br>• 10 MEG<br>HISTORY<br>• LARGE |
| <b>CONTACT HISTORY</b><br>• LOG OF ALL CUSTOMER CONTACTS                                                               | • ALL CSRs<br>• BRANCHES<br>• AGENTS<br>• HEAD OFFICE<br>• CUSTOMERS | INSERT ONLY<br>• ALL CSRs,<br>AGENTS<br>• BRANCHES, HQ<br>• BATCH JOBS,<br>CUSTOMER | CONTINUAL<br>ADDITIONS | LARGE<br>• 100 + MEG                      |

ESS assets from previous engagements, and to add material where the needed information did not exist. The process can be summarized as follows:

1. Reuse and modify if possible.
2. Create new material where needed.
3. In all cases, follow the ESS architectural structure so that any material created will fit into the client's asset base in an integrated fashion.

**State Farm.** Six months after the Company A engagement, we worked with State Farm. The ESS assets had matured and we had pattern diagrams with textual walkthroughs for call-taking scenarios and use cases, as well as building blocks defined.

Our walkthrough of what happens when a call comes in was now well-defined, and our discussions with State Farm on this asset were quite straightforward.

The basic pattern for the Internet access to legacy systems was also defined, with its dynamic behavior in the form of walkthroughs of a use case for "what

happens when a URL is clicked," and many of the logical building blocks (nodes and components) were also defined by that time.

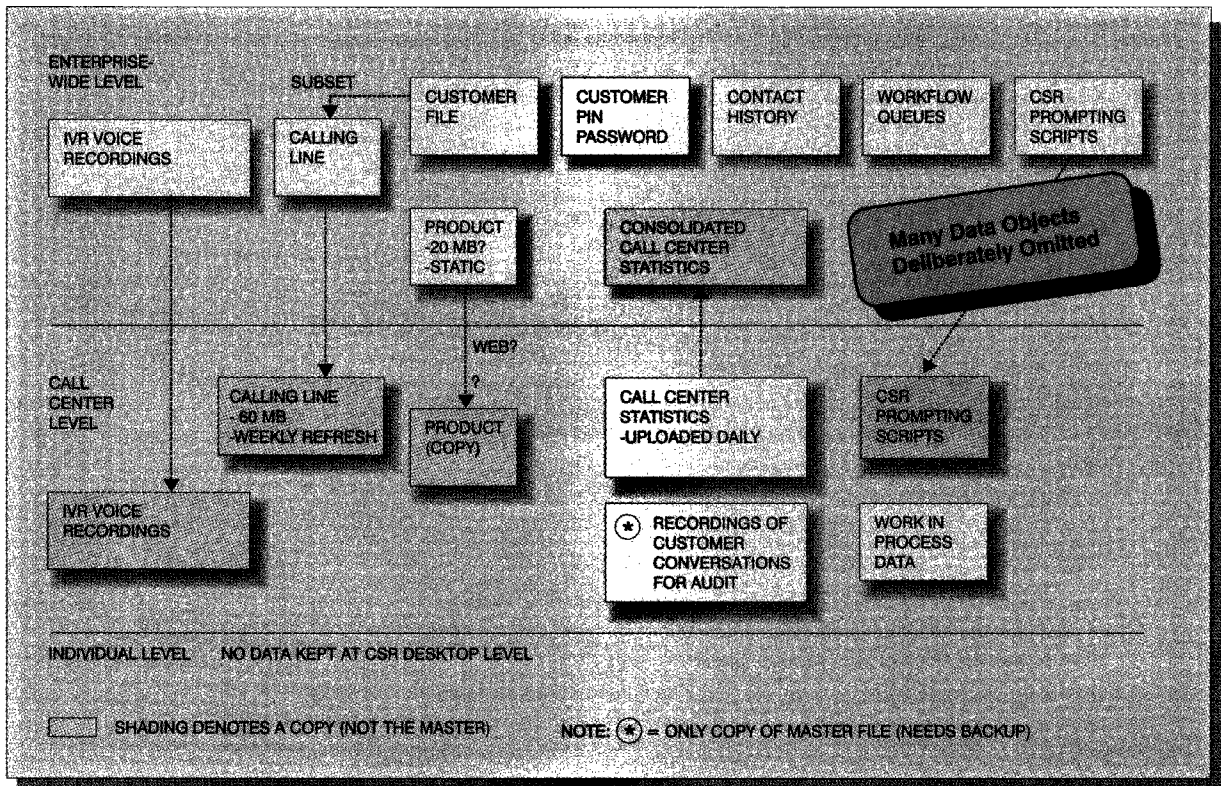
The approach again was to reuse the patterns as a starting point, and add material or modify only as needed. In this case the material was much more complete so there was much more reuse.

Additional material that was developed during this engagement (some of which was subsequently abstracted, generalized, and added to the Design Guidance section of ESS) related to the structure of an application (specifics on how one would separate presentation from business and data access logic, etc.). The security information was also enhanced.

### Observed benefits of using ESS assets during engagements

During the pilot engagement with Company A, the use of assets and the basic structure of the ESS architecture allowed us to be much more productive than if we had not had the material. The engage-

**Figure 13 Data placement.** This placement considers sharability, manageability, security, etc. A fast transmission line is assumed.



ment took four months and required several of our best “gurus” in architecture and design. There was a team from Company A representing various areas of expertise. The team from IBM included:

- A top engagement leader
- A lead architect with extensive experience
- A top architect in networking
- A specialist in the Internet
- Two lead designers with expertise in call center architecture
- A security expert
- A database architect
- Other subject matter experts

One of the core team members of the ESS group worked part time to bring the assets to the engagement and to mentor the team on the structure, approach, and process for using the pilot ESS assets. This assistance provided a quick start in several ar-

reas of the architecture. Having the assets and structure of ESS allowed the IBM and client teams to be much more productive than would otherwise have been the case.

By our next engagement, six months later, our ESS asset base was considerably extended. Roughly the same level of deliverable material was produced, but with considerably more favorable metrics. The engagement time was two months instead of four, and the IBM team was smaller, comprised of:

- An insurance industry application architect
- A lead architect (from the previous pilot at Company A)
- A small amount of time from subject-matter experts (security was one of the subjects)
- A member of the ESS core team to assist with start-up, checkpoint walkthroughs, and advice during the engagement

However, unlike the IBM team, the client team was not reduced in size. We felt it was important to have a similarly sized client team to ensure "buy in" and participation from the various stakeholders in the client organization. This is an important point—in our experience, it takes time and involvement to ensure acceptance by influential individuals with strongly held opinions.

The net result of more extensive assets was faster delivery time, with fewer highly skilled consultants required. In both cases, the clients were very satisfied with the results of the IBM team's work.

The question might arise, "Why would a firm that charges a fee for services want to implement a new approach that delivers higher quality work products in less time with fewer people?" The answer is that there are not enough highly skilled architects and designers available to meet client needs, and that the elapsed time from request to delivery needs to be shorter. The use of assets tackles both those issues and at the same time produces work products of higher quality with less risk.

In our experience, the client obtains excellent value for cost and tends to renegotiate follow-on contracts to expand the scope or move into new areas and projects. For the client it means better service, while for IBM it means repeat business with satisfied clients.

### After the engagement

Representing architectures in a coherent and consistent way and using a common lexicography brought significant benefit to both IBM and client members of the engagement teams. It promoted fast and clear communication and reduced misunderstanding. It has been found that some enterprises with large IT departments have reaped further benefit from adopting a similar way of representing their systems and thus promoting better communication among their own project teams.

A critical part of a successful engagement is to take the time to step back and reflect on what good new ideas, designs, and patterns may have been developed that can benefit future architectural and design work. This is not easy for a services company, since these days good people are immediately needed for the next engagement. IBM is continuing to seek out key ideas from the best working designs and solutions, and to "harvest" these ideas and store them

back into the asset base of ESS. This applies both to client engagements and to our internal experiences.

**IBM perspective.** A high-profile case is the harvesting of the patterns and building blocks of the highly scalable IBM Internet solutions from the Nagano Olympics. As these technologies are deployed in successful solutions in the Internet banking area, we have enriched the NC transactional pattern to include new building blocks, such as the network dispatcher, which were not in the original pattern.

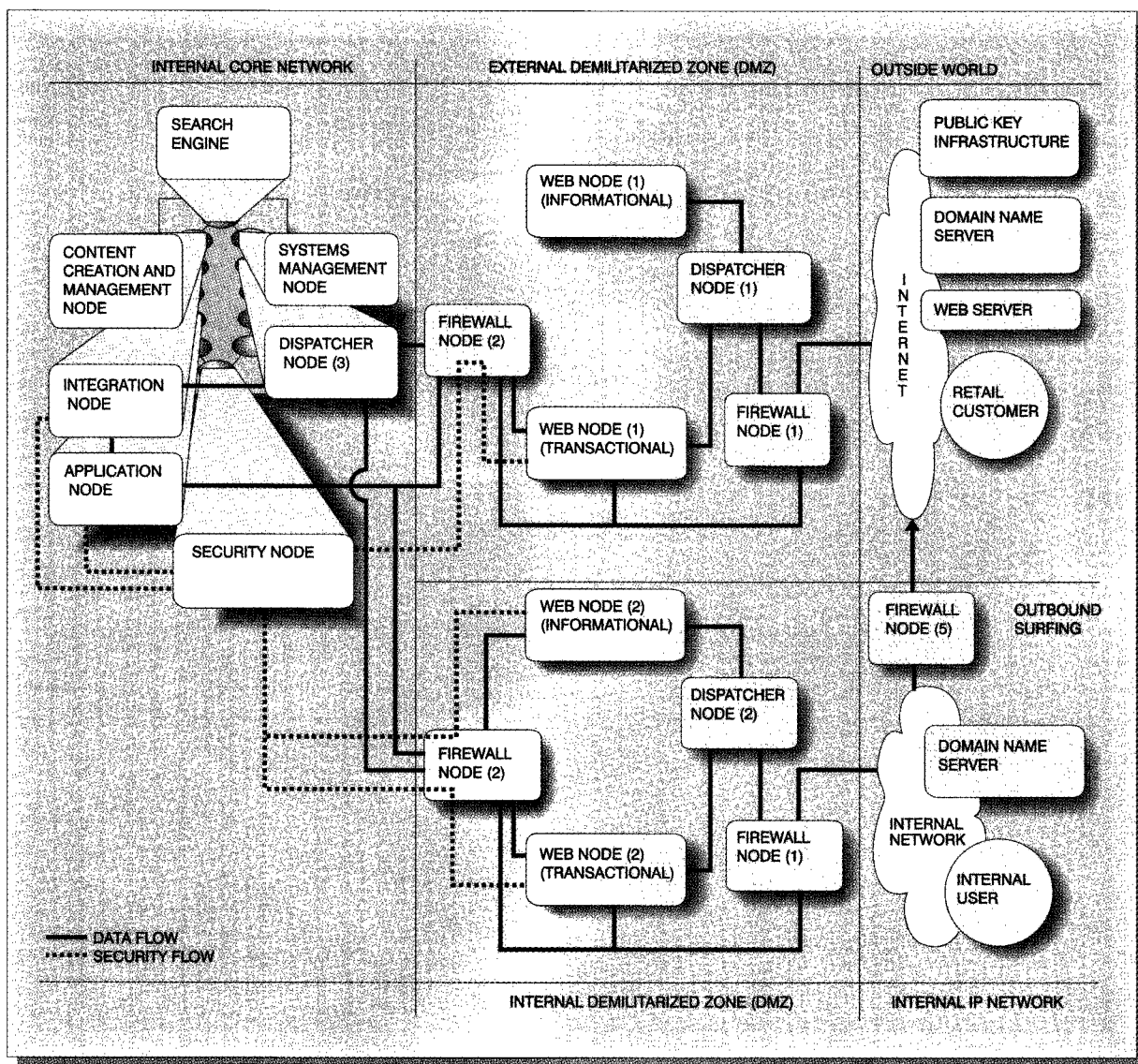
Figure 6 shows the original very basic NC transactional pattern, while Figure 14 shows an elaboration on that pattern based on a highly scalable and successful Internet PC (personal computer) banking implementation. When highly scalable solutions are needed, we now start with this elaboration as a new base pattern. Each of these patterns brings with it the dynamic behavior in the form of a textual walk-through, the details of the building blocks, and the guidance, etc.

At Canada Trust the ESS basic Internet pattern was used to develop an architecture for the intranet. The process used was the following:

1. The consultants working with Canada Trust used the basic ESS design pattern for the Internet in Figure 6 as a starter set.
2. They evolved a new, highly scalable pattern for the intranet that would share the same internal core banking systems. They also used scalability experience from Nagano, which had been utilized in PC banking systems elsewhere.
3. They developed the new pattern shown in Figure 14 (which includes the diagram, textual walk-throughs of use cases, the definition of some new nodes, detailed information on each new node and the components within it, selection criteria for vendor products to implement the nodes, etc.).
4. Canada Trust has since adopted the repository from ESS as a way to store its own version of the architectural assets.

A critical point is that this newly evolved pattern is in the same basic structure as the rest of the ESS assets. This common structure makes it very easy to harvest the new ideas and use them to improve the asset base. In the past it was difficult to capture good ideas from successful real-life projects; usually there is no common structure—no common "look and feel" to the designs. ESS provides that common structure.

**Figure 14** Highly scalable evolution of the basic pattern: Combined Internet and intranet PC banking network diagram



**State Farm perspective.** State Farm found that while the content of the ESS asset base helped by giving them a “quick start” in defining their own architecture, it was the structure of the asset base that was of particular interest. They used parts of the ESS semantic structure to create their own architecture asset repository. State Farm stored their architecture on an internal Web site and used HTML (hypertext markup language) and links to store their data, pro-

vide the capability to “drill down” to detailed information, and share the information throughout the company. Representatives remarked that the ability to share the architectural vision in this way significantly improves understanding and communication within their very large IT department.

Mike McFadin, a manager in State Farm’s System Department, provided the following perspective.

The mission of the State Farm Technical Architecture program was to establish an enterprise architecture at the end of a 90-day period. To accomplish this, we concentrated on creating an architecture that represented the plan for technologies to be used by our computing systems. We used the ESS structure and vendor-independent research materials to achieve our goals.

The layering and drill-down structure we acquired from ESS, coupled with our decision to use the intranet and HTML to publish our architecture, has been effective in allowing a large developer community to access and use this material. A developer can start with the logical template of a given technology design; via hyper-linking of HTML we have connected this information with our specific product choices and technical guidance documentation. Our architecture is now accessible by any developer from any desktop at anytime using his or her Web browser.

Creating an enterprise-scale technical architecture is not an easy task, and our team worked many long days to achieve our goals. Reusing as much as we could from ESS helped us achieve our goal to get our technical architecture off the ground in a short amount of time. This was only the beginning. We still needed to make a significant investment in education and marketing to our developer community for this to work. We conducted many overview presentations and completed several multiple-day training sessions as we rolled out the architecture in our organization.

**Canada Trust perspective.** The following are quotes from Canada Trust's management team. Robin Hewson, Assistant Vice President, Architecture and Information Management, Canada Trust, provided this perspective.

I think of architecture as a process that helps in making technology decisions. Often in the past we have discussed our business requirements, then seemed to "pole vault" right into discussing vendor products. Architecture is giving us a more structured process to define what we want to do, and how we want to do it, before debating the choice of vendor products.

The ESS assets brought value to Canada Trust as an architecture definition approach. The whole idea of architectural definition via conceptual views, pattern diagrams, nodes, and component

definitions and structures was impressive. We have adopted the ESS structure, and the approach IBM followed with "starter sets," as our method for creating new architectures. It brings architecture from an art to a process. This is important, because it provides us with a level of rigor we have missed at Canada Trust.

Frank Coletti, Manager, Enterprise Architecture, Canada Trust, contributed these remarks.

We have initiated a process using "Data and Function Placement" workshops for project teams that are doing new development or implementing packages. In the past these teams would consult one of our experts in architecture or operations, who would use what sometimes appeared to be a "black art" to provide advice on how to proceed. The new architectural structure has provided a documented approach, taking some of the mystery out of the decision making and resulting in more consistency across the work of individual experts.

We are moving toward an engineering approach. We are attempting to focus the knowledge of our experts into a structured architecture, using the ESS material as a starter set and applying their expertise to ensure that it is tailored to our needs here at Canada Trust. We then follow a workshop process to imbed this architecture into each major development project.

Tom Wheatley is one of the certified IT architects from IBM Consulting who worked on the Canada Trust engagement. His following comments are from a practitioner's point of view.

I feel that one of ESS's greatest strengths during the early part of the engagement is that it provides a "straw man" that can focus discussion. This is important at contract time—in order to understand requirements and set expectations—but it is also very valuable when starting the engagement. Instead of starting with a blank white board and having days of philosophical discussion on what color pens to use, we can get to work right away with an example. It should be carefully reviewed and changed or enriched as required, but in its current state it provides a quick start for valuable work on real deliverables.

It is valuable to note that one ESS engagement can (and should) build on previous engagements. For

example, after the intranet architectural engagement at Canada Trust, follow-on work defined the architectures for Web surfing by employees, and doing Internet mail. These engagements were performed by different client and IBM personnel, but they took up where the intranet mandate stopped—following the same structure and approach. That provides valuable continuity to clients and reduces wasteful revision or rework of existing assets. Hopefully it also increases the chances of successful integration.

### Concluding remarks

We have been very encouraged by our early experiences. We believe this approach has the potential to deliver real benefit to IBM and its clients. It is reasonable to anticipate that the value will increase as the process matures and becomes more pervasive throughout our architecture projects (within IBM and for our clients). We expect that further experience will expand and refine the assets.

We have seen that:

- At proposal/bid time the use of the high-level assets to provide contextual views helps the client and IBM, ensuring that they have a common understanding of the business requirement. The supporting detailed assets help build confidence in the proposed implementation.
- In contract definition and scoping the use of the assets as “straw men” improves communication. At this critical stage, the assets help in identifying and testing assumptions that have all too often in the past been the downfall of the project. The assets help to both define and document the scope of the contract for all parties.
- During the engagement the observed benefits were shorter delivery times and more confidence in a successful solution. Because in the early stages of the engagement the assets focus discussion, the valuable work on real work products can start quickly. In some cases the results were startling—the assets led to productivity gains of up to four times in these architecture engagements, although of course no two client environments are exactly the same. Other experiences since the engagements described here reinforce this view and have helped our consultants to deliver results within a very short window of time, allowing our clients to meet business imperatives.
- After the engagement there are lasting benefits to both IBM and our clients. Some of our clients feel

that adopting similar approaches for documentation and process in their own organizations will bring them continued and further value. Within the IBM practitioner community, the use of the standard structure and language has helped in capturing improved, updated, and extended assets which, in their turn, deliver benefits in subsequent engagements.

There is clearly much left to do in expanding and enhancing the assets and in fully adopting the emerging Architecture Description Standards. Preserving the technical vitality of the assets in our fast-changing world is a challenge. We believe this challenge can be addressed by an active harvesting program that involves each new ESS engagement, together with a quality assurance process implemented through a worldwide technical review board of leading IBM specialists and practitioners in relevant disciplines. Should practitioners always use patterns that are already proven and used in real projects (which, though minimizing risk, may not take advantage of new technologies)? What about patterns, recommended by our top technology specialists, that potentially provide competitive advantage to our clients, but may increase project risk in the early days of the technology? A technical review board can help address these questions.

Perhaps the biggest challenge is in promoting the assets across the worldwide community of IBM architects. This involves communication and education but, above all, motivation. The latter implies a cultural change at multiple levels. The “heroic model”<sup>7</sup> is well ingrained. Reusing assets, as much as building those assets for reuse, needs to be viewed as desirable behavior. Consultants and service professionals need time and appropriate support between engagements to capture new and improved or updated assets. Reuse, building on experience across the whole community, has to become a fundamental part of the way in which our practitioners increase their architectural skills and knowledge.

The effective reuse of assets in IT solution development has long been a key goal in the evolution from handcrafting solutions toward true software engineering. Our experience in these engagements gives us confidence that the ESS approach makes a significant contribution to our ability to reach this goal.

Reuse is not free. The business case needs to be made to continue the investment in the infrastructure to support the processes of harvesting, hardening, qual-

ity assurance, communication, and education. We believe our experience to date clearly supports that business case. The foundation has been laid for providing a higher quality of service to our clients, better business for IBM, and a more rewarding way for our practitioners to work and increase their knowledge and effectiveness.

## Acknowledgments

The authors gratefully acknowledge the work of numerous IBM contributors as the ESS assets were piloted and then broadly deployed in multiple client engagements. We would also like to thank the clients who participated in the early engagements; their input and encouragement was vital and is greatly appreciated. These clients included State Farm and Canada Trust, who have very kindly agreed to having their names used in this paper. Particular thanks are due to our reviewers, including George Galambos, Burnie Blakeley, and John Fetvedt.

IBM contributors during client engagements discussed here include Amir Alexan, Angus Chisholm, Peter Colbeck, Christian Corso, Bruce Crossman, Jim Davey, George Galambos, David Gamey, Qing Ge, Bill Graber, Edward Hood, Paul Kraeger, Leo Marland, Richard McDonald, Silvia Pighin, Sugantha Raj, Colin Rous, George Semler, Emeline Tjan, and Tom Wheatley.

\*Trademark or registered trademark of International Business Machines Corporation.

\*\*Trademark or registered trademark of Lotus Development Corporation.

## Cited references and notes

1. E. C. Plachy and P. A. Hausler, "Enterprise Solutions Structure," *IBM Systems Journal* **38**, No. 1, 4-11 (1999, this issue).
2. A "silo" is a part of an IT solution that is not integrated with other parts. The name comes from the visualization of such a solution—the different applications stand apart from each other, like farm silos in silhouette against a prairie sky.
3. P. T. L. Lloyd and G. M. Galambos, "Technical Reference Architectures," *IBM Systems Journal* **38**, No. 1, 51-75 (1999, this issue).
4. R. Youngs, D. Redmond-Pyle, P. Spaas, and E. Kahan, "A Standard for Architecture Description," *IBM Systems Journal* **38**, No. 1, 32-50 (1999, this issue).
5. A "use case" describes one way that the IT system could be used. For example, "draw funds from checking account" is a use case.
6. An introduction to IBM's Open Blueprint is available at <http://www.software.ibm.com/openblue>.
7. The "heroic model" of software development starts with customer requirements to build a one-of-a-kind solution. It may include products from multiple vendors and require custom-

ized middleware to integrate the products into the solution architecture. Often the developers must perform heroic feats to complete such a solution.

*Accepted for publication September 28, 1998.*

**Tina Harris** *IBM United Kingdom Ltd., 1 New Square, Bedford Lakes, Feltham, TW14 8HB, England (electronic mail: tina\_harris@uk.ibm.com).* Ms. Harris is a member of the IBM Solutions Consulting and Integration Practice. Currently, she is the program manager for ESS; she joined the team in 1997 to lead ESS deployment. Her career in IBM has spanned many roles, including instructor, technical specialist, lead systems engineer on enterprise accounts, sales, and management. She received a B.Sc. degree in mathematics from City University, London.

**John W. Rothwell** *IBM Canada Limited, 3600 Steeles Avenue East, Markham, Ontario, Canada L3R 9Z7 (electronic mail: Rothwell@ca.ibm.com).* Mr. Rothwell is a certified senior consulting IT architect in the e-Business Architecture Consulting Practice, based in Toronto, Canada. He works as a lead architect in consulting engagements in the banking, insurance, and retail industries. For the past two years he has worked as a key core team member of the ESS project, bringing current experience from our client engagements to help shape the content and structure of the ESS assets. Prior to his involvement with the ESS project, Mr. Rothwell worked as a consultant in client/server computing, where he coauthored and taught function and data placement and design classes. Previously he worked as a senior consulting systems engineer for several clients in the petroleum, retail, insurance, banking, and utilities industries, with expertise in database systems. He worked in the area of database performance on the early relational database systems and published performance papers for the DB2<sup>®</sup> (DATABASE 2<sup>™</sup>) group at the IBM Santa Teresa Laboratory. Mr. Rothwell is a registered Professional Engineer in Ontario; he received a degree in chemical engineering from the University of Waterloo.

**P. T. L. (Tim) Lloyd** *IBM United Kingdom Ltd., P.O. Box 31, Warwick, CV34 5JL England (electronic mail: tim\_lloyd@uk.ibm.com).* Mr. Lloyd is a consulting IT architect in the IBM United Kingdom Object Technology Practice. He led the ESS Technical Architecture team from its inception in 1996, and subsequently became team leader for overall ESS development. Currently, he is lead architect for the ESS Operations team. He previously worked as enterprise systems engineer and solutions project consultant for several large IBM enterprise clients in the government, retail, and utilities industries, as technical architect on a large insurance solutions development project, and as regional specialist in various subject areas, including large systems hardware and software, mass storage systems, and office systems. Other areas of experience include the application of image processing and workflow in enterprise-scale solutions. He received a B.Sc. degree in theoretical physics from Manchester University.

Reprint Order No. G321-5699.