Management of New Federated Services

Preeti Bhoj, Deborah Caswell, Sailesh Chutani,
Gita Gopal, Marta Kosarchyn
Broadband Information Systems Laboratory
HPL-96-131
September, 1996

Internet, federated systems, federated service management, Broadband Interactive Data Services, fault diagnosis, customer support, service contracts

The explosive growth of the Internet, widespread use of the World Wide Web, and a trend towards deployment of broadband residential networks are stimulating the development of new services such as interactive shopping, home banking, and electronic commerce. These services are *federated* since they depend on an infrastructure that spans multiple independent control domains. Managing federated services and providing effective support to the customer of these services is difficult, because only a small part of the environment can be observed and controlled by any given authority. We characterize different dimensions of this problem, using our experience with the deployment of a system that gives the home consumer broadband access to community content as well as to the Internet. This type of system is referred to as Broadband Interactive Data Services of BIDS. We then focus on diagnosis and describe a customer support tool that was developed to partially automate diagnosis in BIDS. We use the experience with this tool to derive a blueprint for a general architecture for managing federated services. The architecture is based on service contracts between control domains.

Management of New Federated Services

Preeti Bhoj, Deborah Caswell, Sailesh Chutani, Gita Gopal, Marta Kosarchyn
Hewlett-Packard Laboratories, Palo Alto, CA 94304

Abstract

The explosive growth of the Internet, widespread use of the World Wide Web, and a trend towards deployment of broadband residential networks are stimulating the development of new services such as interactive shopping, home banking, and electronic commerce. These services are *federated* since they depend on an infrastructure that spans multiple independent control domains. Managing federated services and providing effective support to the customer of these services is difficult, because only a small part of the environment can be observed and controlled by any given authority. We characterize different dimensions of this problem, using our experience with the deployment of a system that gives the home consumer broadband access to community content as well as to the Internet. This type of system is referred to as Broadband Interactive Data Services or BIDS. We then focus on diagnosis and describe a customer support tool that was developed to partially automate diagnosis in BIDS. We use the experience with this tool to derive a blueprint for a general architecture for managing federated services. The architecture is based on service contracts between control domains.

Keywords: Internet, Federated Systems, Federated Service Management, Broadband Interactive Data Services, Fault Diagnosis, Customer Support, Service Contracts.

1. Introduction

The explosive growth of the Internet and the widespread acceptance of the World Wide Web have caused a corresponding growth in new offerings of networked services. The Internet is already the world's largest business computer communications network, with more host connections than all other networks combined. In addition, cable and telephone network operators, as well as Internet service providers, are making a concerted effort to sign up residential customers with a variety of IP-based data services. Access technologies such as cable modems and Asymmetric Digital Subscriber Line (ADSL)[MINO95] offer the potential of providing millions of residential customers with *broadband* access, with bandwidths ranging from 384 kbits/sec to 10 Mbits/sec downstream, and up to 1.5 Mbits/sec shared upstream.

Ubiquitous availability of broadband connectivity over the next few years will cause a major paradigm shift in industries such as retail merchandising. Consumer services such as Web-based interactive shopping [ONLI96, PATH96] and home banking [WELL96] are already available, others such as on-line digital photo-finishing and network delivery of customized music, may follow. The standards-based multi-service Internet infrastructure offers a cost-effective alternative to proprietary Electronic Data Interchange (EDI) systems [LIMS] for inter-business transactions. American companies already buy $500 billion worth of goods electronically each year, but Web-based electronic commerce offers even more potential due to its relatively low entry-cost. This has prompted several companies to experiment with Web-based inter-business solutions [BUSI96].

While a great deal of excitement exists regarding the potential of Internet-based services, the operational characteristics and *management* of these services have received very little attention. As these services become more business-critical, or as the Internet reaches more residential consumers who are less sophisticated technically and do not have the benefit of extensive IT departments, new needs will emerge for ease of use, predictability and reliability of services, posing unprecedented challenges for management technologies. Traditionally, much of the research and operational experience has been either in managing networks and network elements, or in managing components of a computing environment such as servers and printers. What is needed now is *service management*, which includes managing networks, servers,

application software, client devices and any other components critical to the service. In addition, these Internet services all cross administrative boundaries. In the case of a Web-based interactive shopping mall accessible by residential users, different administrative domains include the user's home PC, the Internet access provider, the shopping mall service provider, and the various retailers available through the shopping mall. Components in each of these domains must cooperate and function correctly for the interactive-shopping service to work. Management of these services, which we refer to as *federated service management,* is a challenge since only a small part of the environment can be observed and controlled by any given authority.

Customer support is a critical but often neglected aspect of management that is particularly difficult in a federated environment. Our experience in working with a distributed system to deliver high-speed data services to the home has been that good customer support is very important to determining the customer experience with complex Internet services, particularly for relatively unsophisticated users. Borenstein et al., [BORE96] observed in their experience of running First Virtual Holdings, an Internet-based electronic commerce service company, that the biggest unexpected problems centered on customer service, and that "an Internet-savvy customer service department is an absolute prerequisite for anyone providing commercial services to the net". While good customer support is essential, economics for the mass market dictate that customer support departments cannot rely solely on human expertise to handle the growing complexity as well as increasing numbers of users. Our research is aimed at identifying technological support to help simplify the complex problem of customer support for federated services.

The goal of this paper is to articulate the challenges facing service management in federated systems, detail the issues surrounding customer support, and describe our work to date in this area. In Section 2, we discuss examples of federated systems in everyday use, such as the telephone network and the Automated Teller Machine (ATM)/Point of Sale (POS) network, and contrast why the challenges are more severe in the kinds of systems being envisioned for the future. In Section 3, we describe a type of federated system being deployed today to provide broadband interactive data services (BIDS) to residential customers. In Section 4, we outline our experience in providing customer support for a trial deployment of BIDS. We have implemented a system to aid in testing and diagnosis for customer support; this system is also briefly described in Section 4. We used this experience to derive a blueprint of an architecture and requirements for a federated management system, which is discussed in section 5. We end in Section 6 with a summary and future work.

2. Federated Systems

We first define the terminology used in this article. A *federated system* is defined to be a system composed of different administrative entities cooperating to provide a service. A *service* is an application with a well defined interface and functionality. *Federated service management* is the management of services that span multiple heterogeneous control domains, and which rely on correct functioning of components across those domains. A *control domain* is defined to be an administrative domain that is managed by a single administrative entity, typically a business. There are several successful examples of operational federated systems offering networked services, including the telephone network, the ATM/POS network, EDI systems for inter-business movement of supplies and products, and the Federal EFT network [JUNC91]. In this section we focus on the public, open systems such as the ATM/POS network and the telephone network, rather than the private, closed EDI systems, and extract the characteristics that render these systems manageable and reliable. We then contrast these characteristics with those of the future Internet-based services.

The federated systems that are currently operational typically have a small number of different types of business entities that participate in the end-to-end service. For example, the different control domains for an ATM/POS network are the retailer's store, the retailer's headquarters, the retailer's bank, the switching organization such as Interlink which serves as an intermediary to route transactions, and the customer's card-issuing bank [PERR88]. The POS transaction flows from the store to the headquarters' computers where it is consolidated and sent to the retailer's bank. If the customer happens to have a card from the

same bank, the transaction is completed; otherwise the retailer's bank sends a message to the switching organization which sends a message to the customer's bank which completes the transaction. Standards mandate all aspects of the system from message format to physical thickness, size and embossing of the cards. The American Bankers Association (ABA) publishes interpretations of the standards for POS debit systems, along with the responsibilities in a POS system of card-issuing banks, manufacturers of terminals, retailers, retailers' banks, and switching organizations.

Similarly for the telephone network, the entities that collaborate to set up a call are well defined. For example, if a customer makes a call from New Jersey to a corporation XYZ with a private PBX in California using an AT&T calling card, the participants involved include the Local Exchange Carriers (LECs) for New Jersey and California, AT&T as the Inter-Exchange Carrier, and the corporation XYZ. All interactions between various control domains and pieces of equipment have been specified in documents such as the LATA Switching Systems Generic Requirements (LSSGR)[BELL96]. In addition to detailed requirements, there are stringent certification procedures for equipment suppliers who build to these interfaces.

Both of these federated systems are highly reliable and provide a high degree of customer satisfaction with respect to support. They share a set of characteristics that make this possible, or at least easier to provide, than the Internet-based services that are now becoming available.

Each of these systems was created for the purpose of providing a *single service*[1].

- All the system components and their interfaces were designed and implemented with a priori knowledge of the single service they enable.
- A common understanding of semantics exists in all interactions between domains.
- Standards prevail at all levels in the system, including the application level.

Regulatory policy guides the development and operation of each of these types of systems.

- The legal responsibilities are clearly defined for all participants. Bodies such as the ABA, specify these responsibilities for the ATM/POS network. A host of policy-making bodies (federal and state regulators, legislatures, and courts) make decisions that collectively form a set of publicly-known policies for the phone companies.
- A very high level of reliability, and by implication testing, is imposed on these systems.
- There are strict certification procedures for the suppliers of system components. This limits the total number of vendors in the market place.

These systems were developed during a time of *minimal competition* for the provision of services.

- They had the luxury of a fairly long maturation process that lead to a large and stable system core.
- The cost of poor reliability was deemed to be sufficiently high to make high reliability a design goal of the system.

In contrast, the kinds of federated systems that are being deployed today to support services such as Web-based interactive shopping, inter-business electronic commerce, and digital on-line photo-finishing, exhibit characteristics that differ radically from those listed above.

These systems are *multi-service systems*, where the exact mix of services being provided changes with time.

- The fundamental premise of Internet-based services is to reuse an existing general purpose infrastructure for new applications.

---

[1]At least as originally designed. The phone network has grown over the years to offer many more services than a basic phone call, but the new services are derived from or provide enhancements to the original service.

- Although standards exist at the lower levels, such as the Internet Engineering Task Force (IETF) and Worldwide Web Consortium (W3C) standards for transport level protocols and Web interaction protocols, there are no application level standards, nor commonly accepted application semantics.
- The configurations of participants change and grow very rapidly as new business opportunities are explored for offering new services. For example, participants offering services such as Web hosting, or search engines are relatively recent phenomena.

The *use* of Internet is *completely unregulated.*

- There are no legal precedents for service contracts among participants. It is not even clear who can serve as an appropriate regulatory body to oversee compliance with service contracts.
- No testing and reliability requirements are being enforced so far on service/component providers.
- The myriad choices for equipment and vendors, with general purpose, off-the-shelf equipment being configured into new and complex systems, impede the application of certification procedures.

The pace of evolution of Internet-based services is frenetic and marked by *intense competition.* It is has been said that the pace of change on the Internet should be measured in "dog-years".

- The opportunistic nature of the application domain results in transient services and service providers.
- The rapid pace often results in very short product life-cycles and low levels of testing due to time-to-market pressures.
- Reliability is not yet considered a primary design goal.

We believe that the remarkable dynamism of the Internet era precludes the use (at least in the short term) of careful specification and engineering, and standards based construction and operation of service delivery systems that characterized federated systems in the past. Fundamentally, Internet services lack a coherent, top-to-bottom, end-to-end system architecture to serve as a blueprint for evolution. On the other hand, these new services, which started out as novelties, are rapidly becoming mission critical. Therefore, the businesses trying to sell these services to consumers must provide high levels of predictability and ease-of-use, to prevent customer frustration and rejection. We must concentrate research and development efforts in improving the operational characteristics of this new class of federated systems, and in providing solutions for federated service management.

3. Broadband Interactive Data Services (BIDS)

The imminent deployment of BIDS and BIDS-like systems has motivated our research into federated systems. The BIDS systems are emerging due to two phenomena: an ongoing effort to connect homes via broadband networks, and services motivated by the availability of high bandwidth, such as video-on-demand, videophones, home shopping, and lately high speed Internet access and world wide web access. These services are targeted towards residential customers and are referred to collectively as Broadband Interactive Data Services or BIDS. They are federated services since the underlying system is composed of multiple independently managed systems that cooperate amongst themselves to furnish a service. Since these services are targeted towards non-technical customers, effective and efficient operational support is critical to their success.

3.1 BIDS Architecture

The general architecture of a BIDS system is shown in Figure 1. Independent domains are drawn as boxes with dotted lines. A subscriber interacts with the system via an access network. This can be an HFC cable TV network, an ADSL or ISDN network provided by the local telephone company, or a Satellite or Wireless network. The access requires hardware such as PCs and modems and appropriate protocols to talk to the headend (the terminating point for the access network). The user is connected via a local access network to a server complex (shown as the Application Service Provider Domain in Figure 1). The server complex has the necessary infrastructure for managing access to the access network, which includes managing subscription, billing, and security for the subscribers. It addition, the server complex can also provide content to the subscribers (such as a community Web server or bulletin board), and access to the

Internet, other networks (online service providers), and services provided by those networks. The headend converts signals from the local access network onto the high speed network local to the server complex.

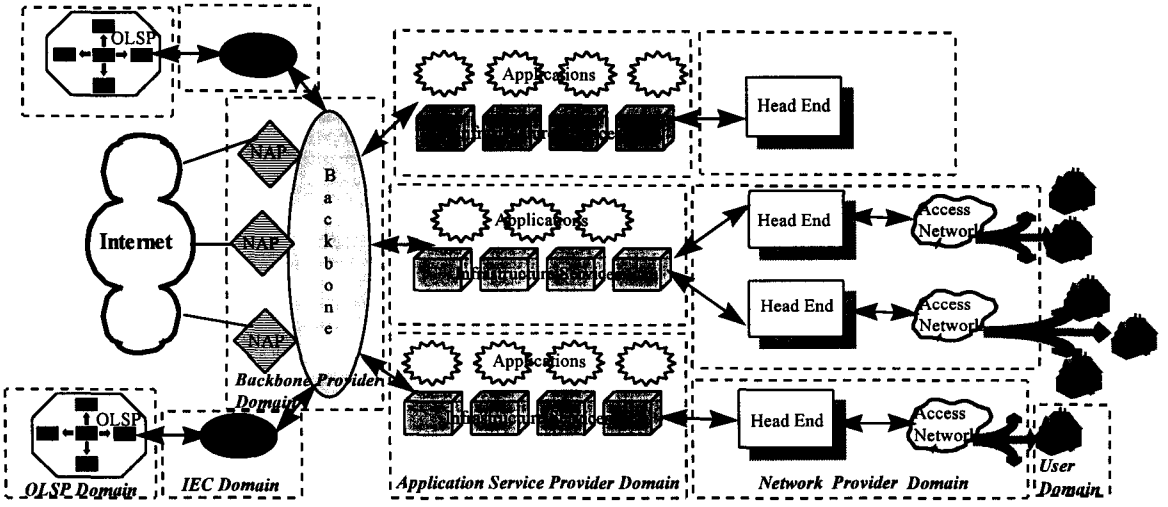

Figure 1. The Architecture of BIDS

Each domain is independently administered and has unique characteristics that can change independently of other domains. New domains can appear and old ones disappear dynamically, since there is no central control across all the domains. Within a single domain, new or different services can be introduced independently of other domains. The complexity within each server complex can also vary; it can be a distributed system connected via a high speed backbone network such as the Asynchronous Transfer Mode (ATM) or Fiber Distributed Data Interface (FDDI) or just one standalone machine. The hardware and the middleware can differ in each server complex and can be upgraded or changed dynamically. The server complex may also host content servers such as video servers, and local web servers. The hosts and services can be replicated to provide high availability and reliability of services. Thus, there is considerable variety and complexity underneath the BIDS architecture.

A customer interacts with the system via a myriad of services, such as a web browser, email, telnet etc. Successful functioning of these applications is dependent on multiple system components. For example, to look at a web page on a web server on the Internet, a user must go through the following steps. He must install the authentication and the browser application on the PC. The software must connect to the subscription manager via the modem, the access network, and the server complex high speed network, to obtain the right credentials for access. Subsequently, the browser accesses the Internet by going through the Internet gateway and the firewall server. Problems with any of these components can cause the browser application to fail to behave as expected by the customer.

Thus, the BIDS architecture is a federated system that spans the Internet and the online service providers.

1. BIDS is multi-service: a large number of services are offered to the end user that span across components and administrative boundaries. The future usage and composition of these services cannot be predicted.

2. BIDS is unregulated: the components of BIDS are in different administrative domains and furthermore, the administrative boundaries can evolve over time. No central authority controls all the domains.

3. There is intense competition: anyone can furnish a service and try to compete with the existing providers of a similar service. The presence of a large, dynamically changing collection of heterogeneous components introduces considerable complexity in the system.

One such framework, called HP-BIDS, has been developed by Hewlett-Packard and deployed on a trial. It is a representative example of federated systems that are likely to emerge in the future to provide high bandwidth data services to homes. The system as actually deployed, is slightly less complex than what is possible under the architecture. There is one access network which is an HFC cable TV network. There is only one server complex and all the content servers in the server complex are in the same administrative domain and hence are under the control of the same management authority. However, there are content providers in other administrative domains on the Internet and within the networks of the online service providers. A recent trial deployment of the HP-BIDS system has given us the opportunity to study the problem of customer support for a federated service, namely broadband access to the Internet. The trial has been going on for approximately 11 months at this point, and the experience gained from it has yielded several useful insights into the issues of diagnosing customer reported problems and dealing with multiple control domains in the process.

4. Management of BIDS

Managing services in an Internet-based federated system such as BIDS is a challenge. In what follows we limit ourselves to the problems faced in diagnosing faults and doing customer support in BIDS and BIDS-like federated systems. This choice is deliberate since the success of the BIDS infrastructure will depend on providing satisfactory customer service to the end-users who are not necessarily technologically savvy. It is also an issue that has been largely ignored by the Internet community with a few exceptions [BORE96]. Borenstein et al., point out that customer support is a critical component of the framework for successfully implementing a payment system on the Internet. This is largely due to the Internet inexperience of a typical user as well as fact that the Internet is not centrally controlled. Customers have difficulty distinguishing between genuine failures and deliberate fraud on part of the participating sellers. Therefore, an Internet Intermediary who has a deep understanding of the Internet, may be required to provide customer support in those cases.

A related concern is whether there is one organization that supports all the services accessed by a customer via BIDS, or a separate organization for each component of the service. From our experience, it seems that there must be one point of contact for customer support for all the services that are accessed by a customer via the access network. This is desirable from the customers point of view, since a customer (who is not technically sophisticated) does not want to be confronted with the scenario where each service provider passes the responsibility of problems to other service providers in the system, whom the customer must call in turn. From the point of support it is also more effective for diagnosis to have an organization that has the overall view of the system as opposed to just one control domain. This provides and end-to-end view of the federated service. The experience of Borenstein et al., bears this out as well. It remains to be seen however, if the same model will scale when the number and variety of available services increases beyond a certain threshold.

4.1 Difficulty of Diagnosis

Diagnosing services provided by a BIDS-like systems is difficult because the underlying system has a unique configuration with respect to the mix of components, no prior operational experience, and little understood failure modes. Due to complex unforeseen interaction between different parts of the system, the failure modes that are observed are very diverse, unexpected, not very intuitive, and show no single dominating factor responsible for malfunctions. Figure 2 shows the actual distribution of faults that we observed in the initial deployment of HP-BIDS over a period of 3 months. The faults observed ranged from network problems, application problems such as email, user errors, problems with online service providers, PC configuration problems, and several others outside those categories.

Since the mix of components is varying, dynamic, and not under central control, interactions between them will continue to be unpredictable. The practical implication of such interactions is that the symptoms of a problem can appear far away from the problem itself, both in space and time, and may have no apparent correlation with the problem. On the other hand, the complaints are stimulated by the failure of services being used by a customer. Diagnosing these services requires information about all the components that a service relies upon, which may not be available if the components are in a domain that does not export this information outside the domain. Yemini et al., [YEMI96] have discussed similar phenomena in the context of large scale heterogeneous networks, focusing on the problem of alarm correlation.

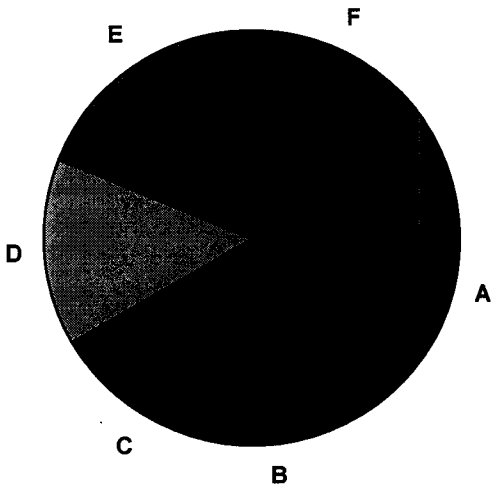

Figure 2. Distribution of Faults in HP-BIDS Trial

4.2 Diagnosis Experience in HP-BIDS Trial

In the deployment of HP-BIDS trial, there is one point of contact for customer support for all the services that are accessed by a customer via the access network. A customer calls this support organization when an application that accesses the network in any way fails to perform according to customer's expectations. The customer support contact tries to identify and diagnose the problem by asking the customer questions, testing different pieces of the whole infrastructure, including the PC configuration, and customer access rights. Testing can done directly by the customer support if they have the necessary access, or on behalf of the customer support by the administrator of the domain within whose purview lies the component. This is a typical technique for doing customer support in the industry, but we discovered that it is too inefficient for the BIDS environment and impossible to scale to a large customer base.

We give two examples that illustrate the spectrum of problems that were encountered. One of the problems that the customer support had to diagnose was a call from a customer whose FTP software (a utility for transferring files across machines) stopped working after he had installed a new version of the FTP on his PC. The problem was traced to a temporary network failure, but not before the customer support had checked the PC configuration, the cable modem, the new version of the FTP that was installed, the state of the subscriber account, and the availability of the servers to which the user wished to connect. Sometimes this had to be done on machines in different administrative domains, which required phone calls to the managers of those domains to get the right information.

A second problem that occurred quite often was a deterioration in performance of applications that used TCP on the HFC access network between a particular PC and the servers in the server complex. It was traced to noise ingress from another user's home (someone switching on a hair dryer, for example),

located downstream on the tree structured cable network. The noise interfered with upstream data traffic and corrupted the acknowledgment packets. TCP interprets loss of acknowledgments as an indication of packet loss, and packet loss to be the result of congestion. When faced with congestion, TCP slows down and retransmit packets, resulting in decreased throughput. The lower throughput propagated to the application level and the applications started timing out. This was seen as a disruption of service by the user. Since upstream channels were not used before, there was no experience with their reliability and actual throughput. The problem arose because a protocol was used in a context for which it was not designed. The solution to this problem consists in installing filters at homes to limit ingress. Once that is done however, the distribution of faults will also change.

All the diagnosis procedures employed to deal with such problems are manual, error prone (such as asking the user to read the parameters from the PC configuration file), and time consuming. If the system has tens of thousands of subscribers, the customer support model will break down, both from the point of view of customers, who will perceive a poor quality of support, and from that of the organization providing the customer support, for whom the process will be very expensive.

In what follows, we first describe a diagnostic server that helped automate some of the diagnosis and trouble shooting procedures in HP-BIDS. This is a good short term solution for small to medium size systems and consists in automating the invocation of diagnostic tests within and outside the local domains. The experience gathered with the HP-BIDS diagnostic server is used to derive requirements for a more general service-contract based architecture that facilitates diagnosis. We feel that in the long term, services over federated systems should be constructed according to that architecture.

4.3 Diagnosis in HP-BIDS

4.3.1 Design Constraints and Goals

For reasons discussed earlier, customer support should be provided by one organization. In case of the HP-BIDS trial, customer support is provided by a third party that is different from the service providers (the cable company and the organization that controlled the server complex). The customer support organization has no direct access to the server complex hosts and yet is given the responsibility of responding to complaints.

The customer support organization should have access to all involved administrative domains via standard interfaces agreed upon beforehand. Each domain may export via these standard interfaces, the ability to test, monitor, or control that domain. In HP-BIDS the customer support organization has PC expertise and can help the subscriber determine whether or not a PC configuration problem is the source of the complaint, but beyond that, they have to call support personnel in the cable plant or server complex to do the troubleshooting.

It is also desirable to support both use of active testing as well as passive monitoring. For example, in HP-BIDS certain aspects such as the state of the login session and the subscription account of a subscriber have to be checked in real time. These are specific to a subscriber and unlikely to affect others. A failure of a low level system component on the other hand can cause multiple applications to fail and can affect hundreds to thousands of users. These problems should not be treated as isolated and independent. Instead, information about them should be shared across trouble calls to make diagnose more efficient. Periodic monitoring of the system, either by the service provider or by the customer service organization can reveal these problems even before they are reported by the customers.

The goal of a diagnostic tool is to enable the customer support engineer to rapidly localize the organizational domain within whose purview lies the source of the problem. This allows faster resolution of trouble calls for the customer, less expense for the customer support process, and fewer irrelevant trouble calls for uninvolved domains. In HP-BIDS, since the customer support organization does

not, however, have the authority or the ability to change the state or fix any problem, the tool need not attempt that either.

4.3.2 The Diagnostic System

The goal of our diagnostic system is to isolate the source of a problem rapidly and efficiently without requiring administrative rights to all the components. It does so by remote testing of different components of HP-BIDS using agreed upon interfaces exported by each component. The meanings of the test results are also agreed upon beforehand. The choice of the tests and the order in which they are to be executed is determined by the study of the history of problem call records.

Figure 3 shows the architecture of our diagnostic system. The customer support engineer uses their desktop computer to interact with a remote diagnostic server. The engineer orders a set of diagnostic tests to be run, and the diagnosis system invokes management interfaces in each of the components relevant to the requested test. The results of each test are returned back to the support engineer.

The diagnostic server has two parts, a diagnostic engine that receives requests from the support engineer and dispatches the necessary tests, and the system specific tests themselves. The system specific tests may in turn invoke diagnostic servers in another domain if necessary. The diagnostic engine understands the relationships between the tests and their input and output data, and can order the tests to be run such that the output of one test can be used as input for a later test.

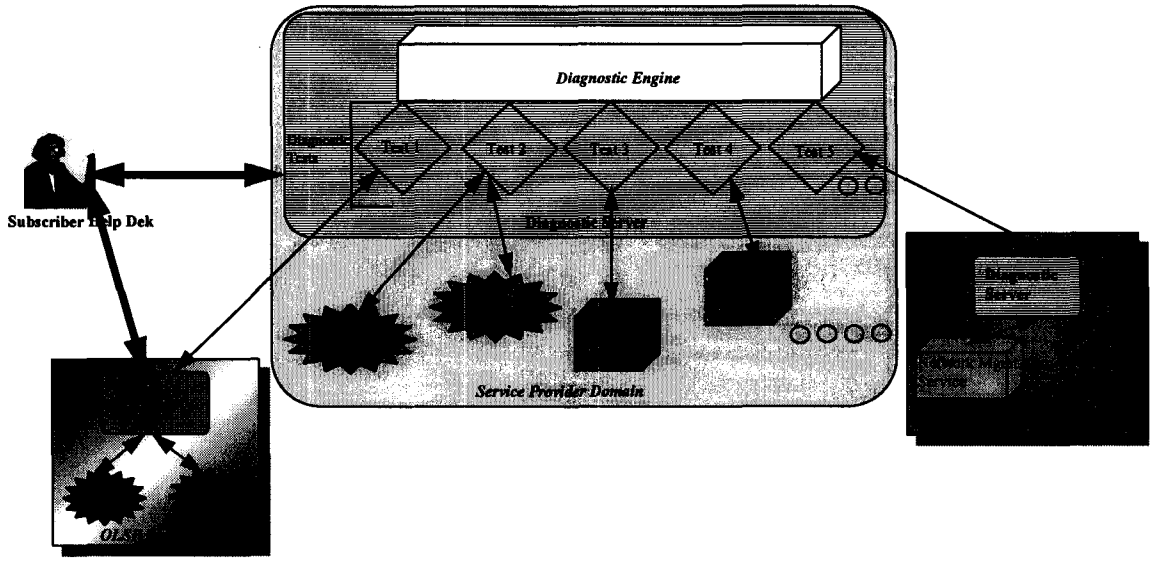

Figure 3. Details of the Diagnostic Server

It should be noted however, that a customer support engineer must understand the system well enough to know which tests are relevant to a customer's problem. In other words, a human troubleshooter is remains a critical component of the diagnosis process.

Our experience with the design, implementation, and deployment of the diagnostic server leads us to believe that one point of contact for customer works very well. It has lead to general diagnostic interfaces that hide low-level system details, and that do not change as the system evolves. These interfaces were derived as needed. A testing infrastructure was also developed to verify component compliance, to take into account dependencies between tests, synchronization of test results, and the security requirements. The diagnosis server does not however, address the problem of automated diagnosis across domain boundaries. Nonetheless, even such a simple tool is an improvement over manual procedures and will cut down

enormously the average time required for customer support. As a result, it will be deployed as a product and will be used internally by HP response center engineers for supporting the BIDS server complex.

5. A General Architecture for Diagnosis and Customer Support

5.1 Facilitating Diagnosis

If BIDS-like systems become commonplace, more and more federated services will be deployed over domains (such as the Internet) over which the service providers have no control. Organizations that provide support for these federated services will find themselves charged with diagnosing problems with these domains as well, in addition to diagnosing the services they provide directly. This dilemma arises from the conflict between the desirability of providing one point of contact for customer support, irrespective of the underlying component services, and the inability to control those components themselves.

We are developing an architecture that will reduce the complexity of the diagnosis process. In what follows, we give a very high level overview of this architecture and our rationale for the design choices. Even though this work is quite preliminary and the details are still being worked out, it offers useful guidelines for design of federated services to facilitate diagnosis and customer support, where none exist so far. We continue to validate and refine this architecture against our experience of HP-BIDS and similar systems.

Any such architecture must take into account the unique nature of the new federated systems introduced in Section 2. A large number of federated services are being deployed rapidly that are not being developed to any standards, since there are practically none that exist now, and the emphasis is on early time to market. Extensive reuse of existing and available components to build other services is reducing costs and time for development, but as a result building blocks are being used in a way that they were not designed to be used (for example, use of TCP over HFC). Emphasis on speed of deployment is creating a high variability in the quality of deployed services. And there are no well defined ground rules or precedents to guide the design of new services.

These factors motivate the following design decisions.

1. Consider services individually as opposed to looking at the entire system. This implies separating the diagnosis requirements of a service from those of the entire system.

2. Separate out the requirements of diagnosis and customer support, which are similar for various services, from the details of the functionality being offered by the services, which can vary a lot.

3. Specify the expectations of a service in the form of contracts. This delineates the exact responsibility of the service towards its users and is specified per service. These contracts allow us to deal across the administrative boundaries.

4. Provide an infrastructure for verifying compliance with these contracts and a trusted third party that can arbitrate in case of conflicts.

5. Furthermore, make the design technique used to facilitate diagnosability of a service applicable recursively, implying that the components that are used to construct a service can be constructed using the same techniques.

5.2 Representing Services in BIDS

The principal ideas of our architecture are brought out by the following example. A service in BIDS is represented by an abstract model, Figure 4, which is elaborated below. At one level there is a service such

as the Web Service, Email, FTP etc., (referred to as a top-level service) that is provided to a user. A top-level service is the entry point for the user into the system and user interacts with the system by the means of this service. In the examples given before, Web Service is a top-level service that a user employs to access the net. This service itself is put together by active collaboration and participation of other services (referred to as component services), which offer more primitive services. In case of the Web access, these are the local access network, the high speed network in the server complex, the session manager, the web server, and perhaps even the PC operating system. The user is not exposed to these component services and does not care what these services are and how they affect the top-level service. Furthermore, each of the component services can itself be recursively composed of other more basic component services and a *federator* for those services. Thus for example, the Web Server can be composed of a jukebox, a server machine, and a high speed bus that connects the two together.

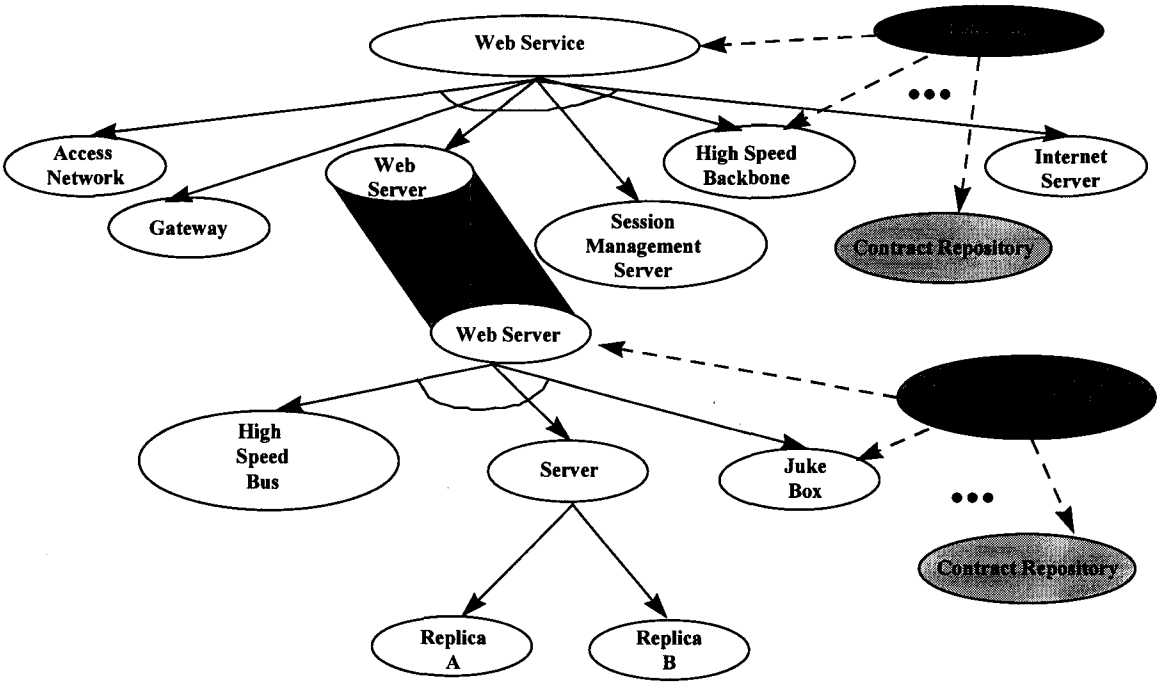

Figure 5. Dependency Graph for Web Service

Providing effective customer support requires coordination across all component services. This task is accomplished by the *federator* (a generalization of the HP-BIDS diagnostic server). The role of a federator can be assumed either by one of the service providers, by the manager of any of those systems, or a third party. The federator is only required to know the dependencies of a top-level service on its component services (the next level down) and to have access to the contract verification procedures , and possibly contracts themselves. These dependencies are shown as solid edges in the AND/OR dependency graph. The rectangular boxes on the side of the ovals, each of which represents a service, indicate the contract verification interface. The broken arrows represent access by a federator.

The rest of this discussion shows informally how the model impacts the process of provisioning a service, deploying it, and providing customer support for it. It is aimed at giving a flavor of the design since the details are still being worked out.

5.3 Service Provisioning

A service provider or the system integrator must locate (or construct) the suitable components, understand their functionality, compose them and configure them to construct a new top-level service, and deploy the service making it available to users. This task is arbitrarily complex if the components are numerous, varied, and not originally conceived for interoperability with each other. The integrator must also provide the missing functionality where needed. Once the service is provisioned, it is turned over to a federator for management and possibly control. Thus in Figure 4, a web service is provisioned by integrating multiple networks, servers, session managers, firewalls and gateways.[2]

Each component explicitly specifies in the form of contracts what is being offered by that component (type and quality of service), what requirements must be met to access that service, the mode of billing, and some tools to verify and possibly enforce compliance with the contracts. These contracts are not a complete description of the service but an abbreviated model that captures the characteristics that are useful from the point of view of diagnosis. The contracts are represented as auxiliary interfaces offered by each component service to the federator. In time, the core of these contracts can become standardized for certain types of services, with a customizable component for special requirements.

Thus for example, if a server such as a web server is managed independently, there is a service contract between it and other components that use it, that specifies the availability and the reliability of the server, and the average response times. It also furnishes information (such as abbreviated system logs) to verify compliance with its service contract. The details of our contract based architecture will be described elsewhere and are outside the scope of this paper. A similar notion of contracts has been developed under ANSA [HOFF93], though not from the point of view of diagnosis and customer support.

The service provider must also construct a structural model of the top-level service that gives the dependencies of the top-level service on its component services. This dependency graph (shown in Figure 4 by solid arrows) is an AND/OR graph [LUGE93] and does not capture the exact interfaces that each of the component offers to other participating components. It is however, required for effective diagnosis of top-level services. From that graph it is clear that the web service depends on the access network, the web server, etc., for it to function correctly. Same style of reasoning can be applied to each of these components. For example, the web server can be constructed using two redundant servers, a jukebox for storing content, and a high speed bus that connects everything together. Thus, it is dependent one of the two servers being functional as well as the jukebox and the high speed bus operating correctly. Since these components are likely to be in the same administrative domain, the system manager can play the role of a federator.

5.4 Customer Support and the Federator

Once a service has been provisioned and deployed, the responsibility of providing customer support is turned over to the federator (who can be the system manager). The federator can also be responsible for other aspects of service management, such as configuration, performance, and planning.

The underlying assumption of the model is that problems arise if a component service does not or cannot fulfill its contract or due to circumstances that are outside the purview of the component services. Thus when a federator receives a customer report of a problem, it tries to identify the service with which the

[2] Some aspects of provisioning can be automated. The problems of locating the right services and understanding what they offer has been looked at in several trading and brokerage models in the context of object management [ITUT95] and service architectures for multimedia [NAHR96]. Cohrs and Miller [COHR89] have looked at the problem of specifying and verifying the configuration of components in a federated system. These provisioning issues are beyond the scope of this paper but are mentioned for sake of completeness.

customer is experiencing problems. Then it tries to determine if the customer is meeting the requirements of access (has the right security and access profile, has paid the bills etc.). If the customer meets the required constraints, the federator systematically verifies that each component service is complying with their contracts. This requires executing the service provided contract compliance verification procedure that hide the internal details of the service. If a component is found to be in violation of the contract, the federator reports the details of the failed compliance test and the particulars of the problem to the faulty component. If the federator has the necessary rights and understands how that problem can be fixed, it can fix it itself. Otherwise, it delegates repair and any further diagnosis and tracks its progress through a trouble-ticket like system. The verification process is driven by the knowledge of how the service is composed and statistics on distribution of faults, the historical experience with the component services, or the reliability data on component services.

The faulty service component (more precisely its internal federator) can repeat the same process to localize the fault to its component services. For example, a top-level federator may notice that the network is dropping too many packets. If that network is in a different administrative domain, it notifies the entity responsible for managing the network of the problem. This entity subsequently tries to localize the source of the problem (whether it is the hardware, protocols, or something else). The top-level federator monitors the progress of the problem/repair and informs the user when the problem is fixed.

The service contracts and structural information can reside in a repository that has controlled access. This enables sharing of the information between multiple federators. Multiple federators can exist in the system simultaneously since anyone who has access to service contracts and structural information about a service can act as a federator, and the total number of services can be large and continue to change and evolve. Furthermore, there can be multiple federators for same top-level services, each of whom is distinguished by its diagnosis technique and the efficiency and accuracy of diagnosis.

The preceding description sketches a blueprint for a service-contract based architecture. Despite its preliminary nature, our early experimentation with expressing service contracts in HP-BIDS and describing diagnosis procedures in terms of contract verification procedures has been positive.

6. Summary

There is a trend towards deployment of BIDS and BIDS-like systems to provide broadband access to homes and businesses to enable services such as interactive shopping, home banking, electronic commerce, and access to World Wide Web. These systems are federated but distinct from the existing federated systems such as the ATM/POS systems and telephone networks. The emerging Internet-based federated systems span multiple control domains, lack any existing service level standards or system architecture, and offer a dynamic mix of services.

Diagnosing faults and providing support to the customers of federated services is critical to their success but extremely difficult to accomplish, since only a small part of the environment can be observed and controlled by any given authority. We have characterized this problem, using our experience with the deployment of HP-BIDS. The HP-BIDS experience established the importance of a single point of contact for providing customer support to users and for doing diagnosis. It also revealed the desirability of automated diagnostic procedures that can be invoked across control boundaries.

We then described a customer support tool that was developed to automate some aspects of diagnosis in BIDS, and used the experience with this tool to derive requirements for a general service contract based architecture that can help ease the problem of diagnosis and management of federated services. We introduced the notion of a federator in this architecture and showed how the federator can diagnose the federated services by checking each component service for compliance with their contracts. This technique can be used recursively within each control domain to achieve the level of detail required in diagnosis.

Work on the service-contract based architecture is ongoing. Open questions include different levels of transparency of service contracts, complexity of describing dependencies and fault models for large systems, and guaranteeing consistency of contracts. Other aspects of the HP-BIDS diagnostic server are also being generalized, including testing support, alarm correlation for service management, and automatic generation of fault models for federated services based on knowledge capture.

The original contributions of this paper consists in formalizing the distinction between the new federated systems and the pre-existing systems, documenting the experience of managing BIDS-like systems from the point of view of diagnosis and customer support, and deriving a set of architectural requirements for federated service management based on this practical experience.

Acknowledgments

We thank Gary Herman, Rich Friedrich, and Ed Perry for comments and feedback, and Ellis Chi for helping with diagrams and editing.

References

BELL96 LATA Switching Systems Generic Requirements, 1996 Edition, http://www.bellcore.com.

BORE96 Perils and Pitfalls of Practical Cyber Commerce, Borentstein, Nathaniel S., et al., Communications of the ACM, June 1996.

BUSI96 Invoice? What's and Invoice?, Business Week, June 10, 1996.

COHR89 Specification and Verification of Network Managers for Large Internets, Cohrs, D. L., Miller, B. P., ACM SIGCOMM89, September 1989.

GIFF85 The Cirrus Banking Network, Gifford, D., and Spector, A., Communications of the ACM, August 1985, Volume 28, Number 8.

HOFF93 The Compatibility of Objects in Distributed Systems, Hoffner, Y., Linden, Rob van der, Beaseley, M., ANSA APM.1066.00.02, October 1993.

ITUT95 Open Distributed Processing Reference Model, Technical Report ISO/IEC 10746-3, 1995.

JUNC91 A Primer for Settlement of Payments in the United States, Juncker, G. J., Summers, B. J., Federal Reserve Bulletin, November 1991.

LIMS EDI - An Overview, Li, M. S., IEEE Colloquium on Standards and Practices in Electronic Data Interchange, Digest No. 106.

LUGE93 Artificial Intelligence: Structures and Strategies for Complex Problem Solving, Luger, G., Stubbelfield, W., The Benjamin/Cummings Publishing Company, Inc., 1993.

MINO95 Video Dialtone Technology, Minoli, Daniel, McGraw-Hill, Inc. 1995.

NAHR96 The QoS Broker, Nahrstedt, Klara, Smith, Jonathan M., IEEE Multimedia, Spring 1995.

ONLI96 Online Shopping Network, http://www.ll.net/osc/Virtshop.htm

PATH96 Pathfinder, http://www.pathfinder.com

PERR88 Electronic banking goes to market, Perry, T., IEEE Spectrum, February 1988.

WELL96 Wells Fargo, http://www.2digm.com/resume/wells2.

YEMI95 High Speed and Robust Event Correlation, Yemini, S. A., Kliger, S., Mozes, E., Yemini, Y., Ohsie, D., IEEE Communications Magazine, May 1996.