Remote Management Services Over the Web

Mohamed Dekhil, Vijay Machiraju,
Klaus Wurster, Martin Griss
Software Technology Laboratory
HP Laboratories Palo Alto
HPL-2000-59
May, 2000

E-mail: dekhil@hpl.hp.com

service management, Web-based services, remote management

Enterprises in today's world are trying to create new value-added services by leveraging (i.e., outsourcing) services offered by other enterprises and focusing on developing and maintaining their own specialized services. The management of these services, which is critical to the success of every business, can be provided as another service that can be outsourced as well. The tasks of service management include monitoring the health of the service, controlling its behavior, and optimizing its end-to-end business operations and resource utilization. In this paper, we introduce the notion of remote management and discuss the requirements for providing dynamic end-to-end remote management for internet-based services. We then describe a prototype demonstrating a web-based remote management service that uses an XML-based messaging scheme to communicate and exchange management and control information between the management system and the managed services and to provide customized views of the state of the managed service.

1. Introduction

The Internet has enabled the emergence of new media for exchanging information, offering services, conducting business, and communicating with one another. A new generation of web-based 3-tier applications has transpired to provide a mechanism for creating new services and transforming old ones to be delivered over the Internet. Service management is the discipline of monitoring the health of the service and controlling its behavior to optimize its usage and maintaining it in a desired state. It also has to deal with all the issues of network [1][2], system [3], and application management [4] [5], while raising the level of abstraction to focus on handling the distributed service components and their interplay. Traditionally, management solutions have been either retrofitted or tightly coupled with the managed systems. Factors such as economies of scale, flexibility in adding and upgrading management features, ability to deal with and manage the highly automated, dynamic, and federated world of internet-based services, and the lack of in-house IT expertise, are driving the need for creating outsourced management solutions Effective remote management of internet-based services requires a clearly defined interplay between the managed services and the management system. A well-managed service exposes sufficient information, measurements, and control points for use by the management system. A good management system uses this information to monitor the state of the service, to analyze and diagnose problems that occur within the service, and to control the service behavior. In addition, the management system provides views of the service state to the user, customized based on user roles and preferences. Figure 1 shows an overview of a remote management service setup. In this setup, the management service communicates with the managed service and provides the user with different views through a web-based management console.

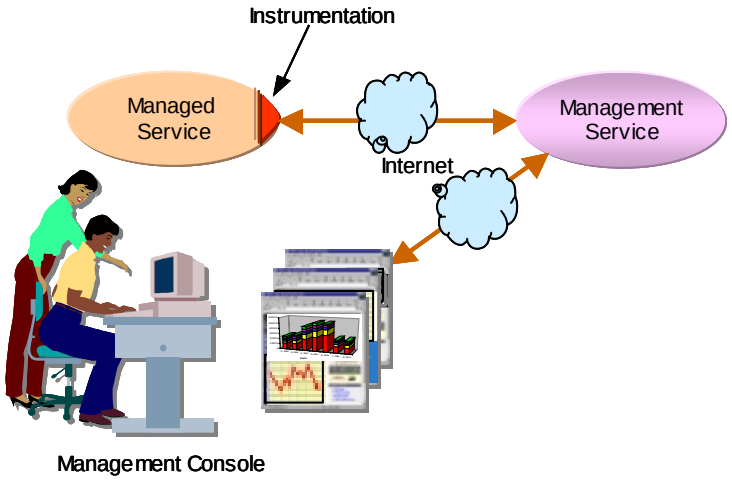


Figure 1: Providing remote management services

The rest of the paper is organized as follows: In Section 2, we discuss business outsourcing and its motivation and describe different models for management

outsourcing. Then, we introduce the notion of remote management as an outsourcing model and present some example of management capabilities that could be offered as remote services. In Section 3, we investigate the different technologies for instrumenting the managed service to provide the required management information and discuss the tradeoffs between the different approaches. An XML-based management vocabulary and communication protocols, used to exchange the management information between the managed service and the management system, are described in Section 4. In Section 5, we describe a management service prototype that illustrates the concepts and utilizes the ideas presented in the paper. Finally, we conclude in Section 6 with a summary and future research directions in this area.

2. Management Outsourcing

2.1 Outsourcing Business processes

Outsourcing is the process of delegating certain parts of the business to other enterprises. Some of commonly outsourced processes in current businesses include:

- Internet access to Internet service providers (ISPs),
- e-mail services to application service providers (ASPs),
- telecommunications to communication carriers,
- credit card authorizations and payments to financial institutions, and
- shipping to postal and parcel services.

For example, an online book selling business could use services provided by advertising agencies to handle their advertising strategies, credit card companies (e.g., Visa, American Express, etc.) to verify and authorize credit payments, and shipping companies (e.g., UPS, FedEx, etc.) to ship orders to customers (see Figure 2).

Figure 2: Outsourcing of business processes

More recently, new forms of outsourcing markets are being created turning the earlier world of enterprises into “virtual enterprises.” Functions that were generally considered to be the main constituents of a business - such as human resources and payroll - have also become candidates for outsourcing. The main motivation behind outsourcing is to focus on mainstream services that increase revenues and to reduce the expenditure in other areas. Check [6] and [7] for more information on Outsourcing practices.

2.2 Impact of Outsourcing on Service Management

Outsourcing of business processes has a profound impact on managing the provided services. Since portions of these services are provided by other enterprises, which are outside of the management domain of the service provider, an end-to-end view of the state of the service requires a cross-enterprise management solution. We call this *federated management*, which implies that the management solution should not be confined within the control boundaries of the managed service. This imposes certain requirements and pre-requisites on the management solution, which will be discussed later in the paper. In addition, outsourcing contracts usually contain some provisions on the quality of service provided by the outsourcer, known as *Service Level Agreements* (SLAs). To enforce these types of agreements, we need tools that measure the quality of the delivered services and evaluate it against the agreed upon levels. This adds even more requirements on the management solution [8].

2.3 Outsourcing Management Operations

Similar to other business processes, management functionality could also be outsourced to one or more external parties. The following are some of the reasons why enterprises would want to outsource management of their applications and services:

- **Expertise:** Service providers would like to focus on their main lines of business while leaving the job of managing their resources and services to those who are experts in management. Different outsourcers could develop different kinds of expertise in application and service management. For example, one outsourcer could specialize in back-up services while another could specialize in managing services offered by e-commerce vendors.

- **Economy of scale:** Enterprises have realized that management of their services and its underlying IT infrastructure is extremely critical to their success. Therefore, they are allocating huge amounts of their budget to IT and service management solutions. On the other hand, offering management solutions is itself the main line of business for management outsourcers. By specializing in certain types of management and by managing many enterprises and resources, outsourcers will be in a better position to offer cheaper and better management solutions to enterprises that need them.

- **Flexibility:** Businesses are evolving to meet new demands every day, which may involve changing applications, upgrading systems, or adding new services with the necessary infrastructure. Consequently, the management solution will have to be modified to cope with these changes. In addition, management solutions are also evolving to provide new functionalities and better capabilities. Outsourcing some of the management tasks enables service providers to be more flexible in keeping up with the changes, instead of allocating more resources to update their management infrastructure with every change in the provided services and the underlying applications and systems.

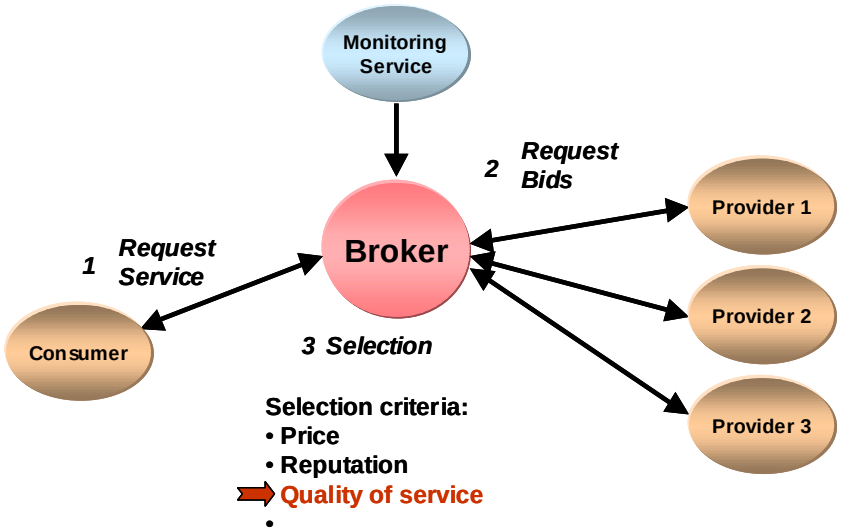
- **New types of management:** Outsourcing enables new types of management, some of which were not possible within the jurisdiction of a single enterprise. As businesses use each other's services to provide new services, end-user transactions often flow through multiple enterprises. Though every enterprise can manage its portion of the overall transaction, little information is exchanged across enterprises to manage the overall end-user experience. As outsourcers act as secure and trusted mediators across businesses, enterprises will be more willing to expose portions of their management information to outsourcers. Using this information, management outsourcers will be able to monitor and audit service level agreements between enterprises, provide an end-to-end view of the user transactions, and perform high-level analysis and diagnosis of the problems that occur.

2.4 Remote Management

There are several models of management outsourcing, ranging from on-site contracting, to remote management, to management as an Internet service. On-site contracting is the traditional form of outsourcing, which is widely used for network and application management. In this model, enterprises contract parts of their IT functionality to contracting firms that specialize in managing these systems. The contracting firm sets up an on-site management system that measures, monitors, and controls those applications. A more advanced version of on-site contracting is to manage applications and services remotely. Outsourcers can set up management systems and consoles that can remotely monitor and control the behavior of applications running in other enterprises, or even on home machines. If management is offered as services that can be dynamically acquired and integrated into a more complex service, we refer to it as remote management services. While custom protocols can be used for communicating between the managed service and the remote management system, the proposed management service model uses the Internet as the communication channel.
The following examples illustrate some of the management capabilities that could be offered as remote services:

- **Remote monitoring and reporting:** In the federated e-business world described earlier, in which parts of the service are outsourced to other enterprises, a single transaction may span multiple enterprises and enter different management

domains. A remote monitoring service can be set up to collect management information (e.g., transaction response time) from the different parties involved



and provide an end-to-end view of the transaction state. Service providers can use this information to verify the service level provided by the outsourcers. Consumers can use it to compare quality of service from different providers. And in a more advanced scenario, dynamic service brokers can use the information to select between providers, as shown in Figure 3.

Figure 3: Using management information to select between service providers

- **Remote backup service:** Outsourcers could offer disk or file-system backup as a remote service. Enterprises or even home-users can use this service to protect their data from accidental corruption. To provide such service, the outsourcer needs access permission to the file system to be backed up. Another solution is to install an agent on the consumer's site, which periodically performs backup operations and sends the contents to the remote site for storage. To retrieve files from the remote service, users log-on to the backup service site and download the required files.

- **Application garage:** Outsourcers specializing in the management of particular applications (such as SAP or Microsoft Exchange) could open "garages" that contains a set of diagnostic and repair tools for isolating and fixing certain problems that occur in those applications. Consumers can access the services of these garages on the Internet whenever they encounter problems with their applications. Agents that run diagnostic tests are automatically downloaded into the enterprise or onto the home-user's machine. Data that is collected by these agents could either be analyzed locally (in simple cases), or else it could be sent back to the outsourcer for advanced diagnosis.

3. Service Instrumentation

To enable remote management, the managed services should provide the necessary information and control hooks to the management service through a set of pre-defined interfaces. *Instrumentation* is the term that is commonly used to refer to the infrastructure that has to be included in the managed service in order to support integration with a management system (see Figure 4).

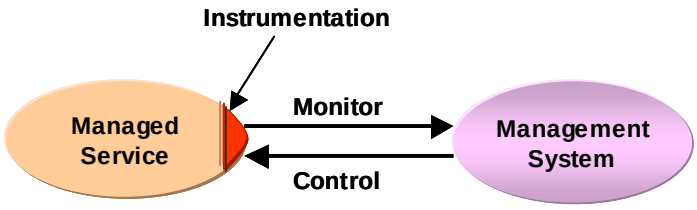


Figure 4: Service instrumentation

Instrumentation cinsists of implementation code that has to be included with the managed service (either invasively or non-invasively) in order to collect and export models and raw measurements, to signal and throw interesting events, and to implement control interfaces for use by the management system. In this section, we examine different instrumentation techniques and present some of the standard approaches used to instrument distributed applications to provide correlated end-to-end view of transaction response time.

3.1 Manageability Trends

Current enterprise and e-business management systems are tightly coupled with the managed application and are usually constrained by the enterprise and management domain boundaries (e.g., firewalls). For a management system to provide an end-to-end view of the service state, a federated solution spanning multiple enterprises and communicating across management domains, is needed. Figure 5 shows the different approaches to integrating a management system with the managed service.
The first integration approach usually employs a proprietary set of interfaces and instrumentation points to collect the required measurements and apply the required commands. It also assumes full access to the required information and full control over the service.
In the second and third options, the managed service exposes a set of well-defined interfaces that can be exploited by the management system in order to perform its management tasks. The composition could be governed by a pre-negotiated (option b) or just-in-time-negotiated (option c) contract. In the second and third options, the management system could itself be implemented as an outsourced service.
The interfaces, which tie the management system together with the managed e-service, exchange information using Internet protocols and standards wherever applicable. This facilitates implementing the management system to collect measurements from the various cross-enterprise sources and also to control applications and services that are provided by external suppliers

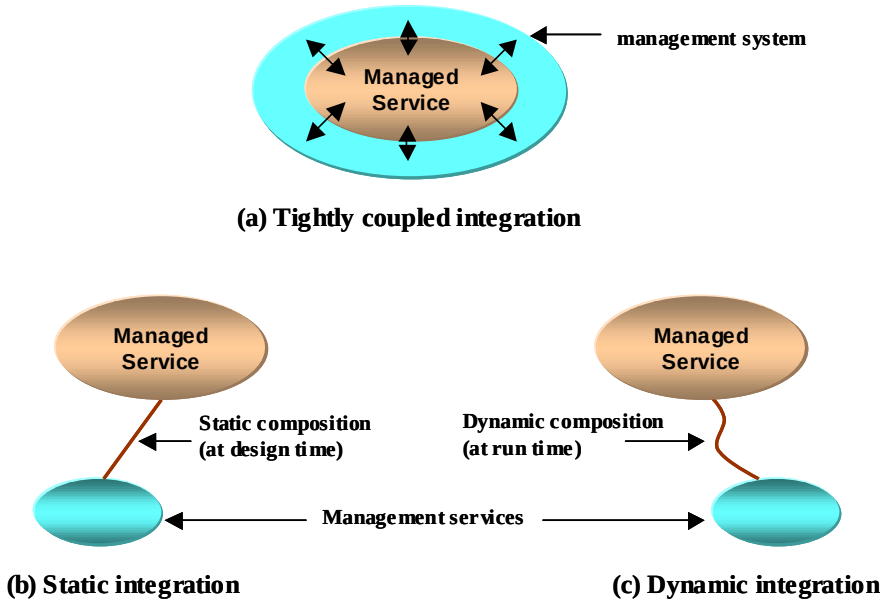


Figure 5: Instrumentation approaches

.

3.2 Instrumentation Standards

There are several instrumentation standards such as ARM [7], SNMP [10][11], and CIM [12], that are used by network, system, and application developers to expose models, events, measurements, and control interfaces to the corresponding management systems. ARM (Application Response Measurement) is an API that defines function calls used to instrument an application for transaction monitoring. An ARM agent collects these calls and correlates them to construct the overall response time of composite transactions spanning over multiple services, as shown in Figure 6.
SNMP (Simple Network Management Protocol) is a standard protocol used to exchange network and some system information in a simple tabular representation called MIB (Management Information Base). Each application can define its own MIB. SNMP gained wide popularity because of its simplicity. However, this simplicity prohibited defining more complex data and expressing relationships between data elements. CIM (Common Information Model) came to the rescue, with rich syntax for representing management information and relationships between managed objects.
In our designs, we used CIM to model the managed services and its workflow, and to represent transactions and metrics. We also used ARM-like mechanism to instrument the services and collect transaction measurements from all involved services, as described in Section 5.

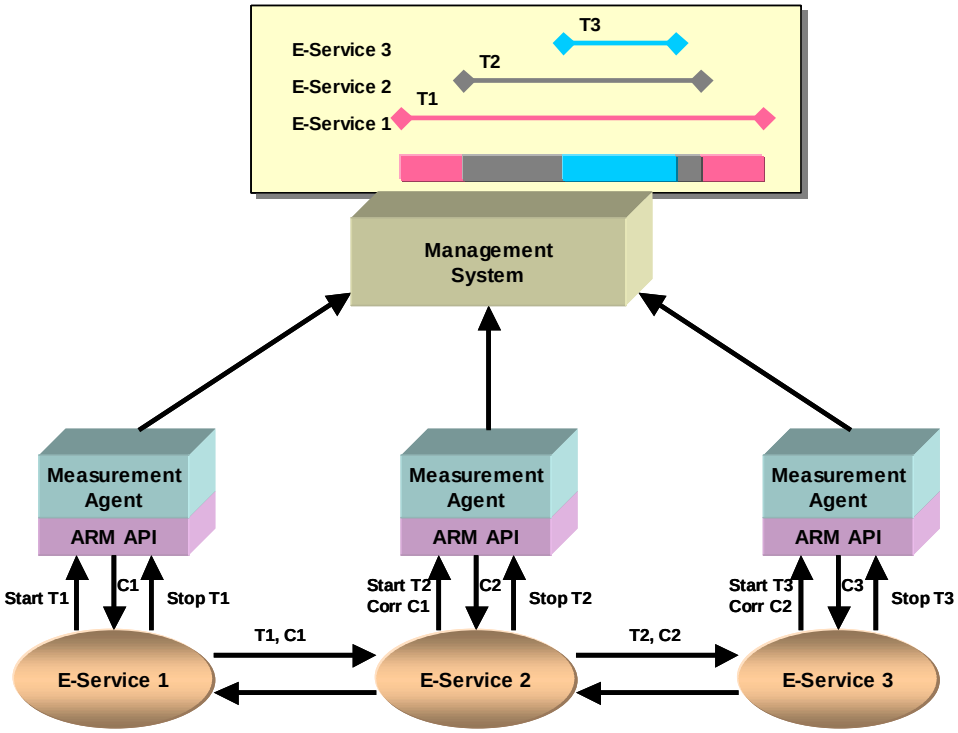


Figure 6: Using ARM to instrument services

3.3 Instrumentation Options

Instrumentation techniques fall under two main categories: invasive and non-invasive. Invasive techniques require adding instrumentation code within the applications that make up the service. This is usually done at design time and provides the highest level of control over the amount and kind of information provided to the management system. Non-invasive techniques are external components that are bolted in after the service is implemented, which allows instrumenting legacy applications and services. ARMing the application is an example of invasive instrumentation, while adding components that intercept transactions, extract the required information, and send it to the management system is an example of non-invasive instrumentation (see Figure 7).
In the invasive case, the instrumentation code could be added by service developers at design time. However, programmers tend to ignore adding these calls, as long as it is not enforced or clearly stated in the design requirements. Another way to add instrumentation is by providing development environments and tools that would automatically insert the required calls into the code. This approach may require initial setup and configuration to select the right insertion points.

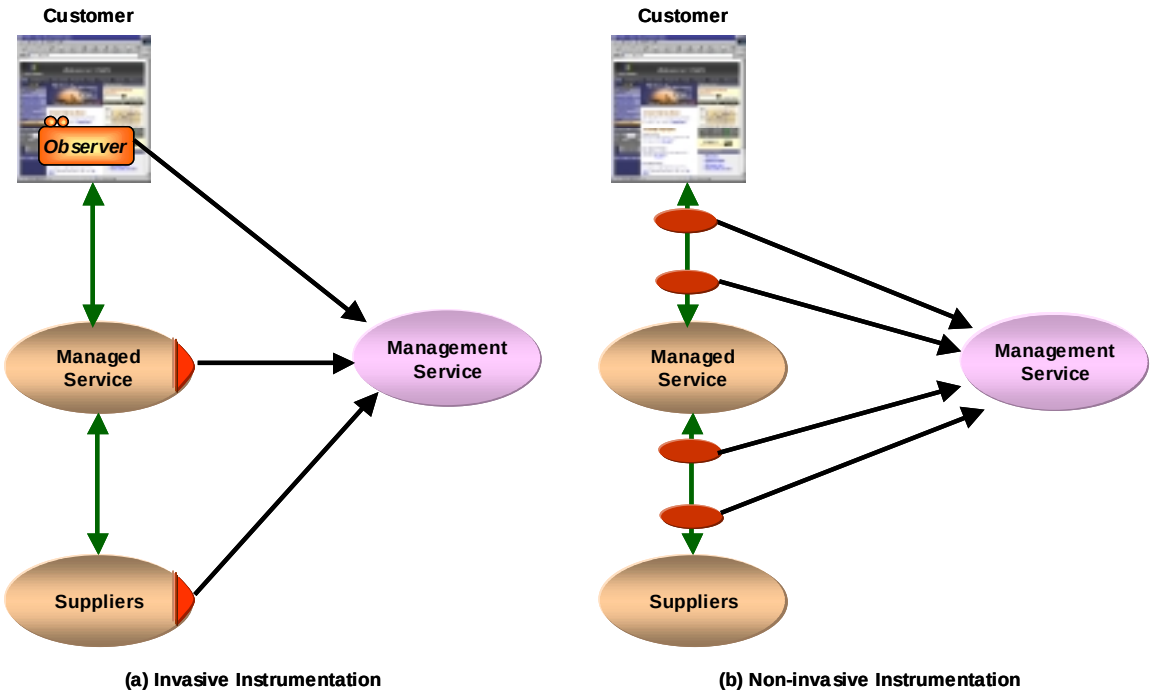


Figure 7: Invasive and non-invasive instrumentation

4. XML Management Vocabulary and Protocols

The managed service has to send instrumentation messages describing the models and events to the management system. Similarly, the management system has to send control messages back to the managed service. There are two basic models of communication for such exchange of messages to occur:

1. **Network Object Model (NOM):** where the communicating parties define and agree on certain interfaces and call methods on each other.

2. **Document Model (DM):** where both parties agree on certain message formats and send documents to each other.

The NOM approach requires both the management system and managed service to define instrumentation related interfaces and their methods at construction time, which makes them strongly bound to each other. On the other hand, the DM approach allows the management system and the managed service to be loosely bound to each other and leaves more room for evolution of messages over time [13]. In this work, we adopted the Document Model and defined certain message schemas to facilitate the exchange of instrumentation messages that are understood by both the parties. This work is based on standard agent communication languages such as KQML [14], ACL [15], Speak-easy (from HP), and FLBC [16].

4.1 E-Management Vocabulary

As part of this research effort, we defined the E-Management Vocabulary (EMV), which is a collection of terms and documents that are commonly used in the context of enterprise and cross-enterprise service management. In particular, EMV consists of:

- Basic management terms or building blocks, which form the basis for defining more complex management-related documents.
- XML documents that are exchanged between a business and its management service provider.
- Management interaction protocols for exchanging the above XML documents.

A model-based approach is used to define the required terms and documents. Models were designed to represent the structure of the service and its configuration parameters, transactions supported by the service, workflow and state transition, and the metrics generated by the service. We used UML (Unified Modeling Language) to define these models. From these models, we can use XMI to generate the required XML schemas, or XMLcim (a DMTF standard) to generate CIM-compliant XML documents that could be exchanged with other CIM agents. Figure 8 illustrates this approach.

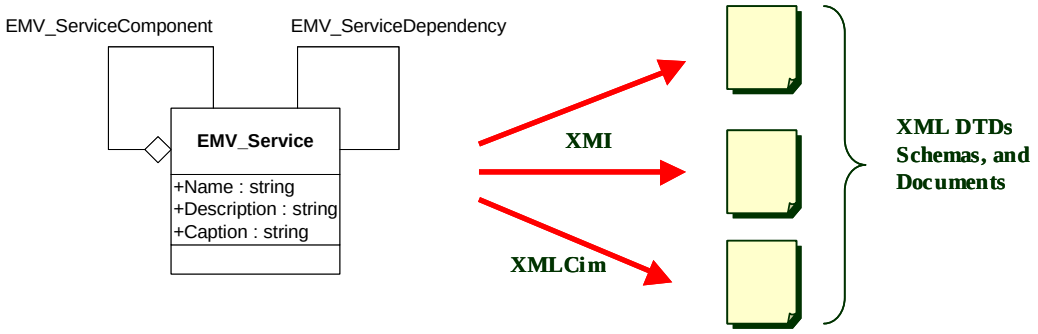


Figure 8: Model-based design of the management vocabulary

4.2 Management Information

The management information provided by the managed service include:

- **Models:** A set of models representing the structure and configuration of the service, state transitions within the service, and workflow of the supported transactions.

- **Measurements:** Real-time measurements whenever transactions are started and stopped at service and sub-service boundaries, availability heartbeats, contract and offer details, lifecycle state changes, etc.
- **Events:** Asynchronous messages sent whenever errors or important events occur within the service.

- **Configuration:** A set of parameters representing current service and environment configuration such as maximum number of users, authentication requirements, machine type, etc.

- **Control points:** Interfaces to change the lifecycle state of the service (e.g., start, stop, suspend, etc.), to improve the availability (e.g., load-balancing, replication, migration, etc.), or to fix certain problems (e.g., rollback, re-index, restart, etc.).

The following example shows the DTD and a corresponding XML document for defining a service:

Listing 1: EMV_Service DTD

```
<!ELEMENT EMV_Service.Name (#PCDATA)>

<!ELEMENT EMV_Service.Description (#PCDATA)>

<!ELEMENT EMV_Service.Caption (#PCDATA)>

<!ELEMENT EMV_Service.ServiceType (XMI.reference)>

<!ELEMENT EMV_Service.ServiceProvider (XMI.reference)>

<!ELEMENT EMV_Service (
          EMV_Service.Name?,
          EMV_Service.Description?,
          EMV_Service.Caption?,
          EMV_Service.ServiceType?,
          EMV_Service.ServiceProvider?,
          XMI.extension*)?>

<!ATTLIST EMV_Service
          %XMI.element.att;
          %XMI.link.att;>
```

Listing 2: Example for EMV_Service

```
<EMV_Service xmi.id = “NetCash PaymentAuthorization”>
   <EMV_Service.Name>
      NetCash PaymentAuthorization
   </EMV_Service.Name>
   <EMV_Service.Description>
      The PaymentAuthorization service of NetCash accepts
      requests for payment authorizations.
   </EMV_Service.Description>
   <EMV_Service.Caption>
      NetCash’s Authorization services for payments
   </EMV_Service.Caption>
```

```
   <EMV_Service.ServiceType>
      <XMI.reference idref = “PaymentAuthorization”/>
   </EMV_Service.ServiceType>
   <EMV_Service.ServiceProvider>
      <XMI.reference idref = “NetCash”/>
   </EMV_Service.ServiceProvider>
</EMV_Service>
```

5. Prototype: Building a Remote Monitoring Service

To demonstrate the feasibility of the proposed approach, we built a web-based remote monitoring service prototype, which provides aggregate views of the end-to-end transaction response time for other services. It utilizes the XML management vocabulary and communication protocols, described in Section 4, and provides web-based views through a management portal, which include, among other things, different remote management and information services. Figure 9 shows an overview of the managed service, which is called the “Accidental Tourist” service.

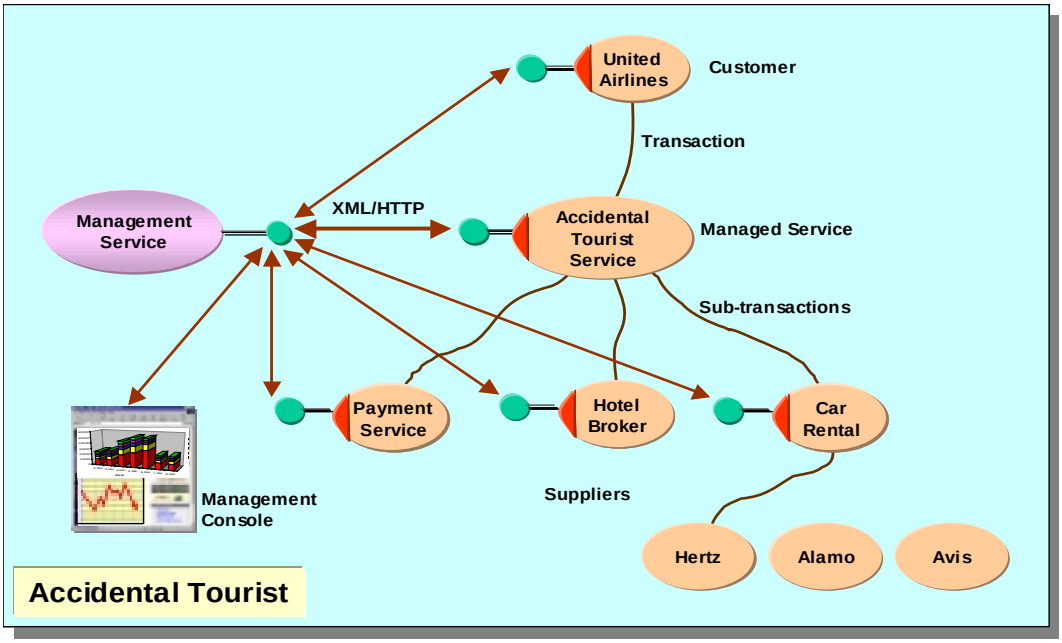


Figure 9: Remote management prototype

Airline companies (e.g., United Airlines) can use the accidental tourist service whenever one of their flights is cancelled to secure accommodation and transportation to its passengers if they have to stay overnight for the next available flight. The accidental tourist service uses several external services such as a hotel broker service, car rental services (e.g., Hertz, Avis, etc.), and a payment service (e.g., Veriphone).
The scenario starts by the managed service sending its service model to the management service followed by a configuration document specifying the required management features. The management service sets the required configuration and starts collecting low-level measurements and events in the form of XML documents from the managed

service. The management console, which provides web-based customizable views to the managed service, updates its views by requesting the necessary information from the management service. The management service may also receive low-level measurements from some of the managed services' suppliers and consumers, based on their contractual agreements. This will provide the end-to-end view of the state of the overall service. Figure 10 shows a screen shot of the management console, displaying various response time metrics of the accidental tourist service broken down by car rental services, hotel broker services, and by the payment service. As a further level of drilldown, one can see the average response times for each of the car rental services - Hertz, Alamo, and Avis.

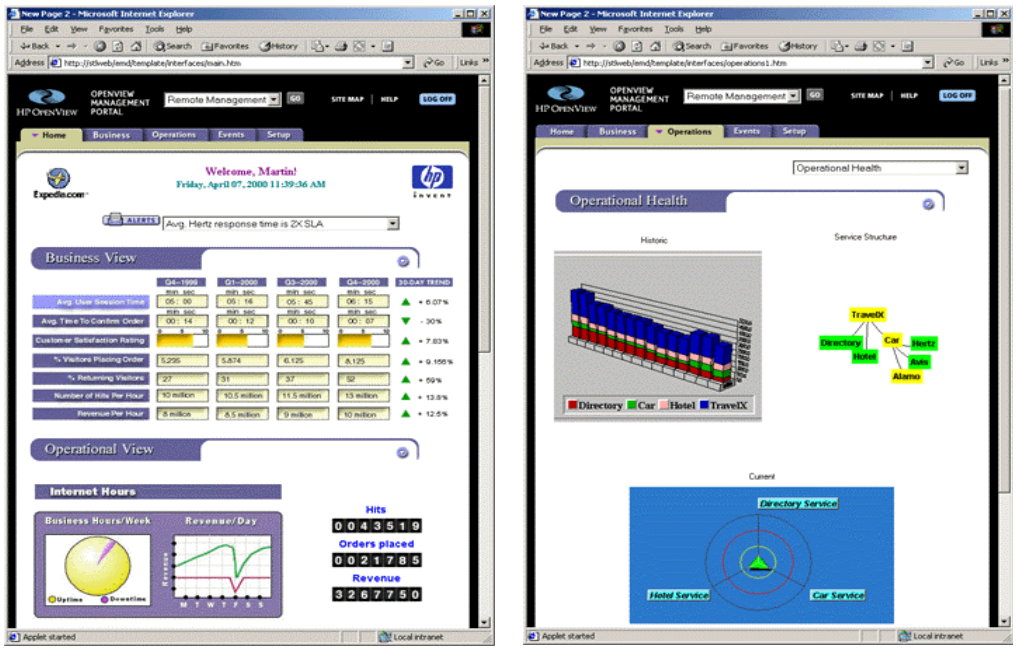


Figure 10: Web-based management console

6. Conclusion

In this paper, we described an approach to providing web-based remote management services and discussed the requirements such services. We presented an XML management vocabulary and communication protocol used to exchange management information and control commands between the managed service and the management system. Several instrumentation technologies were discussed and several options were presented to provide the required instrumentation. Finally, we described a management system prototype, which provides remote monitoring capabilities.
We believe that the management outsourcing market will continue to grow and that remote management services provide a flexible, and dynamic approach to outsourced management capabilities for internet-based services and applications.

References

[1] M. Subramanian, “Network Management : Principles and Practice.” Addison Wesley Longman, Inc. 1999.

[2] S. M. Ballew and M. Loukides, “Managing IP Networks With Cisco Routers.” O'Reilly & Associates, Inc. 1998.

[3] J.P. Martin-Flatin, S. Znaty, and J.P. Hubaux, “A Survey of Distributed Enterprise Network and Systems Management Paradigms.” Journal of Network and Systems Management, Special Issue on Enterprise Network and Systems Management. Vol. 7, No. 1, March 1999.

[4] R. Sturm and W. Bumpus, “Foundations of Application Management.” Wiley, John & Sons, Inc. 1998.

[5] D. Spuler, “Enterprise Application Management with PATROL.” Prentice Hall, Inc. 1999.

[6] H. Raghava Rao, K. Nam, and A. Chaudhury. Information systems outsourcing. Communications of the ACM, Vol.39, No.7, July 1996, pp.27-28.

[7] Outsourcing Academics (http://www.OutsourcingAcademics.com).

[8] M. J. Katchabaw, H. L. Lutfiyya and M. A. Bauer “A Quality of Service Management Testbed.” Journal of Network and Systems Management, Special Issue on Enterprise Network and Systems Management. Vol. 7, No. 1, March 1999.

[9] Application Response Measurement (ARM). http://www.openview.hp.com/solutions/gen-eral/armguide.html.

[10] W. Stallings, “SNMP, SNMPv2, SNMPv3, and RMON 1 and 2.” Addison Wesley Longman, Inc. 1998.

[11] D. Zeltserman, “A Practical Guide to SNMPv3 and Network Management.” Prentice Hall. 1999.

[12] W. Bumpus, J. W. Sweitzer, P. Thompson, A. R. Westerinen, and R. C. Williams, “Common Information Model: Implementing the Object Model for Enterprise Management.” John Wiley & Sons Inc. 1999.

[13] CommerceNet and the ECO framework. http://www.commerce.net

[14] T. Finin, R. Fritzson, D. McKay, and R. McEntire. KQML as an agent communication language. Proceedings of the Third International Conference on Information and Knowledge Management (CIKM’94), ACM Press, November, 1994

[15] P.D. O’Brien and R. Nicol, FIPA: Towards a standard for intelligent agents. BT Technical Journal, Vol. 16, 1998.

[16] S. Moore. KQML and FLBC: Contrasting agent communication languages. Proceedings of the 32nd Hawaii International Conference on System Sciences, Maui, Hawaii, 1998.