# E-services Management Requirements

Vijay Machiraju, Mohamed Dekhil,
Martin Griss, Klaus Wurster
Software Technology Laboratory
HP Laboratories Palo Alto
HPL-2000-60
May, 2000

E-mail: dekhil@hpl.hp.com

e-services management, web-based management, end-to-end management

E-services are the realization of truly federated and highly dynamic e-business components, which communicate across enterprise boundaries to provide high quality and better value services to end-users and businesses over the Internet. The automation, dynamism, and federation characteristics of e-services add additional challenges to the existing problems of traditional enterprise and e-business management. We are interested in developing a comprehensive architecture for e-service management and we believe that this requires a clearly defined interplay between the managed e-service and the management system.

In this paper, we discuss the requirements for effective end-to-end and top-to-bottom management of e-services and introduce the notion of *management access protocol* (MAP), which provides a uniform mechanism to access management information and control hooks on the managed object. The MAP exposes certain interfaces that allow a management system to communicate with the managed service. We then describe a prototype demonstrating the use of XML-based messaging scheme to communicate and exchange management and control information.

# 1 Introduction

E-services are the next generation of modular internet-based applications that communicate with one another over the Internet [1]. There are three important benefits that e-services bring to enterprises more automation, more federation, and more dynamism. By allowing applications to seamlessly interact with other applications within and outside enterprise boundaries, e-services reduce the amount of human interaction needed to exchange or transfer information, thus enabling an extra degree of automation. For example, intelligent software agents wrapped as e-services can be used to automate the negotiation of services between businesses or in triggering the use of other e-services [2]. Secondly, as service providers focus on creating and offering value-added services, they are more inclined to outsource non-strategic parts of their businesses to external service providers [3]. Enabling the development of new services by composing existing services is at the heart of e-services. This allows complex business services to be federated into smaller sub-services across several enterprises. Finally, the extra automation and federation make e-services more dynamic. Instead of statically binding a set of services to each other, one can dynamically discover new e-services with the right set of features (through broker agents, for example) and bind to them at run-time [4].

There are two key players in the e-services market - the service providers, who create and offer new e-services, and the service consumers, who utilize and consume existing e-services (possibly to create other new ones). Service providers create new e-services by using the services offered by other providers (suppliers) and by building their own custom applications. For example, a provider could offer a book selling e-service by outsourcing shipping, payment, and billing to other service providers (see Figure 1). In addition, the book selling e-service provider may need a set of complex distributed applications such as web servers, databases, and data mining agents to keep track of customers and their interests.

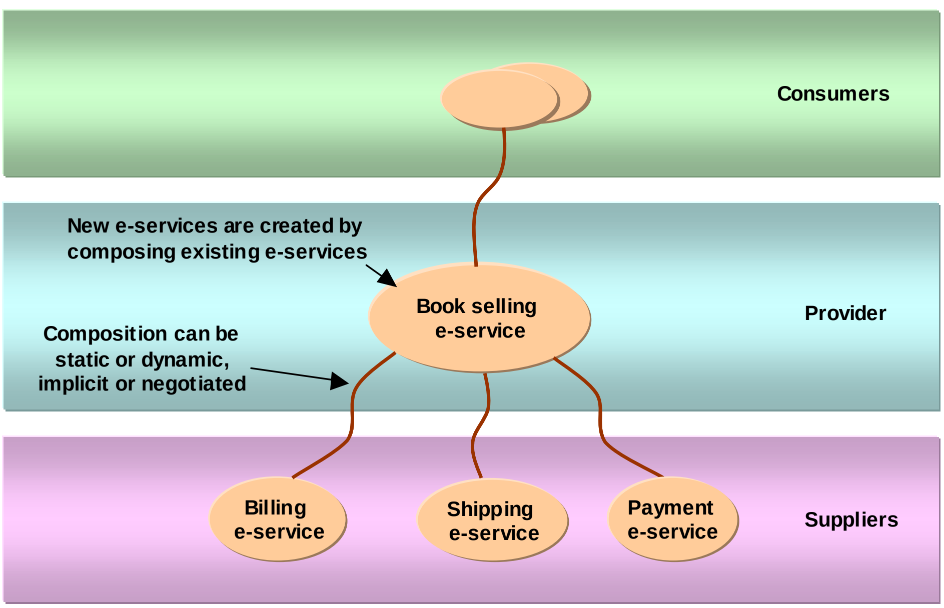


**Figure 1: E-service composition**

The problem of e-service management is to maintain the e-service in a desired state by monitoring and controlling its operations and use of resources. Poor performance, lack of availability, crossing certain thresholds, and contract violation are some of the ways in which e-services can deviate from their desired state of operation. Such deviations could occur for various reasons including errors in application or service logic, failures in networks and systems that host these services, improper configuration parameters, and unauthorized intrusion. An e-service management system either detects and corrects problems in real-time, or it observes the trends in order to predict and rectify the situation before problems occur. E-services management systems can also perform historical analysis of the services they manage and help in planning activities. In addition, they can be integrated into more powerful business-management systems to assess the impact of the services on the overall business.

In this paper, we start by discussing the requirements for effective end-to-end and top-to-bottom management of e-services. These requirements are derived from existing management standards and products and from our own experience in creating e-business management solutions. We then present a mechanism for uniform access to management information and controls at different levels of the managed object using a well-defined management access protocol (MAP). Finally, we describe a prototype that illustrates the use of XML-based MAP to remotely manage e-services and to provide different views on its behavior.

# 2 Management Requirements

A well-managed e-service exposes sufficient information, measurements, and control points for use by the management system. A good e-service management system uses this information effectively to monitor the state of the service, to analyze and diagnose problems that occur within the service, to control its behavior, and to expose some or all of the metrics and controls points for use by other services, management systems, and graphical interface tools. In this section, we describe the e-services management space and explore the requirements on the e-services management system to cover that space. For more details, refer to the e-service management requirements document [5].

## 2.1 The E-services Management Space

To effectively monitor and control the state of an e-service, certain requirements on the managed e-service and on the management system needs to be satisfied. These requirements are derived from a management space defined as the cross product of management tasks (e.g., monitoring, analysis, control, etc.), metrics (e.g., performance, availability, contracts, etc.), and time (current, historic, and future estimates). This space can be applied end-to-end or top-down to provide the required view (see Figure 2).

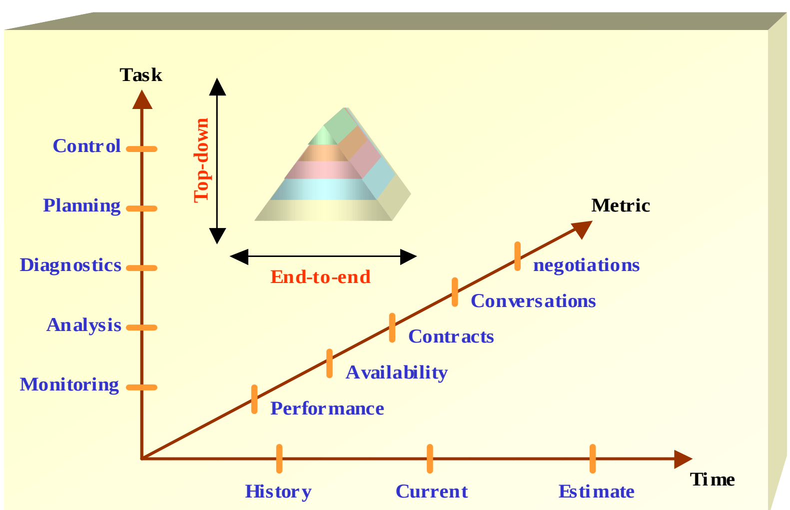


**Figure 2: The e-services management space**

E-services introduce new metrics such as contracts, negotiations, and conversations. For example, contract metrics include number of active contracts, average number of completed contracts per day, number of violated contracts, etc. The problem of end-to-end management is to manage the entire interaction starting from the consumers, executing through the provider's applications, and flowing into external suppliers. Top-to-bottom management involves managing the infrastructure elements such as networks, systems, and applications to meet the overall service and business goals.

## 2.2 E-services Management Requirements

In this section, we examine a list of requirements that should be satisfied by e-service management systems. This list has been compiled from various existing management products and standards [6][7], from other related research activities [8][9][10], and from our own experience [11].

### 2.2.1 Basic Tasks

Management is broadly defined as monitoring and controlling the managed system. A management system helps in:

- collecting information and measurements from all possible sources,
- aggregating these measurements and information into high level metrics,
- discovering any deviations of the monitored metrics from desirable levels,
- understanding or diagnosing the reasons for deviations,
- fixing the associated problems by controlling the managed service, and sometimes in
- predicting and proactively managing the service.

To facilitate these tasks, a management system could provide an interface for accepting low-level information and measurements from the managed e-service. In addition, it could also provide the following interfaces for other high-level management systems and graphical interface tools:

- an interface for exposing some or all of the collected measurements and computed metrics in the form of service state (e.g., health and availability),
- an interface for accepting input for desirable states in the form of service levels, ranges, base-lines, or policies, and
- an interface for exposing high-level control points to control the managed e-service.

### 2.2.2 Levels of Abstractions

An e-service management system converts the low-level data and low-level control points of the managed e-service into high-level management information and control points. In general, a management system adds value by consolidating the data collected from multiple sources, aggregating the collected data, and increasing the level of abstraction of the collected data or exposed interfaces. The data that is collected from the managed e-service may include:

- *Measurements*: The term “measurements” is often used in reference to the low-level numeric data that is provided by the managed e-service. Examples of measurements include start and stop times for transactions and CPU utilization of a process.
- *Models*: An application’s or e-service’s local perspective of its topology or structure (topology model), transaction workflow (workflow model), or state transitions (state transition model).
- *Events*: The term event is used to describe non-periodic state changes (anticipated or unanticipated) in the e-service. An example of an anticipated event is the change in the state of an e-service from “started” to “initialized.” An example of an unanticipated or undesirable event is an error in the execution of a transaction.

Low-level control interfaces that are supported by an e-service may include:

- *Lifecycle control*: An interface to start, stop, suspend, or resume an e-service.

- *Configuration control*: An interface to change the configuration parameters of an e-service. Each e-service may define its own configuration parameters. An example of a configuration parameter that can be controlled is the buffer size in an application.

An e-service management uses the above low-level data and control points and raises the level of abstraction from:

- measurements to metrics,
- local models to consolidated global (end-to-end and top-to-bottom) models,
- events to actions, and
- low-level control interfaces to high-level commands.

### 2.2.3 The temporal dimension

All the information collected or computed by the management system could be exposed along three temporal dimensions:

- *Past*: Metrics aggregated over a period of time can be reported as historical data. Such data can be used for publishing reports, historical analysis, and planning.
- *Present*: Current or real-time values of various metrics can be computed as they are measured. These metrics can be used for real-time analysis, diagnosis, and control.
- *Future*: Future values of metrics can be estimated by various learning and prediction techniques. These estimated values can be used for pro-active management.

For example, a management system may calculate the average response time of a transaction type over the last 3 months (past), could provide real-time value of the current connected users to a particular service (present), and may calculate the estimated number of transaction per hour based on current trends (future).

### 2.2.4 End-to-end Management

Every transaction, initiated by a service consumer, follows a particular set of steps (a workflow) before being fulfilled. In particular, it starts in the consumer's applications, flows through the Internet (internet service provided by an ISP), executes through the business applications of the provider, and may even flow into the applications of the external suppliers. In each of these stages, it can also flow through e-service middleware that provides mechanisms for secure cross-enterprise communication. An end-to-end management system should tap into all the possible sources of information through this workflow in order to monitor, diagnose, and fix problems. In particular, there are three types of sources from which transactional and other measurements or metrics could be taken (see Figure 3):

1. *Consumers*: This is where most of the transactions are initiated. Metrics collected here would help in understanding the consumer's perspective of the e-service quality.
2. *E-service provider*: The business logic in the provider's e-service is implemented using complex applications that are distributed across provider's intranet. Examples of these applications include web servers, databases, and other custom application components. Measurements collected from each of these service elements can be put together to understand service bottlenecks in provider's environment.
3. *External suppliers*: Since some of the facets of an e-service are outsourced to other providers, it is essential to collect some information from those e-services as well. However, these suppliers may expose only limited information regarding the management of their e-services. As external manageability becomes one of the criteria that differentiates one e-service from another, more and more suppliers will expose portions of their internal state for manageability by consumers. Ability to track shipping transactions through FedEx and UPS is an example of external manageability.

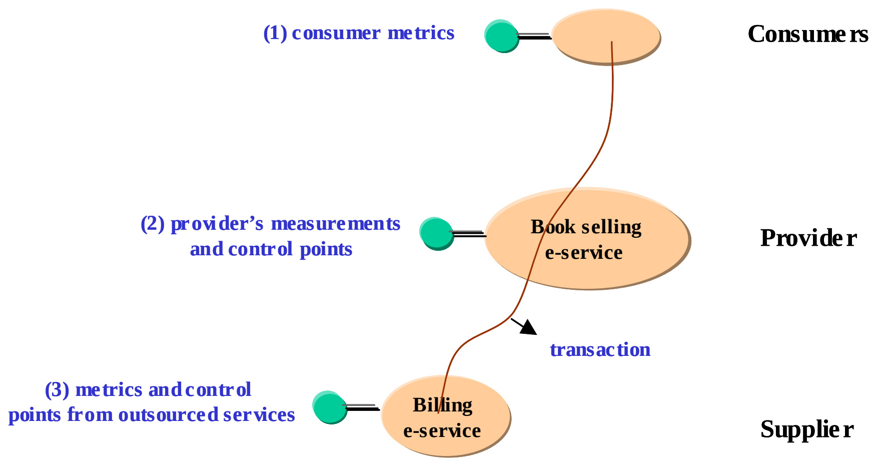


**Figure 3: End-to-end view of e-service transactions**

Metrics collected from the participants of an e-services transaction can be used to complete the end-to-end picture of that transaction's workflow.

### 2.2.5 Top-to-bottom Management

E-services are built using agent technologies, traditional applications, and middleware components, which interact with each other to offer a service to the consumers. These applications are in turn hosted on systems that are distributed across the network. This hierarchy is often referred to as the service stack and is shown in Figure 4. At the top-most layer in the service stack, we have the overall business of the provider, which may consist of one or more e-services.

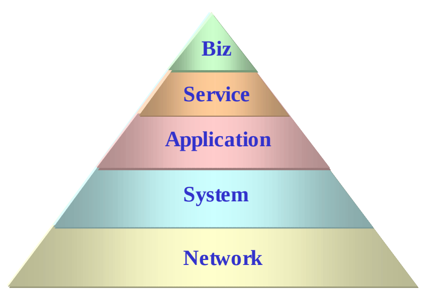


**Figure 4: Top-to-bottom view (service stack)**

In order to manage an e-service, one should manage elements that influence the e-service at all layers in the service stack. There are two approaches to achieving this: In the top-down approach, a set of business goals and objectives drive the requirements on what kind of management should occur at the service layer. Similarly, the service level objectives drive the application level objectives and so on. In this approach, only those measurements that are necessary to accomplish a task or verify an objective at a higher level are collected from the lower levels. The second approach is the bottom-up approach where a network management system collects all the measurements necessary to manage the network irrespective of the system, application, and service goals. Similarly, an application management system selects a set of network and system metrics that it needs and manages the application layer without consideration to the service level objectives. In other words, a management system at any level uses the information from the lower levels, but does not depend on higher levels.

Irrespective of the approach, an e-service management system should be able to perform the following two tasks related to top-to-down management:

- *Roll-up*: Events, measurements, and metrics collected at lower levels should be rolled-up to higher levels in the service stack. For example, metrics related to network performance should be rolled-up into system, application, and the overall service performance.
- *Drill-down*: An e-service management system should facilitate drill-down across the layers of the service stack in order to identify the root cause of a problem. For example, when a service fails due to a network failure, an e-service management system should be able to drill-down through the application, system, and network layers to identify the exact cause of the service failure.

## 2.3 Requirements on the Managed E-service

In order for the management system to fulfill all these requirements, the managed e-service should provide all the necessary information. This includes:

- *Models*: A set of models representing the structure and configuration of the e-service, state transitions within the e-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*: Events whenever errors occur within the e-service.
- *Control points*: Interfaces to change the lifecycle state of the e-service (start, stop, suspend the e-service), to increase the availability (load-balancing, replicating, moving), etc.

To access this information, standard interfaces need to be defined and protocols for communicating requests and replies between the management system and the managed e-service seed to be developed. In the next section, we talk about such protocol and the need for a standard vocabulary to formulate management messages between the management systems and the managed e-services.

# 3 Management Access Protocol

Current enterprise and e-business management systems are tightly coupled with the managed application and 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, which spans over multiple enterprises and communicates across management domains, is needed. Figure 5 shows the different approaches to integrating a management system with the managed e-service.

The first integration approach usually employs proprietary set of interfaces and instrumentation points to collects the required measurements and applies required controls. In the second and third options, the managed e-service exposes a set of well-defined interfaces that can be exploited by the management system in order to perform its management tasks. In the last option, the management system (or a subset of the functionality thereof) is dynamically composed as an outsourced facet into the e-service. The composition could be governed by a pre-negotiated or just-in-time-negotiated contract. In the second and third options, the management system could itself be implemented as an e-service.

The interfaces, which tie the management system and 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.

An e-service could be built out of software agents [12] that execute conversations with other e-services in order to negotiate for a contract, or in order to trigger their execution. Many state-of-the-art intelligent agent technologies such as rule-based inference, fuzzy logic, neural nets, and genetic algorithms are being adopted to automate tasks that are traditionally handled by humans [13]. Such automation is also driven by the recent evolution of business communication languages and ontologies [14][15][16].

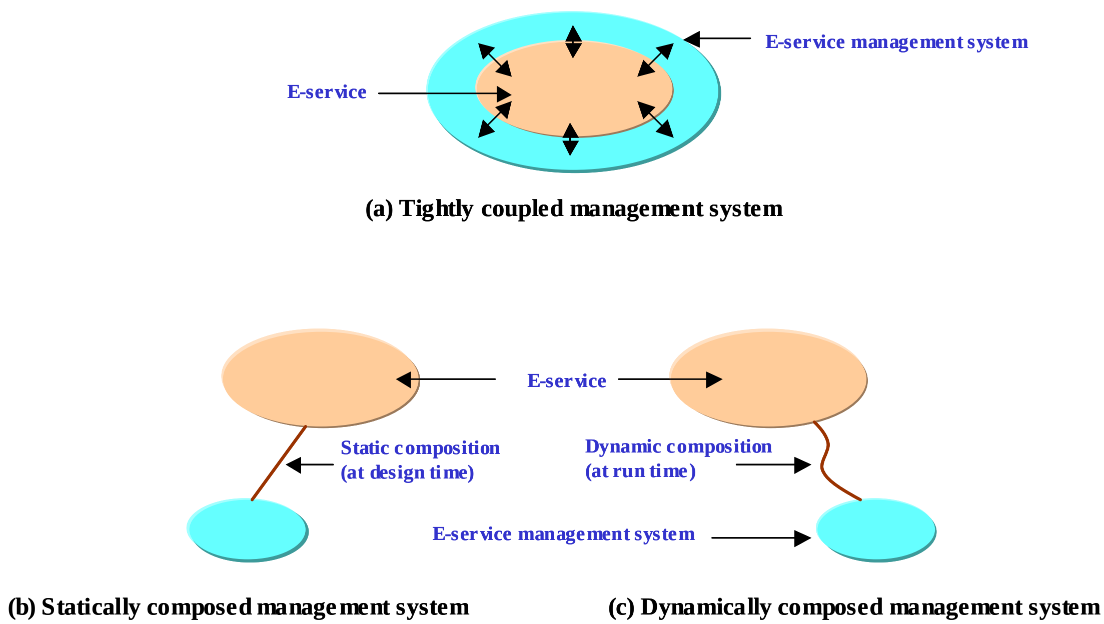


**Figure 5: Integrating management solutions with the managed systems**

As with other types of communications between e-services, exchanging management requests and information requires defining standard messages that contain pre-defined set of documents (e.g., XML documents) and defining protocols for exchanging these messages. We call such protocols *management access protocols* (MAPs). Figure 6 shows some of the documents exchanged between a management system and the managed e-service (the provider), along with the other parties involved in carrying out the transactions related to its services (consumers and suppliers) to get the end-to-end view of the service state.

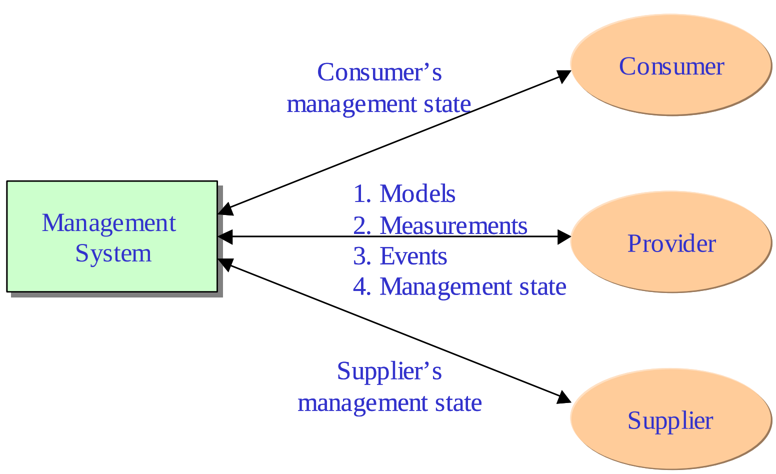


**Figure 6: Documents exchanged as part of a MAP for e-services management**

The documents that are exchanged between e-services contain the required information formulated using certain vocabulary. We have developed an initial draft for such protocols and vocabulary and produced a conceptual and user's guide describing the terms used and their meaning for e-services management [17]. This vocabulary is based on standard agent communication languages such as KQML [18], ACL [19], Speak-easy (from HP), or FLBC [20]. Messages built out of vocabularies from these languages constitute the basic unit of agent communication.

Using the same MAP to access management information and send control commands at the different levels of the service stack, shown in Figure 4, would provide a uniform mechanism to end-to-end and top-to-bottom management view. In other words, we can view every component in the service stack as an e-service and treat it the same way. For example, we can view an SQL application as an e-service, which supports the standard MAP for getting its metrics and setting its parameters. Such uniform view allows us to reuse most of the communication and access components of the management system.

# 4 Prototype: Building a Management E-service

To illustrate some of the concepts presented in the paper and to demonstrate the usability of the Management Access Protocol described earlier, we designed a management e-service, which employs an XML-based messaging scheme and uses the management vocabulary to manage another e-service, called the accidental tourist e-service. Airline companies (e.g., United Airlines) can use the accidental tourist e-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 e-service, shown in Figure 7, uses several external e-services such as a hotel broker e-service, car rental e-services (e.g., Hertz, Avis, etc.), and a payment e-service (e.g., Veriphone).

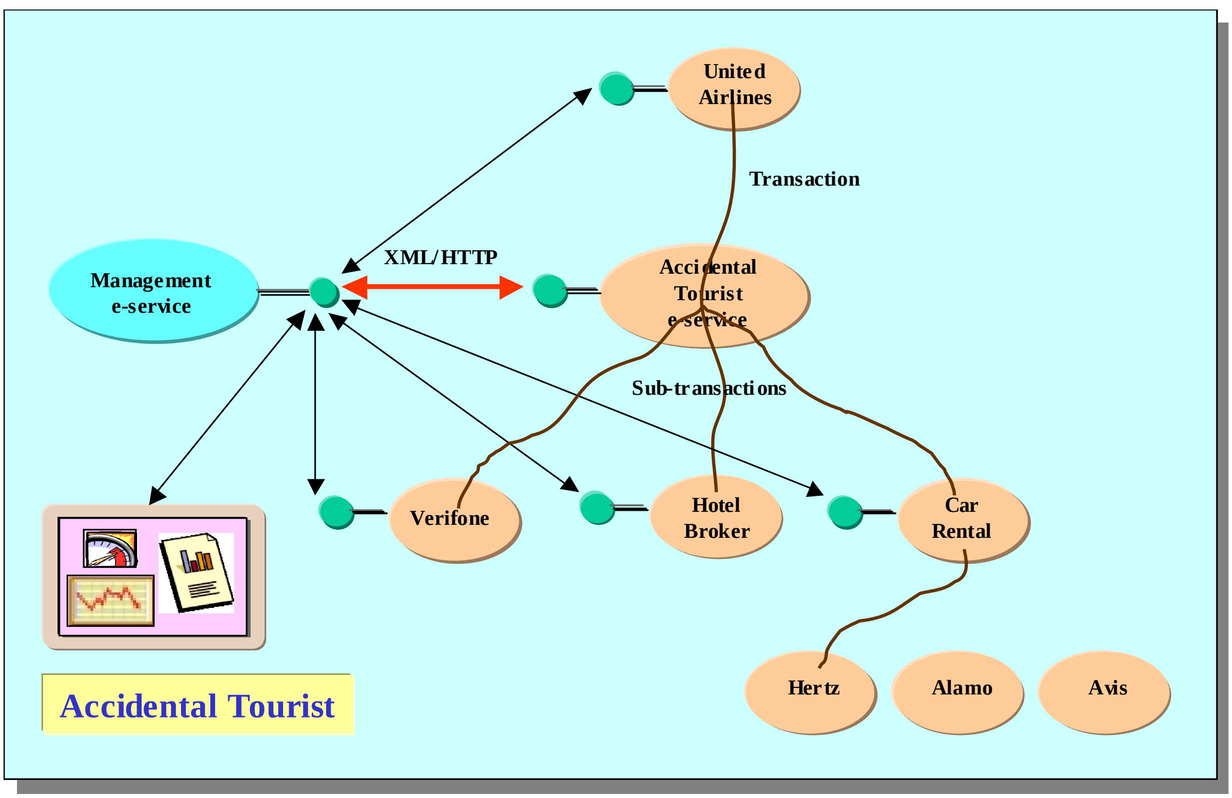


**Figure 7: The accidental tourist e-service**

Assuming that the negotiation and contracting phase between the accidental tourist (the managed) e-service and the management e-service has already completed, the management cycle starts by the managed e-service sending its service model to the management e-service followed by a configuration document specifying the required management features it needs. The management e-service sets the required configuration and starts getting low-level measurements and events through measurements documents from the managed e-service. The management console, which provides web-based customizable views to the managed e-service, updates its views by requesting the necessary information from the management e-service. The management e-service may also receive low-level measurements from some of the managed e-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 8 shows a screen shot of the management console, displaying various average response time metrics of the accidental tourist e-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.

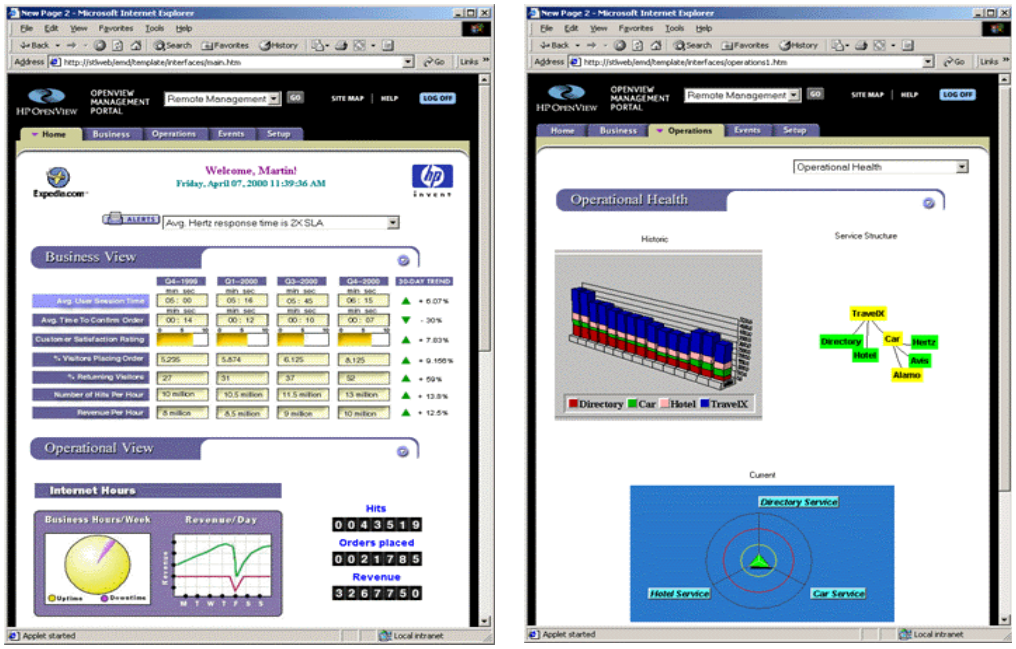


**Figure 8: Web-based management console**

# 5 Conclusion

E-services introduce extra automation, federation, and dynamism, which pose new challenges to the e-services management systems. In this paper, we presented some of the requirements on the management systems and on the managed e-service to generate well-managed services and powerful management systems to manage these e-services. These requirements were derived from various existing management products and standards and from our own experience in building enterprise management solutions.

We then introduced the notion of management access protocol (MAP), which defines the communication protocols between e-services when exchanging management messages. We also defined a management vocabulary to capture the syntax and semantics of different phrases and terms used within a management document. Finally, we presented a management e-service prototype, which utilizes the concepts discussed in this paper to manage the end-to-end aspects of an e-service.

We are in the process of completing the management vocabulary and finalizing the definition of the management access protocol including the different management documents and its flow between e-services. Next, we would like to tie e-services management with enterprise management to provide top-to-bottom management within the e-service provider's infrastructure. This would provide a complete management solution covering most of the management space discussed in this paper.

## References


[1] E-services. http://www.hp.com/e-services.

[2] P. Maes, R. H. Guttman, and A. G. Moukas. Agents that buy and sell. Communications of the ACM, Vol.42, No.3, March 1999, pp.81-91.

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

[4] D. Martin, H. Ohama, D. Moran and A. Cheyer. Information brokering in an agent architecture. Proceedings of The Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM 97), London, April 1997.

[5] V. Machiraju, M. Dekhil, M. Griss, and J. Holland, E-services Management requirements. HP Labs Tech Report. October 1999.

[6] HP OpenView. http://www.openview.hp.com.

[7] Application Response Measurement (ARM). http://www.openview.hp.com/solutions/general/armguide.html.

[8] P. Bhoj, S. Singhal, and S. Chutani. SLA management in federated environments. HP Laboratories technical report HPL-98-203, December 1998.

[9] P. Bhoj, G. Murthy, S. Ramanathan, and K. Scott. Measurement and monitoring enhancements for WebQoS. HP Laboratories technical report HPL-1999-106, September 1999.

[10] S. Frolund and J. Koistenen. Quality of service aware distributed object systems. HP Laboratories technical report HPL-98-142, August 1998.

[11] V. Machiraju, M. Dekhil, K. Wurster, P. Garg, M. Griss, and J. Holland. Towards generic application auto-discovery. HP Laboratories technical report HPL-1999-80, July 1999.

[12] M. Griss. My agent will call your agent. To be published in Software Development in January 2000.

[13] J. P. Bigus and J. Bigus. Constructing intelligent agents with Java. Wiley Computer Publishing, 1998.

[14] R. J. Glushko, J. M. Tenenbaum, and B. Meltzer. An XML framework for agent-based e-commerce. Communications of the ACM, Vol.42, No.3, March 1999, pp.106-114.

[15] eCo framework. http://eco.commerce.net.

[16] Common Business Library (CBL). http://www.commerceone.com/solutions/xml/cbl/cbl.htm.

[17] V. Machiraju, M. Dekhil, M. Griss, and J. Holland, E-Service management vocabulary: conceptual and reference Guide. HP Labs Tech Report. November 1999.

[18] 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

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

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