Requirements for Management of E-Services

Vijay Machiraju, Mohamed Dekhil, Martin Griss, Jerremy Holland
Hewlett-Packard Laboratories
1501 Page Mill Rd. MS 1U-14
Palo Alto, CA 94304
USA

Abstract

E-services are next generation applications that allow seamless construction of dynamic services over the Internet. Services and payments are exchanged between providers and consumers using automated negotiations and contracting agents. The automation, dynamism, and federation characteristics of e-services add additional challenges to the existing problems of traditional enterprise and e-business management. This document describes 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.

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 [2]. For example, intelligent software agents wrapped as e-services can now be used in automating the negotiation of services between businesses or in triggering the use of other e-services. Secondly, as service providers focus on creating and offering value-added services, they are more and more inclined to outsource non-strategic parts of their businesses to other 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 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 (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.

As increasingly more number of enterprises rely on e-services as a new channel for conducting business, management of e-services will become extremely important. The problem of **e-service management** is to maintain the e-service in a desired state by monitoring and controlling its operations and resources. Poor performance, lack of availability, crossing pre-determined 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 - errors in application or service logic, failures in networks and systems that host these services, improper configuration parameters, and unauthorized intrusion are some examples. An e-service management system can either detect and correct these problems in real-time or observe the trends, predict, and rectify the situation even before problems occur. Because of the nature of data collected and abstracted by them, e-service management systems can also perform historical analysis of the services they manage and help in planning activities. Further, they can also be integrated into more powerful business-management systems to assess the impact of the services on the overall business.

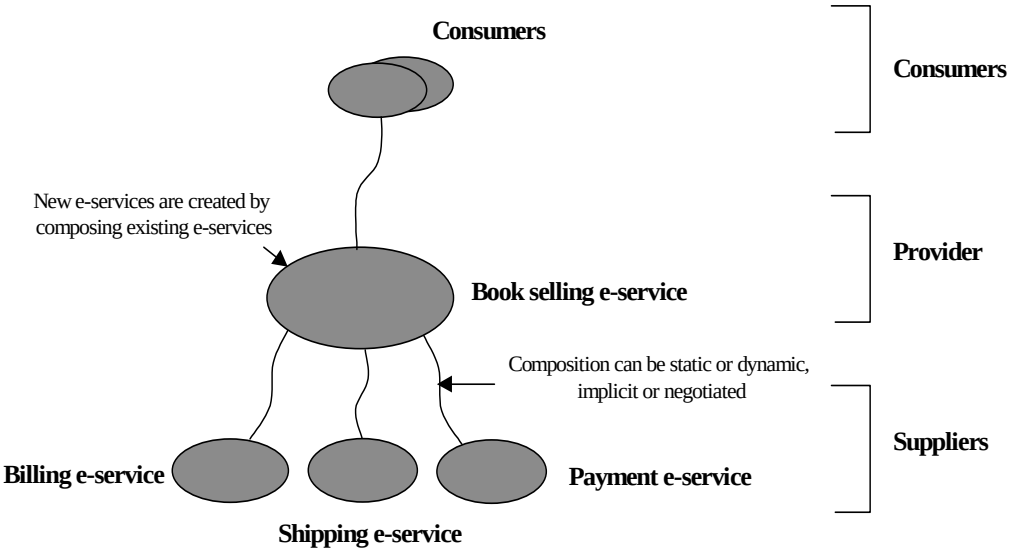


Figure 1: E-service composition

While the extra automation, federation, and dynamism make e-services powerful, they make them hard to manage at the same time. Besides, enterprises are demanding end-to-end and top-to-bottom management of their e-services, instead of managing individual pieces of their IS infrastructure. The problem of end-to-end management is to manage the entire interaction starting from the consumers, executing through the provider's applications, and perhaps flowing into external suppliers. Top-to-bottom management involves managing the infrastructure elements such as networks, systems, and applications to meet the service and business goals.

We are interested in developing a comprehensive architecture for e-service management. We believe that this requires a clearly defined interplay between the managed e-service and the management system. 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 even to expose some or all of the metrics and controls points for use by other services, management systems, and graphical interface tools. In this document, we begin by examining 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 [5, 6], from other related research activities [7, 8, 9], and from our own personal experience [10]. The rest of this document is organized as follows: Section 2 covers some more background on e-service management and presents a set of basic requirements. In section 3, we list the various metrics along which e-services could be managed. In section 4, we provide a few more requirements on e-service management systems that arise from the necessity to manage the extra automation, federation, and dynamism in e-services. We summarize with some conclusions in section 5.

2. Basic Concepts

2.1 From monitoring to control

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 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. Besides, 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**,
- an interface for accepting input for desirable states in the form of service levels, ranges, baselines, or policies, and
- an interface for exposing high-level control points to control the managed e-service.

To illustrate this with an example, consider “average response time of a transaction” as one of the metrics that should be managed. An e-service management system can

- collect low-level measurements from the e-service (start and stop times for every instance of that transaction type),
- aggregate these measurements into an average response time by computing the response time of each transaction as the difference in the start and stop times, and by averaging all the response times over a period of time,
- expose the computed average response time for use and display by external entities,
- accept as input (or automatically learn) a baseline for the desirable average response times,
- discover and notify (through an event) whenever the monitored average response time crosses the baseline,
- diagnose the problem of poor response times by correlating the computed averages with individual sub-averages for each of the sub-services,

- inform the user of the management system about the root cause of the problem,
- provide a control interface for an external entity to increase the performance of an individual sub-service (which is diagnosed as the root cause of the problem, say), or
- automatically control the sub-service until the average response time is back within limits.

One can extend these steps for any metric that is managed by the e-service management system. A list of all the metrics is presented in section 3. Table 1 summarizes this as a set of requirements that should be satisfied by the e-service management system.

Req. No.	Requirement
1.1	**Monitoring:** An e-service management should collect measurements and events from various information sources (provided by the e-service) and abstract those measurements into high level metrics. A list of the metrics for which monitoring should occur is described in section 3.
1.2	**Exposing service state:** An e-service management system may expose some or all of the computed metrics for use by other management systems, services, or graphical interface tools. A list of metrics that can be exposed as part of the service state is presented in section 3.
1.3	**Accepting input for desirable state:** An e-service management system may accept input for (or learn) desirable ranges for each of the monitored metrics. Such input could occur in the form of acceptable service levels (SLAs, baselines, thresholds, etc) or in the form of policies. A list of the metrics for which desirable levels could be set or learnt automatically is presented in section 3.
1.4	**Event notification:** An e-service management system may raise events whenever the monitored state of metric deviates from the desirable state. Event notification could also occur for other pre-determined changes in service state. A list of metrics for which events could be generated is given in section 3.
1.5	**Analysis and diagnosis:** An e-service management system may analyze/diagnose the root cause for discrepancies between the monitored state and the desirable state.
1.6	**External control:** An e-service management system may expose a set of high-level control interfaces for controlling the behavior of the service. Other services, management systems, or graphical interface tools could then use such interfaces to control the behavior of the service. A list of metrics that can be controlled through these external control interfaces is presented in section 3.
1.7	**Automatic control:** An e-service management system may automatically control the behavior of the e-service in case of deviations from desirable state. The method for controlling could be governed by pre-defined policies or by intelligent learning.
1.8	**Pro-active management:** An e-service management system may predict (or learn) the behavior of an e-service and start managing it pro-actively even before problems occur.

Table 1: Overall e-service management requirements

2.2 Levels of abstraction

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 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 or stop 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**
- low-level control interfaces to high-level control interfaces.

Table 2 summarizes this into requirements that should be satisfied by an e-service management system.

Req. No.	Requirement
2.1	**Levels of abstraction:** An e-service management should raise the level of abstraction of the low-level measurements collected from the e-service into high-level metrics. An e-service management system may also turn the low-level events into high-level events or automated actions, local models into globally consolidated models, and service-specific low-level control points into generic high-level e-service control interfaces.

Table 2: Levels of abstraction requirements

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.

To illustrate this with the example of the average response time metric mentioned earlier, information about average response times of a transaction can be organized and presented as:

- A report on average response times over the past 6 months, and a summary of the reasons for poor response times whenever they occurred,
- Current average response time of the transactions, averaged over the past 2 minutes and reasons for deviations from the expected normal,
- An estimated trend in the average response time over the next few hours, and probable reasons for the estimated deviations.

From this discussion, we can summarize the requirements on an e-service management system as shown in Table 3.

Req. No.	Requirement
3.1	**Past:** An e-service management may store and expose historical data for all the managed metrics. It may also use such data to perform historical analysis, planning, and reporting. A list of metrics along which such management could occur are presented in section 3.
3.2	**Present:** An e-service management system may manage an e-service in real-time. A list of metrics along which such management could occur in real-time are presented in section 3.
3.2	**Future:** An e-service management system may predict the values of certain metrics and use the estimated values to pro-actively manage an e-service. A list of metrics that can be predicted are presented in section 3.

Table 3: Temporal requirements

2.4 End-to-end e-service management

The functionality that is offered by a provider through an e-service is typically exposed to the consumers as a set of **business transactions**. In other words, consumers execute business transactions to utilize an e-service. Using the example of Figure 1, the book selling e-service provider could offer a catalog browsing transaction, adding books to shopping cart transaction, and a check-out transaction - all of which are part of the overall book selling e-service. Similarly, the shipping company could offer shipping transaction as part of its shipping service. In addition, the shipping, billing, and payments service providers can have negotiation transactions which result in a contract between the provider and the consumer (book selling e-service in this case) before any other transactions are executed. In general, all the business transactions initiated by a consumer and executed by the provider can fall into one of the following three stages of the e-service lifecycle - the negotiation stage, the execution or fulfillment stage, and the termination stage (Figure 2).

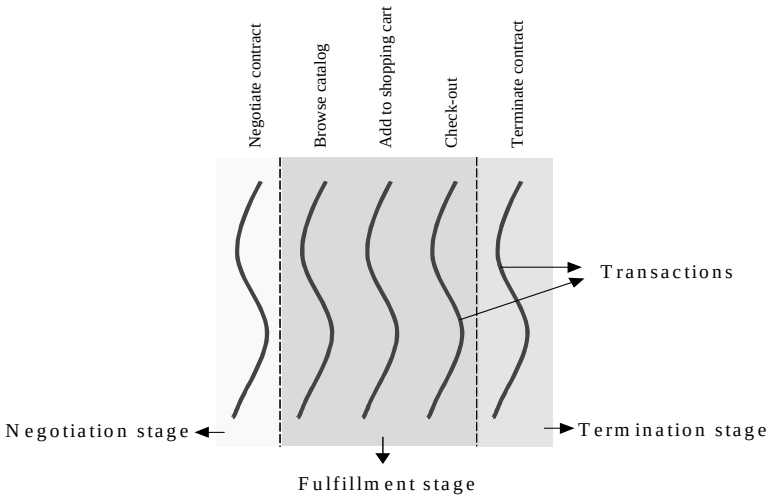


Figure 2: Different business transactions are executed by the e-service during different stages of its lifecycle.

Every single transaction that is initiated by a consumer follows a complex **workflow** of steps 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 could be taken (Figure 3):

1. **Consumers:** This is where most of the transactions are initiated. Measurements collected here would help in understanding the consumer's or end-user'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, only limited information regarding the management of their e-services might be exposed by these suppliers. 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. Measurements collected from composed e-services can be used to complete the end-to-end picture of a transaction workflow.

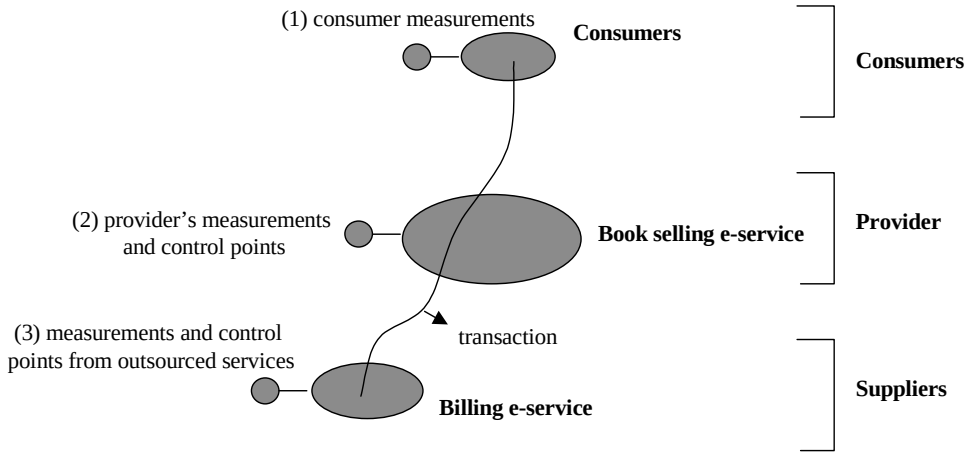


Figure 3: Measurements and control points for the management system to interface with managed system

The three types of sources listed above act as sources from where measurements could be collected and as control points where a management system could interface with the managed e-service in order to fix discovered problems. At each of these sources, either the managed e-service directly or the e-service middleware indirectly can provide the necessary measurements and control points. Table 4 summarizes the end-to-end management requirements.

Req. No.	Requirement
4.1	**End-to-end management:** An e-service management may collect measurements from the consumers, providers, and from external suppliers and use those measurements to perform end-to-end management of the e-service. This includes computing and exposing end-to-end metrics and using the manageability access points provided by the external suppliers to control the behavior of outsourced applications.

Table 4: End-to-end management requirements

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.

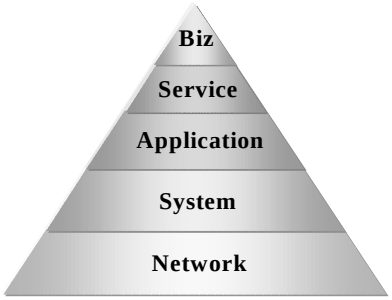


Figure 4: Service stack

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.

Table 5 summarizes these requirements:

Req. No.	Requirement
5.1	**Roll-up:** An e-service management system may roll-up the events, measurements, and metrics collected at lower levels in the service state to higher levels.
5.2	**Drill-down:** An e-service management system may (facilitate) drill-down across the various layers in the service stack to gain a better understanding the problems that occur at the service level.

Table 5: Top-to-bottom management requirements

3. E-service Management Metrics and Models

All the functionality of an e-service management can be classified into a set of management categories such as performance management, availability management, and contract management. In this section, we list the metrics and models along which management could occur under each of these categories (summarized in Figure 5). For each of the metrics and models presented in this section, an e-service management system can perform monitoring tasks (collect relevant measurements and compute metric values or construct a model), expose state of the service (expose computed metric values and constructed models), accept input to set desirable levels, identify and diagnose problems (whenever monitored values deviate from desirable values), and perform control tasks (expose control interfaces to fix problems or automatically bring the metrics to their desirable levels). Also, an e-service management system can store, compute, and expose the metrics and models along the three temporal dimensions - past or historic, present or real-time, and future or estimated. A complete description of these dimensions was presented in section 2.

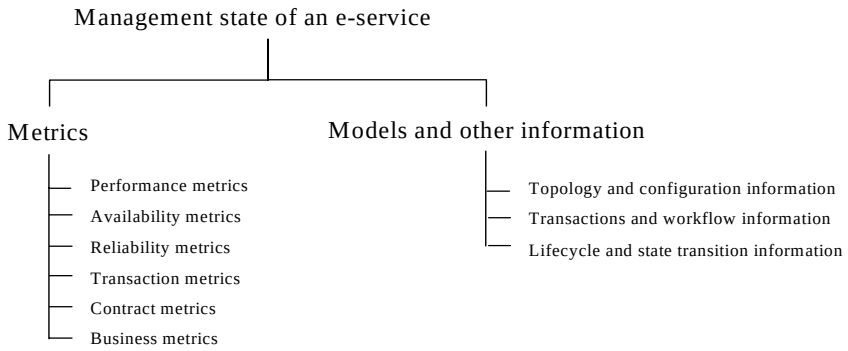


Figure 5: Metrics and models used by the e-service management system

3.1 Performance management

Maintaining the performance levels of an e- service is one of the primary tasks of an e-service management system. Some of the management features that fall under this category include monitoring the end-to-end response times of transactions, monitoring the sub-transaction response times through individual service elements, monitoring the response times and throughput against predetermined baselines, diagnosis for performance bottlenecks, and fixing performance problems by controlling service elements in a service-specific manner.

Table 6 lists a set of response time related metrics.

Req. No.	Requirement
6.1	Average response time for each transaction type.
6.2	Throughput for each transaction type.
6.3	Average number of simultaneous transactions processed (capacity).
6.4	Average response time for a certain consumer.
6.5	Throughput for a certain consumer.
6.6	Response time for a certain transaction.
6.7	Transaction – sub-transaction breakdown of the above metrics for each of the sub-services.

Table 6: Response time metrics requirements

Table 7 lists a set of workload related metrics.

Req. No.	Requirement
7.1	Number of pending transactions at any time
7.2	Average number of processed transactions over a period of time
7.3	Total number of processed transactions over a period of time
7.4	Number of connected consumers at any time
7.5	Average number of connected consumers over a period of time
7.6	Total number of connected consumers over a period of time
7.7	Number of requests from a consumer

Table 7: Workload metrics requirements

Table 8 lists a set of utilization metrics.

Req. No.	Requirement
8.1	Percentage utilization of the service
8.2	Idle time of the service

Table 8: Utilization metrics requirements

3.2 Availability management

Making sure that the service and all its transactions are available for consumer use is one of the essential tasks of an e-service management system. Features in this category can range from providing availability percentage information to replicating and moving services in order to make them more available. Table 9 lists a set of availability related metrics.

Req. No.	Requirement
9.1	Percentage of availability of the whole service over a period of time
9.2	Frequency of service unavailability
9.3	Percentage availability of a transaction supported by the service
9.4	Frequency of transaction unavailability
9.5	Correlated service – sub-service availabilities.

Table 9: Availability metrics requirements

3.3 Fault management

Application errors and the unpredictable nature of the distributed environment are two of the reasons for errors to occur in service transactions. The job of the e-service management system is to isolate the exact cause when such errors occur and to fix them automatically, if possible. Table 10 lists a set of metrics related to fault management.

Req. No.	Requirement
10.1	Number of failed transactions for each severity level over a period of time
10.2	Number of failed transactions for a certain consumer over a period of time
10.3	Number of failed transactions for a certain transaction type

Table 10: Fault metrics requirements

3.4 Transaction management

Being able to provide the progress of a transaction is one of the important features that could be offered by a management system. In addition, interfaces to abort transactions, to get an estimate on the time to complete the transaction, to change certain steps in the workflow (e.g., change the authorizing manager for an approval after the transaction is submitted), to configure additional parameters in the transaction, and to change the QoS level could also be provided by the management system. Table 11 lists a set of transaction related metrics can be used to manage the e-service.

Req. No.	Requirement
11.1	Progress of a transaction
11.2	Completion time of a transaction
11.3	Workflow model of a transaction
11.4	QoS level of a transaction

Table 11: Transaction metrics requirements

3.5 Contract management

The provider of an e-service can have several contracts - both with the consumers of the e-service and with other providers to whom parts of the service are outsourced. These contracts have several terms on quality of service assurance. It is in the best interest of the provider to make sure

that none of the contracts are being violated - and if they are, understand the frequency, reasons, and impact of those violations. An e-service management system can provide such contract verification and enforcement features.

Besides, an e-service management system can also provide some useful metrics regarding the offers and negotiations. Offers are initial documents that are exchanged between the providers and consumers before entering into a contract. Negotiation is the process of communication that occurs between providers and consumers whereby offers are turned into contracts.

A set of metrics related to contracts, offers, and negotiation are listed in Table 12.

Req. No.	Requirement
12.1	Number of offers processed
12.2	Number of offers rejected
12.3	Number of offers turned into contracts
12.4	Average number of negotiations before contract
12.5	Number of active contracts
12.6	Number of contracts violated over a period of time
12.7	Frequency of contract violation
12.8	Number of contracts violated for a certain consumer
12.9	Number of contracts violated by suppliers over a period of time

Table 12: Contract and offer metrics requirements

3.6 Lifecycle management

An e-service could be in various states of the lifecycle - started, initializing, negotiating, stopped etc. In general, an e-service can be thought of as executing a complex process that is governed by a state transition diagram. An e-service management system could provide functionality to monitor the state transitions of the e-service and to perform simple state transitions (such as restart the e-service or stop the e-service).

Table 13 summarizes the lifecycle-related information or models that can be used by the e-service management system:

Req. No.	Requirement
13.1	Lifecycle state of the e-service
13.2	State transition model

Table 13: Lifecycle model requirements

3.7 Configuration management

Understanding the structure of the e-service, the components that it uses, and the configuration parameters of all the components (e.g., location, components used etc.) is one of the pre-requisites for effectively managing the e-service. The e-service management system can collect, organize, and present all this information in a way that is useful for users of the management system. The same information could also be used internally by the management system to automatically infer and make decisions.

Table 14 summarizes the configuration-related information or models that can be used by the e-service management system:

Req. No.	Requirement
14.1	Topology of the e-service (service and sub-services)
14.2	Configuration parameters of the e-service

Table 14: Topology and configuration model requirements

3.8 Business metrics

By correlating the service state with other business information such as revenue, one could provide business metrics that are useful in assessing the business impact of a service. Examples of such metrics include revenue lost due to performance problems, revenue lost due to violated contracts, comparison of providers who can be used for an outsourcing deal, and customer satisfaction. The e-service management system may offer the business metrics listed in Table 15.

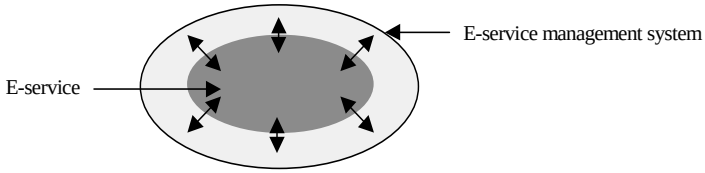
Req. No.	Requirement
15.1	Revenue generated from each service over a period of time
15.2	Cost of each service over a period of time
15.3	Penalties paid for contract violations over a period of time
15.4	Loses due to rejected offers
15.5	Unrealized revenue from outstanding offers and contracts
15.6	Number of current consumers for each service
15.7	Number of returning consumers for each service
15.8	Number of new consumers for each service over a period of time
15.9	Number of contracts for a particular consumer
15.10	Number of service providers of an external service at any time
15.11	Cost of contracts with a particular provider
15.12	Discounts due to contract violations from each provider

Table 15: Business metrics requirements

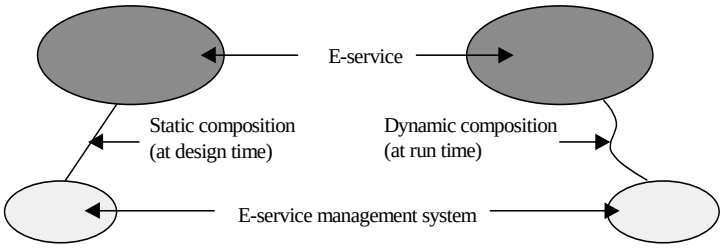
4. Other Requirements on E-service Management Systems

4.1 Relationship between the management system and managed e-service

There are many ways in which an e-service management system can be integrated with the managed e-service (Figure 6). In the first option, the management system is tightly coupled with the managed e-service. This is the way management systems have been built traditionally and it involves using non-standard and proprietary service interfaces to accomplish specialized management tasks. The second option is to build the e-service to be manageable, and then bolt-on (or statically compose) a modular management system to manage it. A manageable 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.



(a) Tightly coupled management system



(b) Statically composed management system

(c) Dynamically composed management system

Figure 6: Methods of applying management to the e-service

The interfaces that tie the management system and the managed e-service should be capable of exchanging information using internet protocols and standards wherever applicable. This will facilitate 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. A summary of all these requirements is presented in Table 16.

Req. No.	Requirement
16.1	**Composition:** The e-service management system may be statically composed into the managed e-service at design time or dynamically composed at run-time.
16.2	**Enterprise boundaries:** The e-service management system may be running on the intranet of the e-service provider or it may be running on the intranet of the management service provider (or both).
16.3	**Contract:** The composition of an e-service management system may be governed by pre-negotiated contracts, just-in-time negotiated contracts, or may not be bound by any contract.
16.4	**Service specification:** It should be possible to describe the management e-service and its features using an e-service specification standard.
16.5	**Offers and Advertisement:** The e-service management system should be able to advertise its features and negotiate upon the features with the e-service provider. Such negotiation should include management tasks, features within each task, sources of measurements, standards for exchange of information, management levels, and timeliness requirements on metrics besides other terms.
16.6	**Subscription vs Query:** The metrics that are provided by the management system could be delivered to the managed e-service either by subscription model or whenever the metrics are queried for.
16.7	**Standards:** The e-service management system should adapt existing standards, wherever applicable – in defining interfaces to the managed e-service, in communication protocols that are used to communicate with the managed e-service, and in vocabulary that is used in defining the management features and concepts.
16.8	**Internet standards:** Wherever applicable, internet standard languages and protocols should be used in interfacing the management system to the managed system.
16.9	**Security:** The communication between the management system and the managed system should conform to the security requirements imposed by the e-service provider. These requirements vary from the internal composition (management system runs on the e-service provider's intranet) to external composition (management system runs on management service provider's intranet and communicates with the managed e-service on the internet).

Table 16: Management system composition requirements

4.2 Managing the extra automation in e-services

An e-service could be built out of software agents [11] 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 fuzzy logic, neural nets, and genetic algorithms are being adopted to automate tasks that are traditionally handled by humans [12]. Such automation is also driven by the recent evolution of business communication languages and ontologies [13, 14, 15]. An e-service management system should be able to manage this extra automation by monitoring and controlling the agents that are used in implementing e-services and the conversations that occur between such agents.

The way a management system manages intelligent agents may be different from the way it manages traditional software elements [16]. These differences occur from the fact that agents are autonomous, non-deterministic, and intelligent. Agents may be constantly learning new facts about the environment and adapting themselves to behave differently. As a consequence, traditional management metrics such as average response time may not always be applicable.

Agents communicate with each other using agent communication languages. Examples of such languages include KQML [17], ACL [18], Speak-easy (from HP), or FLBC [19]. Messages built out of vocabularies from these languages constitute the basic unit of agent communication. High-level protocols for exchanging such messages are also being created. These protocols (also called conversations) can be thought of as a set of messages with a shared context that can be exchanged between agents. Examples of such protocols or conversations include various types of auctions [20] (e.g., dutch-auction, english-auction etc.) and negotiations (e.g., contact-net protocol). An e-service management system may manage the communications that occur between agents at the message level and at the conversation level. Some of the metrics that can be monitored are already covered in section 3.5 (under contract management). However, negotiations are just one kind of conversations that could occur between e-services. An e-service management system may offer more general conversation management metrics.

Req. No.	Requirement
17.1	**Agent management:** An e-service management system may manage the intelligent agents that are used in implementing e-services. For this purpose, it may define a new set of metrics or augment the metrics described in section 3.
17.2	**Agent communication management:** An e-service management may manage the communication between agents. This includes managing the messages that are exchanged between agents and the high-level conversations that occur between them.

Table 17: Automation management requirements

4.3 Managing the extra federation in e-services

As mentioned earlier, e-services can be composed out of other e-services. What this means is that an e-service management system may have to collect information from multiple enterprises in order to get a complete end-to-end picture of the offered service. Further, an e-service management system should expose portions of the internet service state to consumers, so that they can use it in managing their own services. A summary of the requirements is given in Table 18.

Req. No.	Requirement
18.1	**Cross-enterprise management:** An e-service management system may collect information from multiple suppliers in order to manage the overall e-service.
18.2	**Metric composition:** An e-service management may compose metrics collected from suppliers in order to compute and expose new metrics for the overall e-service.
18.3	**Exposing service state:** An e-service management system may expose portions of the internal service state for use by the consumers of the e-service. Such exposure could be governed by contracts between the provider and the consumers. Different consumers may have different degrees of exposure.

Table 18: Federation management requirements

4.4 Managing the extra dynamism in e-services

The composition of e-services to create new e-services can be static or dynamic. In the dynamic case, new or different e-services could be selected for composition every time such services are needed. This selection could be driven by some selection criteria that includes terms such as price, desired service levels (performance and availability), and manageability (the degree of exposure and the level of control offered by the composed service). Brokers and other mediators could also play a useful role in the selection process.

An e-service management system should first be able to cope with the dynamics of the e-service by taking into consideration the fact that the overall structure of the service could change frequently. Secondly, whenever multiple suppliers are used to offer the same service, an e-service management system should be able to provide useful metrics for comparing one supplier with one another. Examples of these metrics include a comparison of the contract violations between suppliers and a comparison of the potential benefits due to the differences in levels of manageability by different suppliers.

Req. No.	Requirement
19.1	**Handling the dynamism:** An e-service management system should be able to cope with the dynamics of the e-service by being able to handle dynamically changing suppliers.
19.2	**Comparison of choices:** An e-service management may provide comparison metrics that would help e-services in making better choices at selecting one supplier as opposed to the other.

Table 19: Dynamism management requirements

5. Conclusion

The range of features that can be supported by an e-service management system is large and complex. By organizing these features in various orthogonal dimensions such as management categories (performance management, availability management, etc.), management levels (monitor, expose, analyze, control, etc.), and temporal aspects (history, real-time, prediction), we can list a comprehensive set of requirements along each dimension. Many of these dimensions and require-

ments are applicable not only to e-service management systems but also to other management systems such as network, system, and application management. We also listed the requirements that are specific to the management of e-services such as contract management, managing the automation (e.g., agents and conversations), managing the federation (e.g., cross-enterprise monitoring and control), and managing the dynamism (e.g., comparison of suppliers).

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

1. 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.
2. Measurements: Real-time measurements whenever transactions are started and stopped at service and sub-service boundaries, availability heart-beats, contract and offer details, lifecycle state changes, etc.
3. Events: Events whenever errors occur within the e-service.
4. 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.

Standard interfaces have to be defined between the e-services and e-service management systems for exchanging the above information. Standard interfaces also need to be defined for accessing the service state computed by the e-service management system. This involves defining an e-service management ontology, representing the terms in that ontology using a cross-enterprise communication standard, and defining interfaces for accessing that information.

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. Oohama, 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]. **HP OpenView.** http://www.openview.hp.com.

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

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

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

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

[10]. 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.

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

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

[13]. 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.

[14]. **eCo framework.** http://eco.commerce.net.

[15]. **Common Business Library.** http://www.commerceone.com/solutions/xml/cbl/cbl.htm.

[16]. S. Hagg. **A sentinel approach to fault handling in multi-agent systems.** *Proceedings of the Second Australian Workshop on Distributed AI, in conjunction with Fourth Pacific Rim International Conference on Artificial Intelligence (PRICAI'96)*. Cairns, Australia August 27, 1996.

[17]. 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

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

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

[20]. M. Tsvetovatyy and M. Gini. **Toward a virtual marketplace: Architectures and strategies.** *Proceedings of the First International Conference on the Practical Applications of Intelligent Agents and Multi-Agent Technology (PAAM'96),* London, U.K., April 1996, pp.597-613.