Lockheed Martin Information Technology
3114 Lord Baltimore Drive
Baltimore, MD 21244

LOCKHEED MARTIN

WORK PRODUCT REPORT

Title of Report:	Web Application Monitoring Approach Document
Contractor:	Lockheed Martin
Contract Number:	600-99-34628
Task Title:	Internet Architecture Support
Task Number:	5-610
Work Order Title:	AeDib Architecture Support
Work Order No.:	5-610.80
Work Product No.:	80-11
Date Submitted:	January 26, 2004
Submitted By:	Steven Schulman

Task Manager: Subodh Atal **Date:** January 26, 2004

Manager Name

WEB APPLICATION MONITORING APPROACH DOCUMENT

PREPARED BY: LOCKHEED MARTIN

JANUARY 26, 2004

TABLE OF CONTENTS

LIST OF FIGURES

1.0 Introduction

Under Work Order 5-610.80, AeDib Architecture Support, the Social Security Administration (SSA) tasked Lockheed Martin with researching concepts and implementation options of Web application monitoring. Pursuant to high-level client discussions, this document provides:

- An understanding of the Office of Systems Electronics Services (OSES) Web Application Monitoring (WAM) vision,
- A broad discussion of potential research topics incorporated by this vision,
- Elements of an approach for engaging this research effort, and
- A brief summary of the research work that has already been accomplished.

1.1 BACKGROUND

SSA's Web site provides access to a number of applications that support citizens, businesses and other government agencies. While broad statistics regarding customer interaction with these online services are available, such basic measures of Web application behavior that track quantitative site statistics regarding application events do not necessarily indicate the quality of the services provided.[1] From a user's perspective, aspects of Web application service quality may include availability, responsiveness, consistency, predictability, and reliability.

1.2 MOTIVATION

The business motivation driving research into WAM is the recognition that there are currently no mechanisms in place that monitor user perceptions of the quality of services delivered by SSA's Web applications. Various organizations within SSA currently monitor different aspects of the technology environment that enable Web applications[2]. However, there is no current monitoring of run-time behavior of Web applications in terms of their impact on the user experience, or of the efficacy with which the intended business services are actually supplied. The business focus of WAM is to monitor the Web applications to determine if the expected application behavior is occurring.

---

[1] Bouch, A., Kuchinsky, A., Bhatti, N., Quality is in the Eye of the Beholder: Meeting Users' Requirements for Internet Quality of Service, Proceedings of the SIGCHI conference on Human factors in computing systems, ACM, April 2000.

[2] OTSO, Web Application Response-time Product (WARP); Stress testing via LoadRunner.

1.3 VISION

The business vision for WAM encompasses more than the passive observation of the quality of Internet business services. This vision for a fully implemented WAM architecture is based on the derivation of Web application management strategies from monitoring observations and the use of this information as feedback to system controllers. For example, knowing that 40% of users abandon a Web application on page 7 of a 12-page application is not sufficient. This implies that WAM must not only know its operating environment but also be able to 'reason' over it. In other words, the WAM must be able to analyze its environment in a logical, orderly, and robust manner, and be able to infer conclusions.

1.4 INITIAL UNDERSTANDING

Certain project goals, requirements, constraints, expectations, and exclusions were discussed at the initial Lockheed Martin/SSA meeting on Web application monitoring. They are presented in this document as potential guiding principles for the research and development effort moving forward on this project.

1.4.1 Goals

The primary goals of WAM are:

1. *The system should be general in nature and be able to monitor all types of Web applications*. This implies that the architecture and design of WAM should focus at the meta-application level. That is, the focus should be on elements that are common to all Web applications.
2. *The system should learn over time*. This implies that WAM will maintain and evolve a model of Web application behavior with respect to its run-time environment. It also implies that WAM will use this model to evaluate Web application behavior against success criteria, and flag anomalous or otherwise suspect behavior.
3. *The system should not require additional programming effort or modification*. This could be accomplished in various ways. For instance, WAM could present a declarative interface to individuals to modify program parameters; some level of autonomy could be designed into the system so that it could alter its own program parameters.

1.4.2 Requirements, Constraints, and Expectations

WAM requirements, constraints, and expectations include the following items:

1. Provide the ability to individually monitor each application within the system.
2. Provide a visual dashboard user interface.
3. Minimize the number of inconsequential error messages. This is a qualitative evaluation criterion that articulates an expectation for WAM.
4. Have a minimum impact on applications. This is an implementation constraint that could limit the range of available technologies.
5. Operations may be based on packet or other information passing through the network. This is an implementation alternative.
6. Implement monitoring external to applications. This is an implementation constraint.
7. Base tracking on existing information (e.g., page ID). This is an implementation constraint that may or may not have an impact on the fidelity with which Web applications can be monitored.
8. As long as there is no impact on operations, the use of log files is possible. This is an implementation constraint that may or may not have an impact on the fidelity with which of Web applications can be monitored.
9. Track application behavior at a deeper level than existing monitoring systems. This reinforces the expectation that WAM should not duplicate ongoing monitoring activities by other components in SSA.

1.4.3 Exclusions

The following items were enumerated as exclusion items that WAM will not do.

1. Provide measurement for Service Level Agreements (SLA).
2. Provide application profiling[3].
3. Provide throughput and utilization tracking.
4. Provide application optimization.

[3] Application profiling in a controlled environment is already being addressed with the use JProbe with CICADA.

1.5 DOCUMENT OVERVIEW

This document consists of four sections and five appendices.

- Section 1 provides the basis and justification for this document.
- Section 2 discusses the proposed activities and research approach to determine options for implementing the OSES WAM.
- Section 3 presents a summary of the current state of research relating to this project.
- Section 4 provides a summary of this document.

The appendices sections contain additional research information for this paper and include:

- Appendix A provides preliminary research and outlines commercial off-the-shelf (COTS) software products that may meet OSES WAM requirements.
- Appendix B contains a preliminary review of the dashboard research with details of dashboard-centric presentation software.
- Appendix C provides a survey of COTS software that brings intelligence to the Web application marketplace.
- Appendix D is a bibliography section containing references to technical research papers focused on issues directly related to fully understanding and supporting OSES' WAM vision.
- Appendix E is a glossary containing definitions for technical terms used in this document.

2.0 Iterative Approach

An iterative approach to the WAM research project will consist of several activities – scoping, researching, documenting, and prototyping. Activities can freely cycle between and among each project level and activity type.

2.1 SCOPE

Scoping activities define the boundaries of the intended work, and are accomplished by listing the assumptions, conditions, constraints, and other pre-conditions, which establish the approach or understanding of the system under consideration.

2.2 RESEARCH

Extensive research activities will be conducted to elucidate the scoped problem. Research currently in progress is focused on:

- The current state of commercial and open source software, and its ability, directly or with customization, to address OSES WAM requirements, and
- The current state of technology that will support the development of a WAM system that addresses OSES specific requirements.

While not a complete list, identified areas that may be included in future research are included in this section.

2.2.1 Network level monitoring research

Network level monitoring is currently being provided by the Office of Telecommunications and Systems Operations (OTSO), and will not be addressed by the WAM system. However, because network level metrics are available without additional effort, the metrics at this level will be evaluated to determine if they can provide higher levels of insight to a more sophisticated system. There may be instances where specific network level monitoring results may, in combination with other metrics, provide results that are relevant to application level monitoring.

Some examples of possible research areas for network level monitoring and network technologies are:

- HTTP fundamentals,
- Network protocol fundamentals,
- End-to-end service performance measurement and display,
- Network packet tracing,

- Passive application monitoring,
- TCP transaction analysis, measurement, and display,
- TCP-IP fundamentals,
- Simple Network Management Protocol (SNMP),
- Service time estimating techniques, and
- Internet activity monitoring, measurement, and display.

2.2.2 Usability related research

An expected outcome of the WAM system is verifying that business services are effectively delivered to end users. Usability factors and issues play an important role in determining the total effectiveness of the Web application.

Some examples of possible research areas for usability factors and issues are:

- Architecting for usability,
- Designing for usability,
- Remote usability testing,
- Human-computer interaction,
- Effects of the age demographics of users on Web site usability,
- User interface event monitoring, measurement, and display,
- Usability related qualities of service,
- Usefulness related qualities of service,
- User perceived latency versus network latency,
- Metrics associated with usability,
- Automating the capture, analysis and display of usability metrics, and
- Human perception, cognition and information processing.

2.2.3 Web application related research

There are general characteristics of Web applications that may have an impact on service delivery and may need to be monitored.

Some examples of possible research areas for Web application general characteristics are:

- Web site design,
- Web site navigation quality assessment,
- Web site usability metrics, measurement, and display,
- Web site visualization techniques,
- Web communication patterns between government and citizens,
- Web site reliability assessment, measurement, and display,

- Web site interaction logging file format,
- Web transaction aliasing, and
- Cache coherency and replacement.

2.2.4 Knowledge engineering research

Knowledge engineering is a methodology to gather and structure knowledge in a manner conducive to using that knowledge in a computer system. Knowledge engineering research areas have been grouped into two broad topic areas, ontology and intelligent information systems. The results of this research may provide support for addressing the goal of integrating learning into the WAM system.

2.2.4.1 Ontology

Ontology is a formal approach to defining a domain. It results in a standard vocabulary, with synonyms identified for equivalent terminology. The use of ontology reduces the potential confusion from marketing-based language. The use of ontology tools may provide the basis for addressing OSES WAM learning goal. Potential areas related to ontology that may be researched are ontology engineering and computing, the use of ontology to support WAM learning, and how domain and procedural knowledge can be integrated into the ontology. Additional material can be located at Internet sites referenced in the definition of ontology in *Appendix E*.

2.2.4.2 Intelligent Information Systems

The following may be researched to support integrating learning capabilities into the WAM system:

- Semantics,
- Semantics and contextual awareness,
- Logic,
- Artificial intelligence,
- Business intelligence,
- Neural networks,
- Statistical analysis,
- Fuzzy logic,
- Agent-based architectures,
- Expert systems.

2.2.5 Commercial research

One activity for WAM is to examine software currently on the market in order to determine if there are products available that have the functionality to meet OSES WAM goals. A potential concern with using COTS software is that the basic technology assumptions[4] in such software are often hidden behind marketing terminology. This proprietary marketing language may be unreliable when comparing or evaluating software packages.

2.2.6 Organizational research

SSA components are a valuable resource for research information related to WAM. For example the Office of Telecommunications and Systems Operations, Division of Operational Capacity Performance Management (OTSO-DOCPM) is responsible for the analysis of configuration, topology, connectivity, automation, and availability of SSA's national network. It is also involved in AeDib infrastructure implementation and support. Its analysis work supports performance management, tracks resource utilization, and assists in capacity planning. Reports and data resulting from this work may prove to be useful to the WAM, and should be included in the research process. An example of the documentation associated with a discovered information source can be found in *Section 2.3 Documentation*.

2.3 DOCUMENTATION

All aspects of executing work on this project will be documented. Research findings will be documented in a manner that allows complete and intuitive access to the information by all team members. In addition, research findings may also have practical and direct WAM relevance, such as the discovery of information resources produced by OTSO-DOCPM. Documenting a discovered information resource may include the following tasks:

1. Obtain samples of the information resource files.
2. Identify the business value to WAM of the information resource.
3. Make decision to profile information resource (or not).
4. Create profile of the information resource with the following details:
    - Owner,
    - Production frequency,
    - Availability,
    - Accessibility,
    - Format,

[4] Software technology assumptions – the technology upon which the software is constructed or within which the software is expected to operate.

- Semantics,
- Context,
- Identify information resource data elements, and
- Create data element profile (details in sub bullets).
  - Syntax,
  - Semantics,
  - Context, and
  - WAM relevance.

2.4 PROTOTYPES

It may be necessary to develop multiple prototypes for potential approaches as they are identified. These prototypes can help prove the feasibility of the approaches in providing WAM functionality that has been identified as part of the scope, research, document, and model stages. Functional prototypes can be used to provide demonstrations, and thereby obtain feedback assurance that OSES goals are being addressed and that what was defined during the scope stage has been successfully provided.

3.0 Current Work

Based on the goals defined, research has begun in many areas. The following is a summary of research results to date, and additional details can be found *Section 5.0 – Appendices*.

3.1 SURVEY OF WAM TECHNIQUES

A large number of products from a variety of vendors are available for addressing Web application monitoring. The initial survey[5] of the market completed to date focuses on more than 40 companies. Some of the more common techniques used in these products are discussed in the following sections**.**

3.1.1 Web Site Monitoring

This type of monitoring tests the accessibility of a Web site from multiple locations around the world. It checks the time taken by a server to respond to the monitoring agent's request, filters on a specific phrase or keyword, ensures that the page was not altered, and that the page has the proper content.

3.1.2 Obtrusive Monitoring

In *obtrusive* monitoring, also known as *intrusive* or *agent-based* monitoring, typical user transaction steps are recorded and later transmitted to the server. An agent-based synthetic user then uses this information to repeat the same transaction steps from multiple locations around the globe. This agent monitors the transaction process to ensure each step within the application is running correctly. This type of application may have online reporting and problem escalation capability, such as alphanumeric paging, e-mail, and Short Message Service (SMS) instant messages. It may also be capable of initiating corrective measures such as restarting stopped or 'locked' application services.

The obtrusive agents can be *active*, *passive*, or a combination of both.

3.1.2.1 Active Agents

Active agents are programs that simulate user actions and check for expected results. They act within a session and send test data to the system at a fixed interval using a predefined script. Active agents can perform a full range of system and component testing. Examples of active agents are:

- Ping monitor: Validates that the machine reachable over the network and is at least functioning enough to reply,
- DNS monitor: Checks that the machine name is mapped to a network address,
- Process monitor: Assures the process is still active and is not overloading resources,

[5] For details of the initial survey, refer to *Appendix A – A Survey of COTS Web Application Monitoring Products*.

- Central Processing Unit (CPU) monitor: Checks utilization and triggers notification in case of consistent overloading, and
- Memory monitors: Measures memory usage and paging activity.

3.1.2.2 Passive Agents

Unlike active agents, passive agents do not create artificial sessions; rather, they monitor the system by tracking network traffic. Thus, a passive agent might track the number of packets passing though a router or count the accesses from different locations. Passive agents do not increase workload and do not affect the network itself. They more accurately reflect actual usage of the system. One problem with passive agents is that they are dependent on actual traffic on the network. It may be possible to combine active and passive agents to better serve system-monitoring requirements.

3.1.3 Non-Obtrusive Monitoring

Traditionally, monitoring agents have been used to track Web application performance and availability. In most cases, additional monitoring agents are installed throughout an enterprise to monitor locally installed software packages (e.g., database, application, middleware). This often leads to multiple agents and increased cost of maintenance.[6]

As an alternative, *non-obtrusive* monitoring, also known as *non-intrusive* or *agentless* monitoring, operates on the server side and captures details for each request made to the server. This can assist with the immediate identification of a suspected problem area. The overall cost for monitoring an application environment may be less because it eliminates the time, effort and expense of programming, installing and managing intrusive software agents. However, this software executes at the server location and may negatively impact this application node. Another consideration is the effect of proxy redirection and load balancing of incoming traffic and its impact on client session information gathering. In addition, remote monitoring requires constant communication over the network to gather metrics. This increases network traffic, which in turn can reduce the quantity of data than can be collected.

3.1.4 Probes

Probes are hardware monitoring devices built into or attached to the network. These probes can be used to capture performance, and collect fault and availability information from APIs, databases, devices, log files, and other utilities. Probes normally do not add overhead to the network because they are specialized, external attachments.

---

[6] *An Agentless World, Service Management Strategies*, http://www.silasreveille.com/data/UploadFiles/Whitepapers/Whitepaper8.pdf, Corey Ferengul, metagroup.com

3.1.5 Web Trends Analysis

Different methodologies have been developed for analyzing Web site traffic trends, including log analysis, page tagging, and independent data gathering elements. Each methodology is outlined in the following sections.

3.1.5.1 Server Log Analysis

Server log analysis is historically the simplest means of collecting data about a Web site. This methodology consists of parsing and cleansing the log files which are created when a visitor requests any page from a Web server or e-commerce application server. The chronological log file may be transferred to a database and analyzed. Since the log data contains no information about page display durations or about cached activity (Back requests etc.), subsequent analyses can be entirely in error.

A Web server monitor is a program that is installed and executed on the Web server, to monitor usage of the server. It can be used to determine how long a transaction spends being processed by the Web server. This can be used to distinguish between the time spent on the front-end (client and Internet) and the time spent on the back-end (Web server and service.) It can also be used to identify where problems are caused by specific servers.

3.1.5.2 Page Tagging

In this methodology, the user voluntarily provides the information requested. A graphic or script tag is inserted on each page of the Web site from which information is required. In addition, these solutions are often managed by a third-party, so one has to rely on third-party cookies from the customers to track return customers.

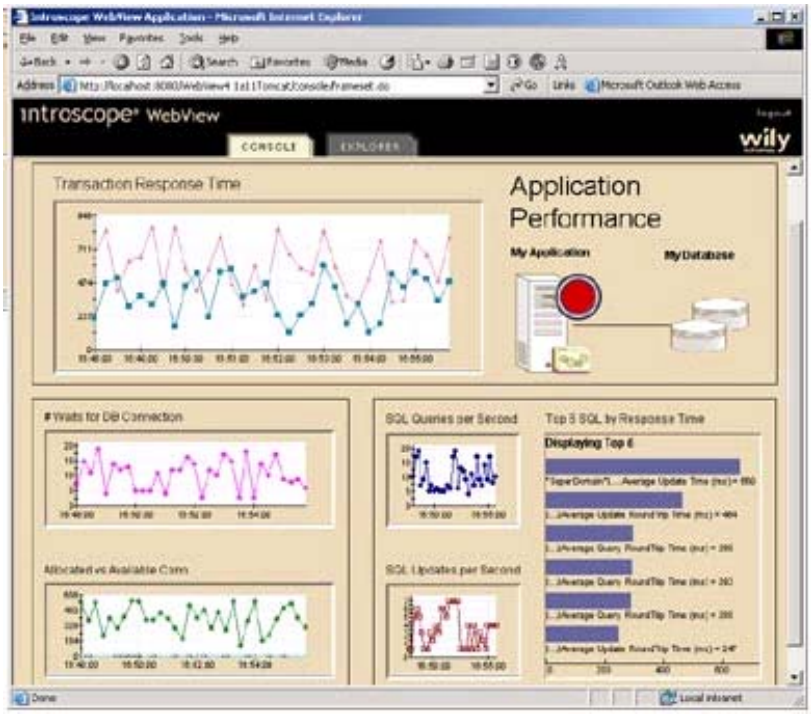
3.1.5.3 Independent Data Gathering Elements (DGEs)

DGEs collect the alarms, events, and polls from the various infrastructural elements such as servers, routers, databases, or application. Each DGE has its own distributed relational database. A DGE collects data according to definitions it has been given. These definitions include which devices it should monitor, how frequently it should monitor them, and what it should do with the results. This information is then progressively summarized over time and analyzed relative to thresholds, key events, alarms, and appropriate actions taken. Queries and reports are then prepared and sent to IT personnel.[7]

[7] Source: http://www.fidelia.com/products/architecture.php

3.2 SURVEY OF DIGITAL DASHBOARDS

After collecting the specified data, a Web application monitoring system must be able to display the data to users in a clear, accurate, and easily-understood manner. While alternative forms of Graphical User Interfaces (GUIs) may be evaluated, dashboards are a common GUI used to provide users with visual and intuitive summary information, drill-down access to details and problem alert notifications. A wide variety of dashboard alternatives is available[8]. Refer to *Figure 1 – Typical Digital Dashboards* for typical digital dashboard displays.



[8] For more survey information, refer to *Appendix B – A Survey of COTS Dashboards*.

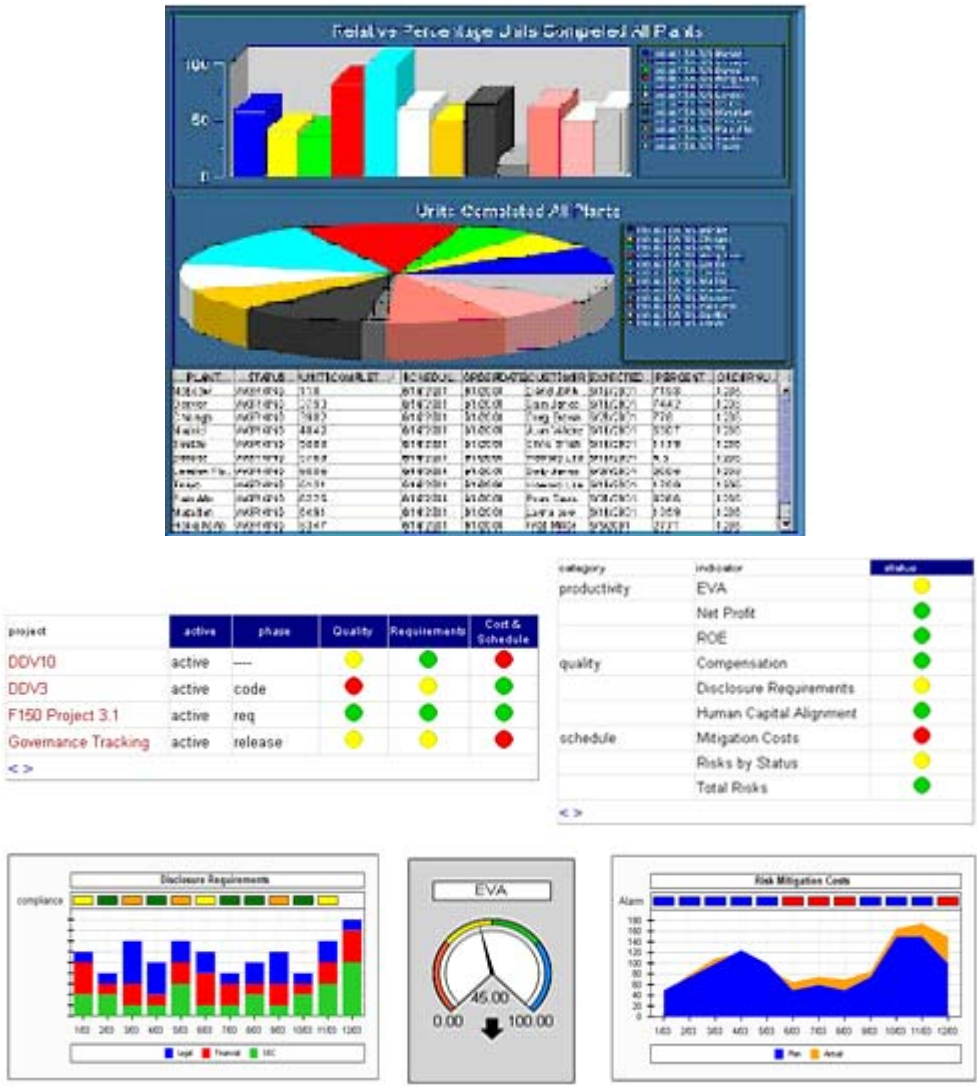


Figure 1 – Typical Digital Dashboards

Some vendor dashboards seamlessly integrate with any relational database and can work within any of the leading application-server environments. This can provide flexibility in converting the metrics from the WAM into the dashboard presentation.

The use of a dashboard supports the following three levels of user roles:

- User: The ability to view charts and dashboards in categories authorized by administration,
- Analyst: Additional abilities to create drill-through paths and ad hoc data extraction, and
- Administration: Additional abilities to create and manage users, assign privileges, and control user access.

3.2.1 Drill down

An important benefit of dashboards is the ability to summarize and hide details, as shown in *Figure 2 – Examples of Drilling Down from Summary to Details*. However, there are times that the details become important. At those times, dashboards provide the capability to drill down and provide access to details. This feature helps provide a deeper understanding of a problem and to recognize and understand the contributing factors associated with a problem. The WAM dashboard may need to:

- Automatically roll-up monitoring data into an easy-to-read summary graphics that provides an online summary of all applications of interest, and
- Drill down details to enable proceeding with a minimum of mouse clicks from a summary view to the performance details of specific applications and indicators.

Figure 2 – Examples of Drilling Down from Summary to Details

3.2.2 Notification

A dashboard may be able to act both proactively and reactively. It may provide real-time Web access information that alerts users, and it may support users' queries on the status of Web applications. The use of dashboards as part of the WAM system can reduce the inherent risks in delivering high-quality application and increases the availability of applications. Color coding can make it easier for the user to

distinguish problems that are urgent from those that are impending, as well as to identify areas that are executing without problems.

Preliminary research has identified several dashboards that are currently available in the market. These products are being analyzed to determine how each addresses OSES-specific WAM goals. The following dashboard abilities are among those that have been identified as having potential importance:

- customize dashboards for rapid deployment with no or minimal programming,
- use included monitoring displays or to build custom displays and dashboards without programming,
- view real-time data from multiple sources and custom data (e.g., XML, SQL, TIBCO™, JMS, SNMP),
- archive historical data for trend analysis,
- perform data calculations (for example, sum of all values in a table column),
- interface directly with all existing databases,
- run on any platform and support all server environments,
- integrate easily with any Web application server,
- view reports from heterogeneous systems in a single consolidated report, and
- provide full compliance with Federal accessibility and Section 508 requirements.

3.3 SURVEY OF INTELLIGENT TOOLS

An initial survey of intelligent technologies for Web monitoring has focused on identifying companies producing adaptive, self-learning and intelligent solutions[9]. Major concepts in such technologies are discussed in the following sections.

3.3.1 Predictive Statistical Modeling

In data mining, statistical predictive modeling is a technique used to predict future behavior and anticipate the consequences of change. Data is collected, a statistical model is formulated, predictions are made and the model is validated (or revised) as additional data becomes available. The model can be a simple linear equation or it can be a complex neural network mapped out by sophisticated software. As an example, predictive statistical model software analyzes data from millions of online sales to extract the profile of fraudulent transactions. Culling data from large, historical databases, the software develops a mathematical formula and applies it in real-time to incoming transactions[10].

[9] For more survey information, refer to *Appendix C – A Survey of COTS Intelligent Tools.*

[10] Source: http://retailindustry.about.com/library/uc/02/uc_fraud2.htm

3.3.2 Neural Networks

A neural network is a system of programs and data structures that approximates the operation of the human brain. It usually involves a large number of processors operating in parallel, each with its own small sphere of knowledge and access to data in its local memory. Typically, a neural network is initially given large amounts of data and rules about data relationships. A program can then tell the neural network how to behave in response to an external stimulus or can initiate activity on its own, within the limits of its access to the external world[11].

Neural agents, also known as neugents, are based on neural-network technology. They can be used to predict customer needs and preferences, forecast outcomes, and dynamically update in real-time to fulfill customer needs. They are true intelligent systems because they track Web site visitors, and not just hits, views, or impressions. Neugents are able to accurately report new visitors, repeat visitors, visit count, and distinct visitors.

[11] Source: http://searchcrm.techtarget.com/sDefinition/0,,sid11_gci809473,00.html

4.0 Summary

This paper provides an approach for researching concepts and implementation options of Web application monitoring. The WAM system will extend the monitoring process where OTSO leaves off. One of OTSO's many responsibilities is to ensure that the Web service infrastructure is always available to support the Web application. The challenge is to assure that the Web application is performing as expected. Since the concept of 'as expected' is dynamic, the monitoring process must have awareness and learning capability to provide meaningful results.

At this early stage of the research process, it is not yet clear which areas of investigation will prove to be most productive. An initial list of areas that preliminary research has identified as a starting point for further research is presented, together with an iterative approach to the research process. This document presents the results of a preliminary survey in the field, including Web application monitoring techniques, digital dashboards, and intelligent tools. This document also includes appendices that identify a number of Web application monitoring tools, digital dashboards and intelligent tools, and list their key features. In addition, a comprehensive list of references on several subtopics within the field, and a glossary of key terms are included.

5.0 Appendices

Please refer to the following pages for additional research information.

5.1 APPENDIX A – A SURVEY OF COTS WEB APPLICATION MONITORING PRODUCTS

S. NO.	VENDOR	TOOL	SUMMARY
1	Chevin – partnered with netfusioninc.com	TeVISTA	A suite of network monitoring tools employs synthetic users. This real-time solution provides 95% of the functionality of an expensive hardware sniffer solution at a fraction of the cost. • TeVISTA uses software agents (Visibility Agents) that are sent by e-mail to all devices on the network and in less than ten minutes can be installed and collecting data. • TeVISTA Enterprise is a comprehensive SNMP device management that gives detail information and statistics for analysis. Unlimited remote enterprise management pollers, includes EtherPeek, the diagnostic component of TeVISTA. SNMP provides both real time network status and statistics and historical distributed management software. Monitor and manage SNMP agents, WEB, E-mail, FTP etc. • The TeVISTA Performance Manager (TPM) is an end to end service that tests delivery of an application. It experiences the network as a real user would by using synthetic users to simulate the experience and then send detailed reports that help with SLAs. TPM will reveal where a traffic jam is happening and then Visibility Agents report on the device, the protocol and/or the conversation.
2	RTTS	RTTS	Collects performance data from across the entire application infrastructure, including applications, transactions, URLs, databases, network devices, firewalls, servers and custom applications. These tools are configured to monitor the infrastructure from an integrated view and to identify when key thresholds and SLAs have been exceeded. They then perform root cause analyses to help pinpoint the problem so that it can be addressed before the clients are impacted.
3	Dotcom-Monitor.com	Macro Recorder	Agent-based, synthetic user. Performs Web monitoring, application monitoring and network monitoring, along with load test. Macro recorder is a free, windows-based program that records Web site transactions, shopping-carts, online ordering systems, etc. like a regular user. It uploads the steps in the transaction to a server. Their agents around the globe repeat the steps in the transaction. Online reporting and problem escalation.

S. NO.	VENDOR	TOOL	SUMMARY
4	Centrisoft	Centerwise™ Suite	Centerwise™, a distributed software solution suite for managing and optimizing application traffic flow and application performance across enterprise networks. The Centerwise suite consists of two products: • Data Point gathers information on what is actually happening at the end user level. • Control Point aligns application performance and usage to business priorities established by the enterprise. Either Data Point or Control Point can be purchased as a separate product, or a customer can address a broader scope of Application Performance Management by purchasing the Centerwise Suite. Centerwise enables enterprises to learn how the applications are performing at the end-user level, the demands being placed by such applications on network resources, and, unlike other performance management software, actually resolve the problems and improve network performance.
5	Application Monitoring	Packeteer	Packet-based traffic monitoring, network utilization, centralized reporting as one component of application traffic management system.

S. NO.	VENDOR	TOOL	SUMMARY
6	IBM	Tivoli	• IBM Tivoli Monitoring Automates monitoring of essential system resources, to detect bottlenecks and potential problems, and to automatically recover from critical situations • IBM Tivoli Monitoring for Applications Delivers higher availability and optimized performance of their mySAP.com and Siebel e-business applications • IBM Tivoli Monitoring for Business Integration Manages IBM's WebSphere MQ and WebSphere MQ Integrator from one central console • IBM Tivoli Monitoring for Databases Ensures the availability and optimal performance of DB2, Oracle, and Informix servers • IBM Tivoli Monitoring for Messaging and Collaboration Deploys best practices resource models to monitor and cure problems that arise in a messaging and collaboration environment • IBM Tivoli Monitoring for Network Performance Gathers, analyzes and reports information on SNA systems and network components • IBM Tivoli Monitoring for Transaction Performance Monitors the performance and availability of e-business and enterprise transactions to ensure a positive customer experience • IBM Tivoli Monitoring for Web Infrastructure Ensures the optimal performance and availability of both application servers and the associated Web servers that feed them IBM Tivoli NetView Proactively manages network resources.
7	Heroix	Heroix eQ: Solution Studio	Associate application, database, server, and infrastructure events with the business activities they affect; Heroix eQ will monitor sets of resources, then integrate the information, and respond by issuing alerts and/or commands. Monitor an application's "signature" to verify proper operation; Heroix eQ detects whatever combinations of operational behaviors we specify – even across infrastructure components, servers, and OS platforms – that may indicate expected operation or may signal trouble ahead. Check how our application is writing to a database; Heroix eQ can issue a query and evaluate the results. Permits manipulation of data from a variety of sources, including native Heroix eQ statistics, performance counters, SNMP traps and variables, database queries, ASCII files, event logs, and more.

S. NO.	VENDOR	TOOL	SUMMARY
8	Exact Solutions	iWatch	Non-obtrusive (non-intrusive, non-agent) monitoring, demo available (e-mail sent) • Database: 24 x 7 Operational Database. No limit on database size. • Depth of Data for Analysis: iWatch captures every detail for each request made to the server which can assist in zoning down to problem areas of the application instantly. • Usability: Web based reporting tool to analyze captured data to highlight performance and usage trends. • High Scalability: iWatch has hardware based solutions for high bandwidth environments like trading desks etc. This allows it to capture data without dropping any packets. • Non-Agent Based Monitoring: Having no agents ensures that the application environment is free of any unexpected external factors (memory leaks, additional disk I/O, sharing of CPU resources, etc.). iWatch is the only non-agent monitoring solution in the market. • Architecture: Server-side monitoring ensures that performance metrics for each client accessing the server is captured. With agents, we have to ensure that the agent software is installed on each client. • Depth of Data for Analysis: iWatch captures every detail for each request made to the server which can assist in zoning down to problem areas of the application instantly. Agents, on the other hand, provide statistical summaries which can often be misleading. • Cost: Overall cost for monitoring an application environment is a lot more affordable and predictive with iWatch. • Usability: Agent based monitoring cannot be used for Web based applications.
9	BMC Software	PATROL Express	Monitoring is accomplished remotely (agentless) with no software residing on the elements being monitored. Also monitors the performance and availability of Web transactions. Monitoring both infrastructure and transactions enables IT to quickly determine how infrastructure problems affect the end-user experience.
10	AlertSite	AlertSite	Synthetic user to test business transactions A professional services company, no tool to evaluate
11	ipMonitor.com Deepmetrix.com	ipMonitor	TCP/IP and business transaction monitoring

S. NO.	VENDOR	TOOL	SUMMARY
12	eG Innovations	eG Enterprise	Agent-based, transaction processing
13	Webmetrics	AppMonitor	For Web monitoring, transaction monitoring, load testing, system monitoring. AppMonitor is a completely outsourced service that requires no software to be downloaded. AppMonitor scripts are executed over GlobalWatch network, which is powered by Tier One infrastructure and connectivity providers and spans 17 cities worldwide. AppMonitor provides detailed HTML- and Flash-based performance reports, which can be generated ad hoc or scheduled for e-mail delivery on a daily, weekly and/or monthly basis.
14	Tevron	CitraTest Acutest Citratest Citra_vu	Windows-based tools, executes synthetic transactions, load and functional testing, client-side only.
15	Sane	NetTracker eBusiness Edition	Log analysis tool, Website traffic monitoring, clickstream analysis
16	Fidelia	NetVigil	No central database, independent data gathering elements: Most systems today require a central database for historical data reporting. This severely limits scalability. Fidelia's BVE software removes the need for a central database through its coordination of multiple independent DGEs.
17	Bay Mountain	IT Appliance	System and network monitoring, transaction and application monitoring. The heart of the system is a SQL-based program that collects measurement data from anything on the network, including hardware, applications, transactions, operating systems, and power levels. All measurements are stored on the IT Appliance's hard drive. This data can be accessed via ODBC by a program such as Crystal Reports® for trend analysis, SLA verification, planning for system expansion, or any other types of data mining or analysis.
18a	DLT Solutions: Quest Software	Foglight®	24x7 Application Monitoring, ensure application uptime and performance. On GSA schedule.

S. NO.	VENDOR	TOOL	SUMMARY
18b	Quest Software	Foglight	• Provides a real-time, correlated view of the entire application stack influencing application performance (e.g. application servers, OS, RDBMS, Web servers and more) • Enhanced Web console maps business processes to application tiers, promoting more efficient use of resources • Advanced transaction recorder makes it easy to measure the end user experience by tracing transactions and response times from geographically dispersed locations • Pre-configured, application-specific monitoring rules cut down on setup time and reduce the need for expensive IT expertise • Out-of-the-box Web reports allow application owners to report on service levels, application events and more to improve accountability • New registry groupings enhance management of larger environments
19	Freshwater Software/Mercury Interactive	SiteScope & SiteSeer	Agentless monitoring; integrated with Topaz, LoadRunner and ProTune, it allows users to define monitor sets that can be reused across these products.
20	Agiliti	No tool	A services company. Agiliti offers a flexible suite of services that allow customers to monitor applications, servers and network infrastructure. Agiliti can also manage wide-area networks and provide a single point of contact for the management of carrier and hardware. Whether the technology is located at customer facilities or hosted at Agiliti, this suite of services allows IT managers to become proactive and address issues before critical systems are impacted. Agiliti's Monitoring and Network Management service integrates proven technologies from Hewlett-Packard, Concord, Cisco, NetIQ WebTrends, and Empirix. The service is provided through a secure connection to the customers environment and reporting information is accessed through a secure Web-based interface from Agiliti – the Customer Interaction Center (CIC).
21a	Operative Software Products	Application Vantage; ClientVantage	Application monitoring, hardware monitoring, network monitoring

S. NO.	VENDOR	TOOL	SUMMARY
21b	Compuware Corporation	Application Vantage	• Database administrators can see individual SQL statements, their duration and how long the server is spending on these commands. • Webmasters can see how HTTP requests for page components including XML, Active Server Pages, CGI scripts, Java scripts, images and animations are affecting front-end performance. • Network engineers can quickly see how the network's quality of service is affecting application response
22	Argent Software, Inc.	The Argent Guardian: Monitoring and alerting	The Argent Guardian is an extremely scalable monitoring and alerting solution. The Argent Guardian proactively manages an enterprise, identifying and correcting issues before the end users even know problems exist. Using a unique agent-optional architecture, the Argent Guardian has the ability to monitor the servers with or without installing agents, giving the power and flexibility to define the monitoring architecture to one's exact needs. When issues arise, the Argent Guardian offers a variety of alerting mechanisms, such as alphanumeric paging, e-mail, and SMS instant messages, as well as providing corrective measures such as restarting stopped or hung services.
23	Wily Technology	IntroScope	Monitors (both proactively and reactively) any production Java application 24x7, without degrading its performance. One can open up the Java application black box using Introscope to get component-level views of entire Java application environment. IntroScope's Blame™ technology isolates bottlenecks in whole application all the way down to individual Servlets, EJBs, Classes, Methods and more. Agent-based.
24	Mercury Interactive	Topaz	• Business Process Monitor Monitors automated business processes • Client Monitor Gathers performance data directly from client devices • Global Monitor Monitors end-user experience both inside and outside the firewall • Real User Monitor Measures the online experience of every user, from every location • SiteScope (see 19) Agentless monitoring of distributed systems
25	HP	OpenView	Performance agent installed on each system.

S. NO.	VENDOR	TOOL	SUMMARY
26	Micromuse	NetCool	NetCool Probes & Monitors™ collect fault and availability information from APIs, databases, devices, log files, and other utilities, and push them into the Netcool ObjectServer for filtering and viewing. With the Netcool/OMNIbus v3.4 application, fault and availability information from more than 1,000 alarm source types can be collected. Using Generic Probes, custom rules files, and Universal Probes – such as SNMP traps and TL1 – Netcool can collect data from virtually any networked environment.
27	LeadByte Corporation	NetworkSmart	NetworkSmart monitors code performance real time and allows developers to: • Measure performance and predict end user response time when writing code, rather than after the code is compiled. • Analyze application files in detail to determine how they are interacting with the client. • Profile database to determine which database calls are causing the longest delay. • View enhanced frame details to see all of the network traffic generated by your application. • Determine network performance using Local Host, no network is required!
28	Marcel Mendes Software	PMON Process Monitor	Process Monitor (PMON) can execute any of the following when a monitored program stops running: register an event, e-mail someone, shutdown server, execute a program or stop a program. Up to 50 programs can be monitored at the same time.
29	Beson Data	WinFeedback 4.0	WinFeedback 4.0 is a Windows Scripting extension for testing, monitoring and automation purposes. It includes a Developer application that makes it easy to create and test new scripts. The recorder captures user input by keyboard and mouse and generates the corresponding script statements. The playback engine identifies the correct target object in Windows graphical interface and correlates the playback sequence with the needed processing time. An individual-owned/maintained and distributed code.
30	Nifty Tools	WatchDog System	Monitor all types of computer systems including IP based systems, IP services, XP/2000/NT/9x systems, file servers, application servers, electronic mail servers, databases, modem/remote access systems, routers and other hardware devices for intermittent and consistent failures.
31	OneStat.com	OneStat suite	Web site statistics, Web analytics, Web metrics, site analysis, Web site traffic analytics, log file analysis tools, traffic analyzer. A non-US company.

S. NO.	VENDOR	TOOL	SUMMARY
32	Silas Reveille		Caters to healthcare industry primarily. Agentless architecture. Leverages existing infrastructure investments, easily monitoring diverse applications running on multiple platforms that likely have been installed at various stages of a company's growth. And it's able to do this without software agents that are required with other enterprise software. Eliminate the time, effort and expense of programming, installing and managing intrusive software probes. Silas Reveille monitors – from a central location – critical business applications as if the customers were using them, to see what's happening out there, where the technology service meets the user.
33	NetIQ	AppManager Suite	Web analytics, traffic analysis, Web site statistics.
34	Team Quest	TeamQuest Model	Capacity planning software for what-if analysis, performance prediction, and server consolidation – retrieve data automatically from HP OpenView Performance to predict capacity requirements using TeamQuest Model's sister product, TeamQuest Model for HP OpenView Performance. It has all the same capabilities as TeamQuest Model, so it's a great tool for server consolidation and capacity planning projects. TeamQuest Lite is a self-contained application with data collection agents, a miniaturized data store, and an intelligent client interface. Continuous connection between multiple collection agents and the client is not required for the operation of either agent or client. Data agents gather statistics on CPU utilization, I/O usage, memory usage, disk space utilization, network file system use, and TCP/IP activity, all unobtrusively in the background. TeamQuest Lite has two main components: 1. a rule-based visual interface and 2. a small data collection agent. Uses built-in rule-based intelligence to rate system performance

S. NO.	VENDOR	TOOL	SUMMARY
35	SL-GMS Solutions	Enterprise RTView Infrastructure Mon	• Monitor business information and enterprise infrastructure in real time • Use included monitoring displays or build custom displays and dashboards, without programming • View real-time data from multiple sources: XML, SQL, TIBCO™, JMS, SNMP, and custom data • Flexible deployment options allow you to embed displays in existing Java™ applications and enterprise portals, or distribute throughout the enterprise as compact Java™ applets • Flexible configuration allows layout to be personalized to individuals or roles in the organization • Archive historical data for trend analysis • Perform calculations on data (e.g.: sum of all values in a table column) privacy-unfriendly techniques. Components • Enterprise RTView is delivered with the Display Builder, the Display Viewer, the Historian and the XML Server—all pre-configured to access a variety of integrated data sources. • With the Historian archived data can be used for comparison to live data for trend analysis, baseline plotting, or evaluation of other metrics. • The XML Server makes it possible to eliminate client access to data sources and provides maximum scalability.
36	NetScout Systems	nGenius	nGenius Performance Manager in combination with nGenius Probes, provides the most comprehensive identification of applications flowing across your network. It recognizes, discovers and monitors virtually any application, including well-known, custom, Web-based, and even complex ones such as SAPr3 and Citrix, delivering the most complete picture of the network and the applications that traverse it.
37	Gomez	Gómez Performance Network (GPN)	Employ an application monitoring strategy known as active probing.
38	Keynote		Employ active probing
39	Netmechanics		Employ active probing. It is a Keynote service.

S. NO.	VENDOR	TOOL	SUMMARY
40	Software Research Inc.	eValid InBrowser Website Quality Solution	Employ active probing
41	Candle Corporation	PathWAI Products	Uses the Web page instrumentation strategy for Web app monitoring
42	Clickstream		Clickstream data collection uses unique client-side scripting and cookies to record page displays and page and frame timings. This means that even if a page is served from a browser's cache, the event will be timed and recorded. It also means that the communication of the tracking data to the server-side data collector is seamless and automatic, with no need for hidden messaging or other privacy-unfriendly techniques. Clickstream requires no client-side software to be installed and, because all data gathering is managed automatically by a single server cartridge, it requires no changes to the target Web site's pages and it can handle any type of Web content whether static or dynamically served. The Clickstream data collector stores the client-side tracking data, which is then passed to your analytical or reporting application(s) for analysis.
43	ClickTracks	ClickTracks Analyzer, Pro …	ClickTracks Pro: • Uses log files • Visual interface • Track Google, Overture, and custom campaigns • Track Revenue, sales, and ROI • Perfect for medium/large Web sites • Perfect for sites using Miva Merchant
44	Quest Software (BB4 Technologies)	Big Brother	Big Brother is designed to let anyone see how their network is doing in near real-time, from any Web browser, anywhere. Big Brother uses a client-server architecture combined with methods which both push and pull data. Network testing is done by polling all monitored services from a single machine, and reporting these results to a central location (the BBDISPLAY). If local system information is required, one can install a BB client on the local machine, which will send CPU, process, disk space, and log file status reports in periodically. Each report is time stamped with an expiration date.

S. NO.	VENDOR	TOOL	SUMMARY
45	Sun Microsystems	Java Management Extensions (JMX)	Monitoring Framework. The JMX technology is an open technology for management and monitoring that can be deployed wherever management and monitoring are needed. By design, this standard is suitable for adapting legacy systems, implementing new management and monitoring solutions and plugging into those of the future.
46	www.mrtg.org	Multi Router Traffic Grapher (MRTG)	Open Source; Network Monitoring A tool to monitor the traffic load on network-links. MRTG generates HTML pages containing graphical images which provide a LIVE visual representation of this traffic. See http://www.ee.ethz.ch/stats/mrtg/ for an example. MRTG is based on Perl and C and works under UNIX and Windows NT.
47	Covalent Technologies	Covalent Application Manager	Application Monitoring

5.2 APPENDIX B – A SURVEY OF COTS DASHBOARDS

S. NO.	VENDOR	TOOL	SUMMARY
1	Infommersion	Xcelsius	Takes Data from Excel spreadsheets. Demos in Flash
2	iDashboards	iDashboards	Rich collection of data-viewing capabilities, Tables, Reports, Library of Charts, 3-D Views, Geographic Maps, Animated Speedometer, and customized data displays. It also supports Microsoft Excel integration with data and chart Export to Excel. iDashboards seamlessly integrates with any relational database supporting ODBC connectivity. It is vendor independent, supporting databases by IBM, Oracle, Microsoft and open source relational databases such as MySQL. It can also be customized to integrate with multi-dimensional OLAP data cubes and legacy data systems. iDashboards works within any of the leading application server environments
3	SL Business Dashboards	Enterprise RTView	• Build Custom Dashboards Rapid deployment, no programming • Use included monitoring displays or build custom displays and dashboards, without programming • View real-time data from multiple sources: XML, SQL, TIBCO™, JMS, SNMP, and custom data • Flexible deployment options allow you to embed displays in existing Java™ applications and enterprise portals, or distribute throughout the enterprise as compact Java™ applets • Flexible configuration allows layout to be personalized to individuals or roles in the organization • Archive historical data for trend analysis • Perform calculations on data (e.g.: sum of all values in a table column)

S. NO.	VENDOR	TOOL	SUMMARY
4	Corda	PopChart Corda's interactive data visualization products (server products, Design Products, Desktop product)	• Interface directly with all existing databases • Run on any platform and support all server environments • Enterprise-wide real-time sharing • Integrate easily with any Web application server • Requires only a Web browser to use • Full compliance with Federal accessibility and 508 requirements • Extensive library of graphic styles and templates for high quality • imaging in small files • Customizable to suit business needs if templates are not sufficient • Demo, trial, code examples
5	Distributive Software	DataDrill Components (DataCenter, ReportCenter and Extractor Kit)	The ReportCenter uses the most current measurement process terminology – the terminology shared by SEI's CMMI, Practical Software and Systems Measurement and the ISO/IEC 15939 software measurement standard. Demo, signup for a walkthrough
6	Open Door Technology: Navision Software	Digital Dashboard Resource Kit 2.2	http://www.opendoor.ab.ca/ODTArticleDashboard.htm Demo
7	Wily Technology	Introscope® WebView	One can choose from a drop-down list of customized dashboards to rapidly monitor the systems and components. Additionally, because application is monitored through a Web browser interface, one can instantly create multiple views of metrics when using the Explorer tree. Introscope WebView is bundled with Introscope 4.2.
8	Team Quest	TeamQuest View	Scale from one to hundreds of systems without losing its characteristic ease-of-use. View reports from heterogeneous systems in a single consolidated report. Monitor systems and applications with the Java interface from the desktop of your choice.
9	Crystal Decisions		

5.3 APPENDIX C – A SURVEY OF COTS INTELLIGENT TOOLS

S. NO.	VENDOR	PRODUCT	FEATURES
1	Computer Associates	Neugents®	CA's U.S. patented Neugents® technology provides predictive intelligence for virtually all aspects of today's collaborative eBusiness infrastructure. CA's Neugents technology is unique in its ability to identify problems that are likely to happen so eBusinesses can take action before losing revenue. When applied to network and system performance metrics, this method gives administrators the ability to reallocate servers, reroute traffic and plan upgrades.
2	Computer Associates / Step	Jasmine ii	Jasmine ii provides a complete solution for building, testing, and deploying intelligent eBusiness applications. It integrates an application server, object transaction monitor, enterprise applications, dynamic personalization, and portal technology into one platform, giving developer's time to focus on delivering compelling eBusiness systems rather than integrating today's diverse technologies. Dynamic Personalization With Intelligence is made possible using Nuegents™, CA's innovative approach to creating eBusiness value. Neugents predict customer needs and preferences, forecast business outcomes across any business process, and drive compelling, personalized portals that are dynamically updated in real-time to fulfill customer needs.
3	Deepmetrix	Deepmetrix Professional: Visitor Intelligence	DeepMetrix® Professional provides the in-depth Visitor Intelligence one needs to Understand, Predict, Respond with confidence to how people use a Web site. Begin with knowing who the visitors are, where they come from, and what content and products they consume. The DeepMetrix Service offers true Visitor Intelligence. It tracks real people, not just hits, views and impressions. It is able to accurately report New Visitors, Repeat Visitors, Visit Count and Distinct Visitors. Most other competing solutions are unable to distinguish one visitor from another and can only offer general trends based on aggregate values.

5.4 APPENDIX D – REFERENCES

5.4.1 Technical Papers

1. Ludmila Cherkasova, Yun Fu, Wenting Tang and Amin Vahdat, "Measuring and Characterizing End-to-End Internet Service Performance," ACM Transactions on Internet Technology, Vol. 3, No. 4, November 2003, Pp. 347-391.
2. Jeffrey C. Mogul, "Clarifying the Fundamentals of HTTP," WWW2002, May 7-11, 2002, Honolulu, Hawaii, USA, ACM 1-58113-449-5/02/0005, Pp. 25-36.
3. Edith Cohen, Balachander Krishnamurthy and Jennifer Rexford, "Improving End-to-End Performance of the Web Using Server Volumes and Proxy Filters," SIGCOMM'98 Vancouver, B.C., ©1998 ACM 1-59113-003-1/99/0008, Pp. 241-253.
4. Terence Kelly and Jeffrey Mogul, "Aliasing on the World Wide Web: Prevalence and Performance Implications," WWW2002, May 7-11, 2002, Honolulu, Hawaii, USA, ACM 1-58113-449-5/02/005, Pp. 281-292.
5. Oscar Ardaiz, Felix Freitag and Leandro Navarro, "Estimating the Service Time of Web Clients using Server Logs," SIGCOMM – Latin America & Caribbean 4/01 San Jose, Costa Rica, ©2001 ACM ISBN 1-58113-354-5/01/0004, Pp. 108-123.
6. Paul Barford and Mark Crovella, "Critical Path Analysis of TCP Transactions," SIGCOMM – Latin America & Caribbean 4/01 San Jose, Costa Rica, ©2001 ACM ISBN 1-58113-354-5/01/0004, Pp. 80-102.
7. Balachander Krishnamurthy, Craig E. Wills, Yin Zhang, Kashi Vishwanath, "Design, Implementation, and Evaluation of a Client Characterization Driven Web Server," WWW2003, May 20-24, 2003, Budapest, Hungary, ACM 1-58113-680-3/03/005, Pp. 138-147.
8. Joseph L. Hellerstein, Mark M. Maccabee, W. Nathaniel Mills III, and John J. Turek, "ETE: A Customizable Approach to Measuring End-to-End Response Times and Their Components in Distributed Systems," 1063-6927/99 ©1999 IEEE, Pp. 152-162.
9. Gerald Quirchmayr and Jill Slay, "Identifying Emerging Web-Communication Patterns Between Government and Citizens," Proceedings of the 12th International Workshop on Database and Expert Systems Applications (DEXA'01) 1529-4188/01 ©2001 IEEE, 4 Pages.
10. V. S. Alagar and O. Ormandjieva, "Reliability Assessment of WEB Applications," Proceedings of the 26th Annual International Computer Software and Applications Conference (COMPSAC'02) 0730-3157/02 ©2002 IEEE, Pp. 1-8.
11. Tutorial on Internet Monitoring and PingER at SLAC:
12. http://www.slac.stanford.edu/comp/net/wan-mon/tutorial.html
13. Service Availability: An Approach to Measurement by David M. Fishman:
14. http://www.nextslm.org/fishman.html

5.4.2 Usability-Specific Papers

1. Marc Chrusch, "Seven Great Myths of Usability," Interactions, September & October 2000, Pp. 13-16.
2. Sharon J. Laskowski, James A. Landay and Mike Lister, "Automatic Capture, Representation, and Analysis of User Behavior," CHI 2002, April 20-25, 2002, Minneapolis, Minnesota, USA. ACM 1-58113-454-1/02/0004. Pp. 922-923.
3. Marilyn Hughes Blackmon, Peter G. Polson, Muneo Kitajima, Clayton Lewis, "Cognitive Walkthrough for the Web," CHI 2002, April 20-25, 2002, Minneapolis, Minnesota, USA. ACM 1-58113-453-3/02/0004. Volume No. 4, Issue No. 1, Pp. 463-470.
4. David M. Hilbert and David F. Redmiles, "Extracting Usability Information from user Interface Events," ACM Computing Surveys, Vol. 32, No. 4, December 2000, Pp. 384-421.
5. Sougata Mukherjea, "Organizing Topic-Specific Web Information," Icarian Inc., 555 North Mathilda Avenue, Sunnyvale, CA 94086, USA, e-mail: sougata@icarian.com, Pp. 133-141.
6. Fabio Paterno and Laila Paganelli, "Remote Automatic Evaluation of Web Sites Based on Task Models and Browser Monitoring, " Short Talks, CHI 2001, 31 March – 5 April, Pp. 283-284.
7. H. Rex Hartson, Jose C. Castillo, John Kelso and Wayne C. Neale, "Remote Evaluation: The Network as an Extension of the Usability Laboratory," CHI 96, April 13-18, 1996, Vancouver, BC Canada, ©1996 ACM 0-89791-777-4/96/04, Pp. 228-235.
8. R. Stanley Dicks, "Mis-Usability: On the Uses and Misuses of Usability Testing," SIGDOC'02, October 20-23, 2002, Toronto, Ontarion, Canada, ©2002 ACM 1-58113-543-2/02/0010, Pp. 26-30.
9. Albert L. Lederer, Donna J. Maupin, Mark P. Sena, Youlong Zhuang, "The Role of Ease of Use, Usefulness and Attitude in the Prediction of World Wide Web Usage," CPR 98, Boston, MA, USA, ©1998 ACM 0-89791-959-9/98/3, Pp. 195-204.
10. Marilyn Hughes Blackmon, Muneo Kitajima and Peter G. Polson, "Repairing Usability Problems Identified by the Cognitive Walkthrough for the Web," CHI 2003, April 5-10, 2003, Ft. Lauderdale, Florida, USA, c 2003 ACM 1-58113-630-7/03/004, Vol. No. 5, Issue No. 1, Pp. 497-504.
11. John Cugini and Sharon Laskowski, "Design of a File Format for Logging Website Interaction," National Institute of Standards and Technology (NIST). NIST Special Publication 500-248, April 2001.
12. Web Metrics Testbed, National Institute of Standards and Technology (NIST).
13. David M. Hilbert and David F. Redmiles, "An Approach to Large-Scale Collection of Application Usage Data Over the Internet", 0-8186-8368-6/98 ©1998 IEEE, Pp. 136-145.
14. David M. Hilbert and David F. Redmiles, "Separating the Wheat from the Chaff in Internet-Mediated User Feedback Expectation-Driven Event Monitoring," SIGGROUP Bulletin, April 1999/Vol 20, No. 1, Pp. 35-40.
15. David M. Hilbert and David F. Redmiles, "Agents for Collecting Application Usage Data Over the Internet," Autonomous Agents, 98 Minneapolis, MN, USA, ©1998 0-89791-983-1/98/5, Pp. 149-156.

16. John D. Gould and Clayton Lewis, "Designing for Usability: Key Principles and What Designers Think," ©1985 ACM, March 1985, Volume 28, Number 3, 0001-0782/85/0300-0300, Pp. 300-311.
17. Cynthia M. Calongne, "Designing for Web Site Usability," JCSC 16, 3 (March 2001), ©2001 by the Consortium for Computing in Small Colleges, Pp. 39-45.
18. Jan Bosch and Natalia Juristo, "Designing Software Architectures for Usability," 0-7695-1877-X/03, ©2003 IEEE, Pp. 757-758.
19. Jeffrey Brainerd and Barry Becker, "Case Study: E-Commerce Clickstream Visualization," Proceedings of the IEEE Symposium on Information Visualization 2001 (INFOVIS'01), 1522-4048/01, ©2001 IEEE, 4 Pages.
20. Shawren Singh and Paula Kotze, "Towards a Framework for E-Commerce Usability", Proceedings of SAICSIT 2002, Pp. 2-10.
21. Ed H. Chi, "Improving Web Usability Through Visualization," IEEE Internet Computing, March-April 2002, 1089-7801/02, ©2002 IEEE, Pp. 64-71.
22. Nan Niu, Eleni Stroulia and Mohammad El-Ramly, "Understanding Web Usage for Dynamic Web-Site Adaptation: A case Study", IEEE Computer Society, Proceedings of the Fourth International Workshop on Web Site Evolution (WSE'02), 0-7695-1804-4/02, ©2002 IEEE, 10 Pages.
23. Danita Hartley, Neville Churcher and Greg Albertson, "Visual Worlds for Web Site Visualization", Proceedings of the Seventh Asia-pacific Software Engineering Conference (APSEC'00), 1530-1362/00, ©2000 IEEE, Pp. 448-455.

5.4.3 Ontology-Related Publications

1. Felipe Castel, "Viewpoint: Ontological Computing," Communications of the ACM, February 2002, Vol. 45, No. 2, pp. 29-30.
2. Michael Gruninger and Jintae Lee, "Ontology Applications and Design," Communications of the ACM, February 2002, Vol. 45, No. 2, pp. 39-41.
3. Clyde W. Holsapple and K. D. Joshi, "A Collaborative Approach to Ontology Design," ," Communications of the ACM, February 2002, Vol. 45, No. 2, pp. 42-47.
4. Nicola Guarino and Christopher Welty, "Evaluating Ontological Decisions with ONTOCLEAN," Communications of the ACM, February 2002, Vol. 45, No. 2, pp. 61-65.
5. Henry Kim, "Predicting Ontologies for the Semantic Web will Evolve," Communications of the ACM, February 2002, Vol. 45, No. 2, pp. 48-54.
6. John O. Everett, et al, "Making Ontologies Work for Resolving Redundancies across Documents," Communications of the ACM, February 2002, Vol. 45, No. 2, pp. 55-60.
7. Barry Smith and Christopher Welty, "Ontology: Towards a New Synthesis," FOIS'01, October 17-19, 2001, Ogunquit, Maine, USA, © 2001 ACM 1-58113-377-4/01/0010, pp. iii-ix.
8. Ian Niles and Adam Pease, "Towards a Standard Upper Ontology", FOIS'01, October 17-19, 2001, Ogunquit, Maine, USA, © 2001 ACM 1-58113-377-4/01/0010, pp. 2-9.

9. David D. McDonald, "The View from the Trenches: Issues in the Ontology of Restricted Domains," FOIS'01, October 17-19, 2001, Ogunquit, Maine, USA, © 2001 ACM 1-58113-377-4/01/0010, pp. 22-33.
10. Erwan Breton and Jean Bezivin, "Towards an Understanding of Model Executability," FOIS'01, October 17-19, 2001, Ogunquit, Maine, USA, © 2001 ACM 1-58113-377-4/01/0010, pp. 70-80.
11. Valentina A. M. Tamma and Trevor Bench-Capon, "An Enriched Knowledge Model for Formal Ontological Analysis," FOIS'01, October 17-19, 2001, Ogunquit, Maine, USA, © 2001 ACM 1-58113-377-4/01/0010, pp. 81-92.
12. Johanna Seibt, "Formal Process Ontology," FOIS'01, October 17-19, 2001, Ogunquit, Maine, USA, © 2001 ACM 1-58113-377-4/01/0010, pp. 333-345.
13. Maureen Donnelly, "Introducing Granularity-Dependent Quantitative Distance and Diameter Measures in Common-Sense Reasoning Contexts," FOIS'01, October 17-19, 2001, Ogunquit, Maine, USA, © 2001 ACM 1-58113-377-4/01/0010, pp. 321-332.
14. Paola Velardi, Paolo Fabriani and Michele Missikoff, "Using Text Processing Techniques to Automatically Enrich a Domain Ontology," FOIS'01, October 17-19, 2001, Ogunquit, Maine, USA, © 2001 ACM 1-58113-377-4/01/0010, pp. 270-284.
15. Massimiliano Carrara and Pierdaniele Giaretta, "Identity Criteria and Sortal Concepts," FOIS'01, October 17-19, 2001, Ogunquit, Maine, USA, © 2001 ACM 1-58113-377-4/01/0010, pp. 234-243.
16. Aaron N. Kaplan, "Towards a Consistent Logical Framework for Ontological Analysis," FOIS'01, October 17-19, 2001, Ogunquit, Maine, USA, © 2001 ACM 1-58113-377-4/01/0010, pp 244-255.
17. Gloria L. Zuniga, "Ontology: Its Transformation From Philosophy to Information Systems," FOIS'01, October 17-19, 2001, Ogunquit, Maine, USA, © 2001 ACM 1-581 13-377-4/01/0010, pp. 187-197.
18. John Howse, Fernando Molina, John Taylor and Sun-Joo Shin, "Type-syntax and Token-syntax in Diagrammatic Systems," FOIS'01, October 17-19, 2001, Ogunquit, Maine, USA, © 2001 ACM 1-58113-377-4/01/0010, pp. 174-185.
19. Lawrence W. Barsalou, "Plenary Talk: The Human Conceptual System," FOIS'01, October 17-19, 2001, Ogunquit, Maine, USA, © 2001 ACM 1-58113-377-4/01/0010, pp. 186.
20. Leo Obrst, Robert E. Wray and Howard Liu, "Ontological Engineering for B2B E-Commerce," FOIS'01, October 17-19, 2001, Ogunquit, Maine, USA, © 2001 ACM 1-58113-377-4/01/0010, pp. 117-126.
21. Brandon Bennett, "Space, Time, Matter and Things," FOIS'01, October 17-19, 2001, Ogunquit, Maine, USA, © 2001 ACM 1-581 13-377-4/01/0010, pp. 105-116.
22. Cleo Condoravdi, Dick Crouch and Martin van den Berg, "Preventing Existence," FOIS'01, October 17-19, 2001, Ogunquit, Maine, USA, © 2001 ACM 1-58113-377-4/01/0010, pp. 162-173.
23. Ariel Fuxman, Paolo Giorgini, Manuel Kolp and John Mylopoulos, "Information Systems as Social Structures," FOIS'01, October 17-19, 2001, Ogunquit, Maine, USA, © 2001 ACM 1-58113-377-4/01/0010, pp. 10-21.

24. Ann Chadwick-Dias, Michelle McNulty and Tom Tullis, “Web Usability and Age: How Design Changes Can Improve Performance,” CUU’03, Nov 10-11, 2003, Vancouver, British Columbia, Canada, © 2003 ACM 1-58113-701-X/03/0011, pp. 30-37.

5.4.4 Neural Network Agents-Specific Publications

1. Quirolgico, Stephen, Communicating Neural Networks in a Multi-Agent System, Ph.D. Dissertation, Department of Information Systems, University of Maryland, Baltimore County, 2000:
2. http://www.research.umbc.edu/~squiro1/thesis/
3. Yong S. Choi, "Neural Net Agent for Information Retrieval in Distributed Web Databases", Ph.D. Thesis, Department of Computer Science, Seoul National University, February 2000
4. Choi, Yoo, Multi-agent Web information retrieval: neural network based approach, Advances in Intelligent Data Analysis. Third International Symposium, IDA-99. Proceedings (Lecture Notes in Computer Science Vol.1642). Springer-Verlag, Berlin, Germany; 1999; xii+538 pp. p.499-511

5.4.5 White Papers on COTS Application Monitoring Tools

1. Dodging the Bullets: How to Avoid Costly Failures with Application Performance Monitoring: http://www.segue.com/html/s_solutions/pdf/whitepapers/Dodging_the_Bullets.pdf
2. Agentless Monitoring: Ready for Prime Time: A White Paper Prepared for Mercury Interactive, May 2003: http://www.knowledgestorm.com/collateral/WTP/14369_10992_77414_Final_EMA_Mercury-_Agentless_WP.pdf
3. A collection of white papers white Papers on Mercury Interactive's SiteScope: http://www.mercuryinteractive.com/products/sitescope/papers.html
4. Using Continuous Service Assurance to Deliver Best Value in IT: http://www.chevin.com/products/pdfs/assurance_whitepaper.pdf
5. A collection of white papers by RTTS: http://www.rttsWeb.com/news/whitepaper/index.cfm
6. Application Performance Monitoring: The Non-Intrusive Approach using iWatch – A whitepaper by Exact Solutions, Inc.: http://www.exact-solutions.com/whitepaper.pdf
7. CenterwiseTM Suite: Product Discussion: http://www.centrisoft.com/pdf-general/Centrisoft_Whitepaper.pdf
8. A collection of white papers by Packeteer: http://www.asl-packeteer.co.uk/content/whitepapers.htm
9. A collection of white papers on NetTracker: http://www.sane.com/products/NetTracker/whitepapers.html
10. A collection of white papers on NetVigil: http://www.fidelia.com/products/whitepapers.php
11. Bay Mountain’s white papers on IT Appliance: http://www.itappliance.com/whitePaper
12. A collection of white papers by Quest Software: DLT Solutions: http://www.dlt.com/quest/resources-whitepapers.asp
13. A collection of white papers by Argent Software: http://argent.com/support/whitepapers/whitepapers.html

14. Red Hat Network: Monitoring Module Overview: http://www.redhat.com/whitepapers/rhn/RHN_Monit_overview_1-1.sdw.pdf
15. A Comprehensive Approach to Monitoring and Improving Web Application Health: http://www.parasoft.com/jsp/products/article.jsp?articleId=1131
16. End-to-End Distributed Application Monitoring using the "Distributed Monitoring Framework" by Brian L. Tierney, Lawrence Berkeley National Laboratory: http://e2epi.Internet2.edu/WK01/wk01_presentations/tierney-DMF.pdf
17. PATROL Express: A Technical Overview: http://documents.bmc.com/products/documents/91/22/29122/29122/Output/090f44b1801f3d4a.htm
18. A collection of white papers by Computer Associates on Unicenter®: http://www3.ca.com/Files/Brochures/unicenter_brand.pdf http://www3.ca.com/Solutions/CollateralList.asp?CCT=19505&ID=315
19. Informed Management Through High-Quality Data Sources by Russ Currie: http://www.nwfusion.com/whitepapers/netscout/
20. a. A collection of white papers by Quest Software on Foglight®: http://www.quest.com/content/list.asp?ContentTypeID=1&GroupBy=CategoryName&Format=table&nav=/foglight/navigation.xml&prod=7,79,74,8,77,72,76,9,78,75,66,73,84&productfamily=app+monitoring
21. b. Lowering the Total Cost of Monitoring Technical Infrastructures Using Foglight® and Big BrotherTM: http://www.quest.com/whitepapers/Fog_BBwp.pdf
22. A collection of white papers by Gomez.com: http://www.gomez.com/main.aspx?m=4&s=3&c=16
23. netScout.com's nGenius: Quick Reference Guide: http://www.netscout.com/files/Quick Reference Guide_0903.pdf
24. White paper on Wily Technology's Introscope® 4.2: http://www.wilytech.com/pdf/product/Wily_Introscope4.2.pdf
25. The Gomex Universal Transaction Agent: A More Consistent and Granular Measurement of Internet Performance Across the First-, Middle- and Last-Mile by Paul aLeroux:
26. A list of white papers at: http://www.gomez.com/main.aspx?m=4&s=3
27. Corey Ferengul, An Agentless World, Service Management Strategies, http://www.silasreveille.com/data/UploadFiles/Whitepapers/Whitepaper8.pdf

5.4.6 White Papers on Page Tagging and Log Analysis

1. NetTracker Log File Analysis vs. Page Tagging: A Guide for Comparing Web Analytics Methodologies: http://www.sane.com/products/NetTracker/pagetag.pdf
2. "Page Embedding – An Accurate Technology" by Seth Hamilton, Deepmetrix Corporation: http://www.deepmetrix.com/images/white_papers/deepmetrix_page_embedding.pdf

5.5 APPENDIX E – GLOSSARY

ACTIVE AGENTS – Programs that simulate user actions and check for expected results.

CLIENT MONITORS – Programs that run on the client's browser to monitor usage.

DIGITAL DASHBOARDS – A typically colored graphical presentation of a project's status or a portfolio's status by project resembling a vehicle's dashboard. Typically, red is used to flag urgent problems, yellow to flag impending problems, and green to signal on projects on track. A digital dashboard consists of Web Parts: reusable components that can contain any kind of Web-based information.

EMBEDDED HOOKS – Modifications build directly into programs to log specifically identified information.

KNOWLEDGE ENGINEERING – Knowledge engineering is the technique applied by knowledge engineers to build intelligent systems. Examples of knowledge engineering include expert systems, knowledge based systems, knowledge based decision support systems, expert database systems, and others. There are two main views to knowledge engineering. The traditional view is known as "transfer view". In this view, the assumption is to apply conventional knowledge engineering techniques to transfer human knowledge into artificial intelligent systems. The alternative view is known as the "modeling view". In this view, the knowledge engineer attempts to model the knowledge and problem solving techniques of the domain expert into the artificial intelligent system.

LOG FILE ANALYSIS vs. PAGE TAGGING – A typical log file analysis methodology consists of parsing and cleansing the log files which are created when a customer requests any page from a Web server or e-commerce application server. As an alternative to log file analysis, the page-tagging methodology consists of inserting a script tag on each page of a site that is to be tracked. This methodology keeps track of how many times each tag is sent or script is executed. Unlike log file analysis, which relies on data collected and stored on the Web servers, page tagging collects data that

must be received back from the client's Web browser. Page-tagging solutions rely on JavaScript, and in order to be able to track return visitors one has to rely on third-party cookies from the third party clients.

METRICS – A standard of measurement, i.e., the "units" used to measure a phenomena or system parameter (inch, ampere, ft/lb, etc).

MONITORING – Monitoring is the process by which we identify and prevent problems with deployed client/server or Web-based systems to ensure their continued functionality and performance. Monitoring involves using a tool that periodically checks a Web site's overall health and sends alerts when it detects problems.

NEURAL NETWORKS – In information technology, a neural network is a system of programs and data structures that approximates the operation of the human brain. A neural network usually involves a large number of processors operating in parallel, each with its own small sphere of knowledge and access to data in its local memory. Typically, a neural network is initially "trained" or fed large amounts of data and rules about data relationships (for example, "A grandfather is older than a person's father"). A program can then tell the network how to behave in response to an external stimulus (for example, to input from a computer user who is interacting with the network) or can initiate activity on its own (within the limits of its access to the external world).

ONTOLOGY – An explicit formal specification of how to represent the objects, concepts, and other entities that are assumed to exist in some area of interest and the relationships that hold among them.

PACKET – A message transmitted over a network. Large chunks of information are broken up into packets before they are sent across the Internet.

PARAMETER – A variable whose value changes with the circumstance of its context.

PASSIVE AGENTS – Programs that track network flow.

PING (Packet Internet Groper) – A utility to determine whether a specific IP address is accessible. It works by sending a packet to the specified address and waiting for a reply. PING is used primarily to troubleshoot Internet connections.

PREDICTIVE MODELING – In data mining, predictive modeling is a technique used to predict future behavior and anticipate the consequences of change. In predictive modeling, data is collected, a statistical model is formulated, predictions are made and the model is validated (or revised) as additional data becomes available. The model can be a simple linear equation or it can be a complex neural network mapped out by sophisticated software.

PREDICTIVE STATISTICAL MODELS – Predictive statistical model software analyzes data from millions of online sales to extract the profile of fraudulent transactions. Culling data from large, historical databases, the software develops a mathematical formula and applies it in real time to incoming transactions. Each transaction then receives a risk score based on its attributes.

PROBE – One or more packets of data that are sent out according to a protocol to probe a destination. The best-known probe is probably ping.

RMON (Remote Monitoring) – Where SNMP gathers network data from a single type of MIB, RMON 1 defines nine additional MIBs that provide a much richer set of data about network usage.

SNMP (Simple Network Management Protocol) – A set of protocols, first developed in the early 80s, for managing complex networks. SNMP works by sending messages called protocol data units (PDUs), to different parts of a network. SNMP-compliant devices called agents, store data about themselves in Management Information Bases (MIBs) and return this data to the SNMP requesters. SNMP 1 reports only whether a device is functioning properly.

TRANSACTION – A transaction is an action or series of actions that changes something significant about the state of an enterprise.

USABILITY – Usability relates to the impact a product has on its end-users. In general it refers to the efficiency with which a customer can do their tasks with the product, and their overall satisfaction with

that process. In Web design, this refers to the capability of a Web site to be used by everyone. Usability issues include interface and navigation design (can the user easily understand how to find their way around the site), content layout (small blocks of text that are not too wide are easier for reading on the Web), and accessibility and compatibility issues.

WEB/HTTP METRICS – There are various terms used to describe the science of recording and interpreting Website statistics. Web metrics, Web analytics, Web stats and site stats, to name a few. 'E-metrics' refers to analysis of electronic businesses.

WEB SERVER MONITORS – Programs that run on the Web server to monitor usage.