UNITED STATES PATENT AND TRADEMARK OFFICE

---

BEFORE THE PATENT TRIAL AND APPEAL BOARD

---

ServiceNow, Inc.
Petitioner

v.

Hewlett-Packard Company
Patent Owner

U.S. Patent No. 7,925,981
Filing Date: May 14, 2003
Issue Date: April 12, 2011

TITLE: SYSTEMS AND METHODS FOR MANAGING WEB SERVICES VIA A FRAMEWORK OF INTERFACES

**DECLARATION OF TAL LAVIAN, PH.D.**

*Inter Partes* Review No. 2015-__

## Table of Contents

Page

**Table of Contents**
(continued)

**Page**

**Table of Contents**
(continued)

**Page**

Declaration of Tal Lavian, Ph.D., in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,925,981

I, Tal Lavian, Ph.D., declare as follows:

1. I have personal knowledge of the facts stated in this declaration, and could and would testify to these facts under oath if called upon to do so.

2. I have been retained by counsel for ServiceNow, Inc. (Petitioner) in this case as an expert in the relevant art.

3. I have been asked to provide my opinions relating to claims 1, 22 and 23 ("Challenged Claims") of U.S. Patent No. 7,925,981 to M. Homayoun Pourheidari et al. ("the '981 patent"), which I understand is owned by Hewlett-Packard, Inc. ("Patent Owner" or "HP").

## I. BRIEF SUMMARY OF MY OPINIONS

4. Claims 1, 22 and 23 purport to disclose a system and computer program product for managing a web service. They do not describe anything that was new or non-obvious by the time the application for the '981 patent was filed in May 2003. As explained in detail in **Part VI** of this Declaration, the features described in these claims were disclosed in product manuals for a prior art product called "BEA WebLogic Collaborate," which were published almost two years before the filing date of the patent. Because each element of each challenged claim is disclosed or suggested by the prior art as described below, and

a person of ordinary skill in the art would have had ample motivation to combine, each challenged claim is obvious. The bases for my opinions are set forth below.

## II. INTRODUCTION AND QUALIFICATIONS

### A. Qualifications and Experience

5. I possess the knowledge, skills, experience, training and the education to form an expert opinion and testimony in this case. A detailed record of my professional qualifications, including a list of patents and academic and professional publications, is set forth in my curriculum vitae attached to this declaration as **Exhibit A.**

6. I have more than 25 years of experience in the networking, telecommunications, Internet, and software fields. I received a Ph.D. in Computer Science from the University of California at Berkeley in 2006 and obtained a Master's of Science ("M.Sc.") degree in Electrical Engineering from Tel Aviv University, Israel, in 1996. In 1987, I obtained a Bachelor of Science ("B.Sc.") in Mathematics and Computer Science, also from Tel Aviv University.

7. I am currently employed by the University of California at Berkeley and was appointed as a lecturer and Industry Fellow in the Center of Entrepreneurship and Technology ("CET") as part of UC Berkeley College of

Engineering. I have been with the University of California at Berkeley since 2000 where I served as Berkeley Industry Fellow, Lecturer, Visiting Scientist, Ph.D. Candidate, and Nortel's Scientist Liaison, where some positions and projects were done concurrently, others sequentially.

8. I have more than 25 years of experience as a scientist, educator and technologist, and much of my experience relates to computer networking technologies. For eleven years from 1996 to 2007, I worked for Bay Networks and Nortel Networks. Bay Networks was in the business of making and selling computer network hardware and software. Nortel Networks acquired Bay Networks in 1998, and I continued to work at Nortel after the acquisition. Throughout my tenure at Bay and Nortel, I held positions including Principal Scientist, Principal Architect, Principal Engineer, Senior Software Engineer, and led the development and research involving a number of networking technologies.

9. Prior to that, from 1994 to 1995, I worked as a software engineer and team leader for Aptel Communications, designing and developing mobile wireless devices and network software products. From 1990 to 1993, I worked as a software engineer and team leader at Scitex Ltd., where I developed system and network communications tools (mostly in C and C++).

10. I have extensive experience in the area of network communications and Internet technologies including design and implementation of computer-based systems for managing communications networks, including the ability to monitor and provision networks. While with Nortel Networks and Bay Networks (mentioned above) my work involved the research and development of these technologies. For example, I wrote software for Bay Networks and Nortel Networks Web based network management for Bay Networks switches. I developed Simple Network Management Protocol (SNMP) software for Bay Network switches and software interfaces for Bay Networks' Optivity Network Management System. I wrote software for Java based device management including software interface to the device management and network management for the Accelar routing switch family network management system.

11. I have extensive experience in network communications, including control and management of routing and switching architectures and protocols in layers 1-7 of the OSI model. Much of my work for Nortel Networks (mentioned above) involved the research and development of network communications technologies. For example, I wrote software for Bay Networks and Nortel Networks switches and routers, developed network technologies for the Accelar

8600 family of switches and routers, the OPTera 3500 SONET switches, the OPTera 5000 DWDM family, and the Alteon L4-7 switching product family. In my lab, I installed, configured, managed and tested many network communications equipment of competitors such as Cisco Systems, Juniper Networks, Extreme Networks, Lucent and Alcatel.

12. I am named as a co-inventor on more than 80 issued patents and I co-authored more than 25 scientific publications, journal articles, and peer-reviewed papers. Furthermore, I am a Senior Member of the Institute of Electrical and Electronics Engineers ("IEEE").

13. I currently serve as a Principal Scientist at my company Telecomm Net Consulting Inc., where I develop network communication technologies and provide research and consulting in advanced technologies, mainly in computer networking and Internet technologies. In addition, I serve as a Co- Founder and Chief Technology Officer (CTO) of VisuMenu, Inc., where I design and develop architecture of visual IVR technologies for smartphones and wireless mobile devices in the area of network communications. The system is based on cloud networking and cloud computing utilizing Amazon Web Services.

14. Additional details of my background are set forth in my curriculum vitae, attached as **Exhibit A** to this Declaration, which provides a more complete description of my educational background and work experience. I am being compensated for the time I have spent on this matter. My compensation does not depend in any way upon the outcome of this proceeding. I hold no interest in the Petitioner (ServiceNow, Inc.) or the patent owner (Hewlett-Packard Company).

**B. Materials Considered**

15. The analysis that I provide in this Declaration is based on my education and experience in the field of computer systems, as well as the documents I have considered including U.S. Patent No. 7,925,981 ("'981 patent") **[Ex. 1001]**, which states on its face that it issued from an application filed on May 14, 2003. I also reviewed U.S. Patent No. 7,945,860 ("'860 patent") **[Ex. 1003]**, which the '981 patent incorporates by reference and identifies as having related subject matter. ('981, 1:7-12.)

16. I reviewed various documents dated prior to May 2003 describing the state of the art at the time of the alleged invention of the '981 patent. As explained below, some of these documents are relied upon as actually disclosing the limitations of the '981 patent, while others are being relied upon primarily for

background purposes. The prior art documents that I rely upon in this Declaration as actually disclosing the limitations of the claims are:

| Exhibit No. | Description of Document |
|---|---|
| **1004** | Introducing BEA WebLogic Collaborate, BEA Systems, Inc., July 2001 ("Introducing Collaborate") |
| **1005** | Administering BEA WebLogic Collaborate, BEA Systems, Inc., July 2001 ("Administering Collaborate") |
| **1006** | Programming BEA WebLogic Collaborate Management Applications, BEA Systems, Inc., July 2001 ("Programming Collaborate") |
| **1008** | Web Publisher's Construction Kit with HTML 3.2, David Fox & Troy Downing, Waite Group, 1996, pp.480-544 ("Fox") |

This Declaration also cites the following additional prior art documents for purposes of describing the relevant technology, including the relevant state of the art at the time of the alleged invention of the '981 patent:

| Exhibit No. | Description of Document |
|---|---|
| **1007** | Java Web Services, David A. Chappell & Tyler Jewell, O'Reilly & Associates, March 2002, pp.1-12 ("Chappell") |
| **1009** | Applied SOAP: Implementing .NET XML Web Services, Kenn Scribner & Mark Stiver, Sams Publishing, 2001, pp.10-48 ("Scribner") |
| **1010** | XML in a Nutshell, Elliotte Rusty Harold et al., O'Reilly & Associates, 2001, pp.xi-xvi, 3-10 ("Harold") |
| **1011** | BEA Unveils Comprehensive Web Services Strategy and Support For Widest Range of Web Services Standards in the Industry, PR Newswire, Feb. 26, 2001 |

| Exhibit No. | Description of Document |
|---|---|
| **1012** | Microsoft Computer Dictionary, Fifth Edition, Microsoft Press, 2002, pp.279-80 |
| **1013** | BEA and Gauss Interprise Announce Strategic Relationship, Canadian Corporate Newswire, Aug. 27, 2001 |

## III. PERSON OF ORDINARY SKILL IN THE ART

17. I understand that an assessment of claims of the '981 patent should be undertaken from the perspective of a person of ordinary skill in the art as of the earliest claimed priority date, which I understand is May 2003.

18. In my opinion, a person of ordinary skill in the art as of May 2003 would have possessed at least a bachelor's degree in computer science (or equivalent degree or experience) with at least four years of practical experience or coursework in the design or development of systems for network-based communication between computer systems, including systems for sending and receiving messages between computers using known technologies such as Web Services and XML.

19. My opinions regarding the level of ordinary skill in the art are based on, among other things, my over 25 years of experience in the field of network communications, computer science and engineering, my understanding of the basic qualifications that would be relevant to an engineer or scientist tasked with

investigating methods and systems in the relevant area, and my familiarity with the backgrounds of colleagues and co-workers, both past and present.

20. Although my qualifications and experience exceed those of the hypothetical person having ordinary skill in the art defined above, my analysis and opinions regarding the '981 patent have been based on the perspective of a person of ordinary skill in the art as of May 2003.

## IV. STATE OF THE ART OF THE RELEVANT TECHNOLOGY AT THE TIME OF THE ALLEGED INVENTION

21. The '981 patent generally discloses a computer-based system for managing web services. In this section, I provide a brief background of the state of web services technology prior to May 2003 pertinent to the '981 patent.

22. "Web services" were not an invention of the '981 patent. Web services were an outgrowth of the World Wide Web phenomenon that began in the mid 1990s. (Chappell, Ex. 1007, at p.7.) In particular, during the early days of the web, business could be conducted using straightforward and simple technologies. For example, using the HyperText Transfer Protocol ("HTTP"), a web server received a request from a client such as a web browser, and processed the request using the Common Gateway Interface ("CGI"), which provided a way for the web server to access an external application such as business application.

The web server could then return a response to the web browser in the form of a HyperText Markup Language ("HTML") web page. (Fox, Ex. 1008, at p.482-83.)

23. But the industry realized that as web-based business grew, especially larger enterprises such as rental car companies and airlines, there was a need for coordination among a potentially large number of distributed systems. (Chappell, Ex. 1007, at p.7.) "Web services" were one of a number of technologies that attempted to address that issue. (Chappell, Ex. 1007, at p.1; *see also id.* at p.9 ("[T]he base [web services technologies] are not themselves very exciting; they are just new dressing for the same old distributed-computing model.").)

24. As described in the Background section of the '981 patent, "web services" are "an approach to distributed computing in which interactions are carried out through the exchange of eXtensible Markup Language (XML) messages." ('981, 1:55-58.) That characterization is generally consistent with other publications on the subject. (*E.g.*, Scribner, Ex. 1009, at p.10 ("Web Services can be described as any functionality that is accessible over the Internet, generally (but not necessarily) using one or more eXtensible Markup Language (XML) messages in the communications protocol."); Chappell, Ex. 1007, at p.1 ("A

web service is a piece of business logic, located somewhere on the Internet, that is accessible through standard-based Internet protocols such as HTTP or SMTP.").)

25. The term "XML" generally refers to an industry-standard set of rules for encoding information. XML "provides a standard format for computer documents," and "is flexible enough to be customized for domains as diverse as web sites, electronic data interchange, vector graphics, genealogy, real-estate listings, object serialization, remote procedure calls, and voice-mail systems, and more." (Harold, Ex. 1010, at p.3.) XML was particularly desirable because it promised a document format that could be shared between computer systems and application programs. By 2001, it was recognized that "XML is one of the most important developments in document syntax in the history of computing," and had "become the syntax of choice for newly designed document formats across almost all computer applications." (Harold, Ex. 1010, Preface, xi.)

26. The specification of the '981 patent acknowledges that web services were already being deployed commercially before the alleged invention. ('981, 2:44-45 ("Enterprises are adopting Web services technology to address their business integration needs[.]").) For example, the technology was being used to

integrate the booking systems of Dollar Rent A Car Systems, Inc. and Southwest Airlines Co. (Chappell, Ex. 1007, at p.6.)

27. One such commercial system was WebLogic Collaborate by BEA Systems, Inc. A press release dated February 2001 (more than two years before the filing date of the '981 patent) explains that "BEA's open and extensible B2B collaboration platform, BEA WebLogic Collaborate, integrates trading partners across the Web and enables complex Web Services to be deployed with transactional integrity, security, and reliability." (Ex. 1011, at p.2.) As I discuss in more detail in **Part VI.A** below, the prior art references describing WebLogic Collaborate that I discuss in this Declaration describe features to monitor and manage a web service, including a web-browser-based Administration Console.

## V. THE '981 PATENT'S TECHNIQUE FOR MANAGING WEB SERVICES

### A. The Specification of the '981 Patent

28. As mentioned above, the '981 patent, entitled "Systems and Methods for Managing Web Services via a Framework of Interfaces," generally describes a web service management system that allows a manager to monitor and control associated web services. Figure 1A provides a general overview of one embodiment of the management system:

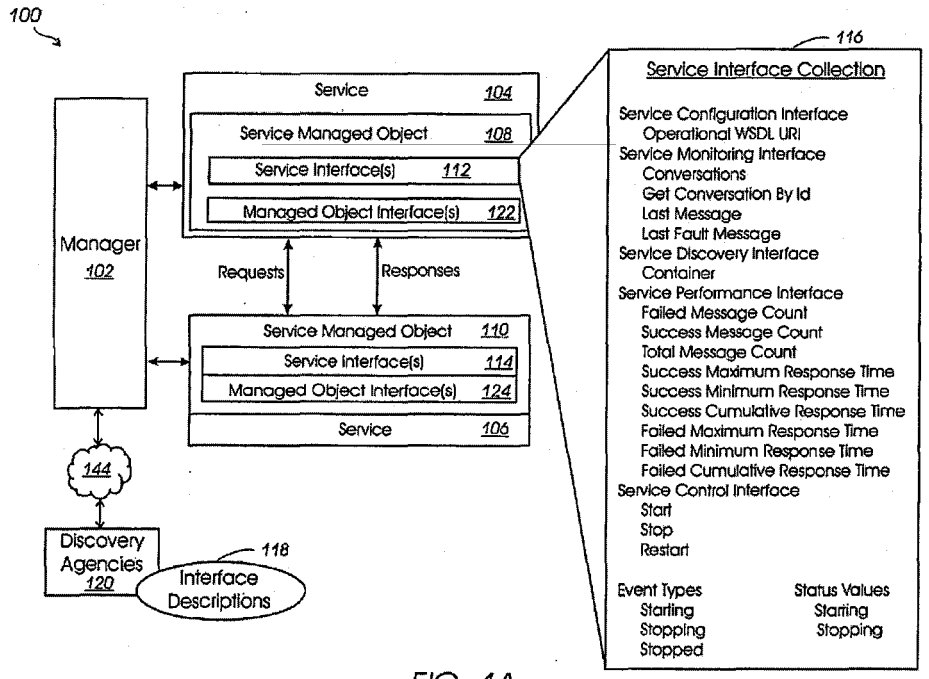



*FIG. 1A*

('981, Fig. 1A.)

29. The management system includes a "service managed object" (e.g., **110**) that has an interface for exposing management features of an associated web service (e.g., **106**) to a manager (e.g., **102**). With reference to Figure 1A (above), the '981 patent describes a "service managed object" as follows:

> <u>Service managed objects **108, 110** represent the management features of resource(s) that perform services **104, 106**</u>. Interfaces in one or more categories can be included in service interfaces **112, 114** for each service managed object **108, 110**. Service interfaces **112, 114** can allow manager **102** to access information regarding the state of services **104, 106**, as well as to control the operation of services

**104, 106.**

('981, 4:51-60 (underlining added).)

30. The '981 patent also states that "[s]ervice managed objects . . . can be considered managed objects." ('981, 7:27-29.) A "managed object" is described as "a management representation of a resource." ('981, 7:26-29.) The '981 patent describes "resources" as broadly including "documents, images, downloadable files, services, electronic mailboxes, and other resources." ('981, 5:66-6:2.) The '981 patent further states that a "[m]anaged object . . . implements managed object interfaces . . . to provide a common set of basic management capabilities to monitor and/or control the underlying resource(s) represented by managed object . . . through various features such as attributes, operations, and event notifications." ('981, 7:30-35.)

31. As mentioned above, a service managed object provides an interface for exposing management features of an associated web service to a manager, such as a list of "conversations" associated with the service. The '860 patent (which the '981 patent incorporates by reference) describes a "conversation" as follows:

> The term "conversation" is a set of related messages sent and received by a particular conversation. Conversations **104, 106** are typically invoked by other resources, such as Web services (not shown). The messages received by a particular conversation **104, 106** may be sent by more than one other conversation, and a particular resource, such as a Web service, can invoke multiple conversations that may or may not be related to the resource's other conversations.

('860, 4:45-52 (underlining added).)

**B. The Claims of the '981 Patent**

32. The two independent claims addressed in this Declaration—i.e., claims 1 and 22—purport to recite a system and computer program product for managing a web service. The first independent claim addressed in this Declaration is claim 1, which recites:

> 1. A system for managing a Web service, comprising:
>
> **[a]** a computer processor; and
>
> **[b]** a service managed object executable on the computer processor, wherein:
>
> **[c]** the service managed object is associated with the Web service and includes at least one interface configured to allow a manager to access management features for the Web service; and

**[d]** the at least one interface is configured to provide a list of conversations associated with the Web service.

('981, 19:34-43 (Claim 1).) I added the bracketed notations (e.g., "[a]," "[b]," etc.) to facilitate easier identification of these limitations in my Declaration. The second independent claim addressed in this Declaration is claim 22:

22. A computer program product tangibly embodied in a computer readable storage medium, comprising:

**[a]** a service interface; and

**[b]** a managed object interface associated with the service interface, wherein

**[c]** the service interface is configured to include information for managing a Web service, including information indicating conversations associated with the service that are in progress.

('981, 21:31-39 (Claim 22).) The other claim addressed in this Declaration—i.e., claim 23—depends from independent claim 22 listed above. I address that claim in more detail in **Part VI** below.

### C. Claim Construction

33. I have been informed by counsel that invalidity analysis is a two-step process. In the first step, the scope and meaning of a claim is determined by construing the terms of that claim. In the second step, the claim as interpreted is

compared to the prior art. Thus, before I address the application of the prior art to the claims of the '981 patent in **Part VI** below, I provide constructions for certain terms in those claims.

34. I have been informed by counsel that a claim in an unexpired patent subject to *inter partes* review must be given its "broadest reasonable construction in light of the specification of the patent in which it appears," which is different from the manner in which the scope of a claim is determined in litigation. I apply the "broadest reasonable construction" standard in my analysis below.

**1. "Web Service"**

35. The term "Web service" appears in both of the independent claims (i.e., claims 1 and 22) that I address in this Declaration. The term is discussed in the Background section of the specification, which states:

> The term Web services, also referred to herein as "services", describes an approach to distributed computing in which <u>interactions are carried out through the exchange of eXtensible Markup Language (XML) messages</u>. . . . Essentially any transaction or bit of business logic can become a Web service if it can be accessed and used by another system over a network such as the Internet.

('981, 1:55-67 (underlining added).)

36. In **Part IV** above, I provided a general overview of web services technology as it existed prior to the '981 patent. In my opinion, the statement in the "Background" section quoted above is generally consistent with how one of ordinary skill in the art would have understood "Web service" as of May 2003. Accordingly, I have interpreted the term "Web service" under its broadest reasonable construction to mean "a service or system that interacts with another system through the exchange of eXtensible Markup Language (XML) messages."

**2. "Managed Object" and "Service Managed Object"**

37. The term "managed object" is recited in a number of ways in the claims. Independent claim 1 recites a "service managed object," while independent claim 22 recites a "managed object interface." I will therefore separately address "managed object" and "service managed object."

**a. "Managed Object"**

38. The term "managed object," generally speaking, refers to an object (such as a software program, process or system) that is responsible for managing a resource. In my opinion, the broadest reasonable construction of "**managed object**" to one of ordinary skill in the art is "**an object for managing a resource**." I derive this definition from the following passage of the specification:

> Referring to FIG. 1B, an embodiment of managed object **128** with managed object interfaces **130** is shown. Managed object **128** is a management representation of a resource. For example, service managed objects **108**, **110** in FIG. 1A can be considered managed objects **128**.
>
> Managed object **128** implements managed object interfaces **130** to provide a common set of basic management capabilities that allow manager **102** to monitor and/or control the underlying resource(s) represented by managed objects **128** through various features such as attributes, operations, and event notifications.

('981, 7:30-35 (underlining added).) This is consistent with an earlier passage in the specification stating that "[s]ervice managed objects **108**, **110** represent the management features of resource(s) that perform services **104**, **106**." ('981, 4:53-55 (underlining added).)

39. The specification describes "resources" broadly as including "documents, images, downloadable files, services, electronic mailboxes, and other resources." ('981, 5:67-6:2.) The specification also describes "management" as including "managed object identity, monitoring, discovery, control, performance, configuration, and security." ('981, 5:8-11.) Based on my

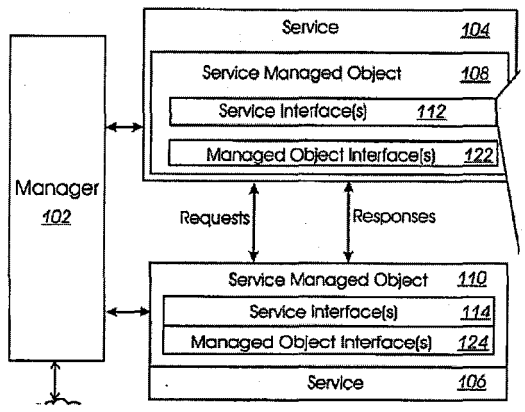
analysis of the specification, I believe the broadest reasonable construction of "**managed object**" is "**an object for managing a resource.**"

**b. "Service Managed Object"**

40. In my opinion, a "service managed object" is simply a managed object with one additional feature – it is "associated with a service." I therefore believe "**service managed object**" under its broadest reasonable interpretation is "**an object for managing a resource that is associated with a service.**"

41. The specification makes clear that that service managed objects can be considered managed objects. ('981, 7:27-29.) The key distinction between the two types of object is that a "service managed object," according to the specification, performs a service and has an "interface" to allow a manager to access management features for the service:

> Service managed objects **108, 110** represent <u>the management features of resource(s) that perform services **104, 106**</u>. Interfaces in one or more categories can be included in service interfaces **112, 114** for each service managed object **108, 110**. Service interfaces **112, 114** can allow <u>manager **102** to access information</u>



> regarding the state of services **104**, **106**, as well as to control the operation of services **104**, **106**.

('981, 4:53-60 (underlining added), Fig. 1A (excerpt).)

42. The claim language is consistent with this description. Claim 1, for example, recites "the service managed object [that] is associated with the Web service and includes at least one interface configured to allow a manager to access management features for the Web service." The underlined portion of the claim language reflects the "service" aspect of the managed object.

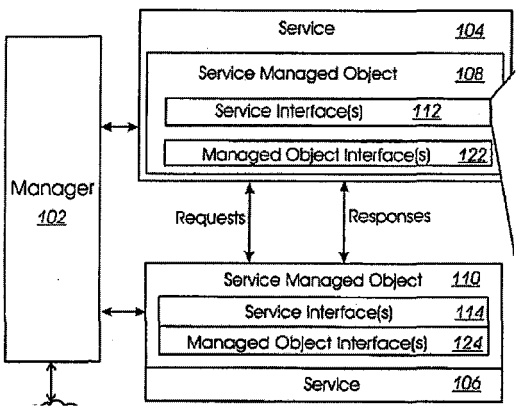
43. The specification further makes clear that although a "service managed object" is associated with a service, the term imposes no requirement of any particular physical relationship between the "service managed object" and its associated "service." Service managed objects, according to the specification, "can be implemented within services **104**, **106**, such as shown for service managed object **108** [in Fig. 1A], or in a layer external to services **104**, **106**, as shown for service managed object **110**." ('981, 4:67-5:3.)

44. Thus, in light of the specification and claim language, I interpret the term "**service managed object**" under its broadest reasonable construction to mean "**an object for managing a resource that is associated with a service.**"

### 3. "Manager"

45. Independent claim 1 recites a "manager." As I explained above in connection with the term "service managed object," claim 1 states that the "manager" can access management features for a web service.

46. The specification of the '981 patent does not provide specific detail regarding the implementation of the claimed "manager." In Figure 1A (excerpted at right), the manager is depicted as box **102** shown on the left of the figure. "Referring to FIG 1A," the specification states, "an embodiment of a Web service management system **100** that allows <u>manager **102** to monitor and control one or more services **104, 106**</u> is shown." ('981, 4:51-53.)




47. The term "manager" does not refer to a human being, but rather, to a software process or system. The specification of the related U.S. Patent No. 7,945,860 **[Ex. 1003]**, which is incorporated-by-reference into the '981 patent (at '981, 1:7-12), makes clear that the "manager" is software-based. ('860, Ex. 1003, 8:57-62 ("In the embodiments shown, <u>manager 102</u> . . . [is] <u>implemented in computer processing systems 160 through 168, respectively</u>." (underlining

added)).) Additionally, the specification explains that the manager (**102**) exchanges information with a Web Service Execution Environment (WSEE). ('981, e.g., 11:50-53 ("WSEE **146** populates data fields in WSEE interfaces **150** that are available to manager **102**. Further, WSEE **146** receives information and control parameters from manager **102** via WSEE interfaces **150**."), 12:43-45 ("Manager **102** can register to receive notification of one or more of the Event types available in WSEE Interface Collection **152**.").) The '981 patent otherwise describes the "manager" in functional terms largely mirroring the language of claim 1. (*E.g.*, '981, 3:43-47 ("The service managed object is associated with the Web service and includes at least one interface configured to allow a manager to access management features for the Web service"), 4:51-53, 4:57-60.)

48. Accordingly, based on the description in the specification, in my opinion, the broadest reasonable construction of "**manager**" is "**a software process or system for accessing management features**."

**4. "Interface," "Managed Object Interface," "Service Interface"**

49. Independent claim 1 recites an "interface" associated with the "service managed object," while independent claim 22 recites a "service

interface" and an associated "managed object interface." I will therefore first address the common term "interface," and then turn to the more specific terms.

**a. "Interface"**

50. The specification does not expressly define "interface." The term "interface" is broadly used by persons of ordinary skill in the art to generally describe something that connects two computer systems, processes or actors together. For example, a "graphical user interface," a term familiar to many computer users, generally describes a way in which a computer makes it features available to a user through icons, windows, buttons and other visual indicators. A graphical user interface is an "interface" because it facilitates interaction between the computer and its human operator.

51. Another common use of "interface" is the term "application programming interface" (API), which refers to a concept that is well known in the art. A person of ordinary skill in the art would have understood that an API generally provides a way for a set of software services to be accessed by another software system. APIs exist at many levels of computer software design and usually take the form of a set of defined software routines, procedures, functions or methods that invoke the software services they represent. For example,

operating systems such as Microsoft Windows and UNIX provide APIs to allow user applications to interact with the operating system to perform tasks such as creating and opening files, sending and receiving information over a network, and receiving user input from a keyboard, mouse or other input device. An API is an "interface" because it facilitates interaction between two different software programs or processes. Similarly, the term "interface" is frequently used in the computer hardware context to describe components such as busses that connect components of a computer system.

52. Both of these examples are consistent with the definition of "interface" found in the Microsoft Computer Dictionary (5th ed. 2002) **[Ex. 1012]** which defines "interface" as "[t]he point at which a connection is made between two elements so that they can work with each other or exchange information." (Ex. 1012, at p.279.) The patent specification uses the term "interface" in a way that is consistent with this definition. For example, as mentioned above with respect to "service managed object," the specification states that a "service managed object is associated with the Web service and includes at least one interface configured to allow a manager to access management features for the Web service." ('981, 3:43-46.) Thus, consistent with its plain and ordinary

meaning to persons of ordinary skill in the art, I believe the term "**interface**" under its broadest reasonable construction is "**a connection point for communication and/or exchange of information.**"

**b. "Managed Object Interface"**

53. Independent claim 22 recites a "managed object interface." The specification of the '981 patent generally describes a "managed object interface" as an interface associated with a managed object. For example, the specification states that "FIG. 1A also shows managed object interfaces **122** associated with service managed object **108**, and managed object interfaces **124** associated with service managed object **110**. Referring to FIG. 1B, an embodiment of managed object **128** with managed object interfaces **130** is shown." ('981, 7:22-26.)

54. Thus, consistent with its plain and ordinary meaning to persons of ordinary skill in the art, I interpret the term "**managed object interface**" to mean "**an interface associated with a managed object (i.e. an object for managing a resource).**"

**c. "Service Interface"**

55. Independent claim 22 also recites a "service Interface." As with "managed object interface," the specification of the '981 patent generally

describes a "service interface" as an interface associated with a service. For example, specification states that: "Service interfaces **112**, **114** [Fig. 1A] can allow manager **102** to access information regarding the state of services **104**, **106**, as well as to control the operation of services **104**, **106**." ('981, 4:57-60.) I therefore have interpreted "**service interface**" under its broadest reasonable construction to mean "**an interface associated with a service**."

**5. "Conversation"**

56. Independent claims 1 and 22 both recite "conversations" associated with a service. As I explained below, in my opinion, the term "**conversation**" should be understood to mean "**a set of related messages for exchange of information**."

57. The specification states that services can exchange information using "conversations." ('981, 14:38-41 ("Information . . . can be exchanged via conversations **214**, **216** between online ordering service **204** and billing service **210**.").) With reference to reference to Figures 1C and 1D, the specification also states that a conversation is associated a set of related messages:

> FIG. 1C shows service **104** containing conversation **156** layered with conversation managed object **158**, which represents the management features of resource(s) associated with conversation

> **156**. Similar to service managed object **108**, and WSEE managed object **148**, conversation managed object **158** can include one or more categories of interfaces in conversation interfaces **160** shown in FIG. 1D as Conversation Interface Collection **162**. Conversation interfaces **160** allow manager **102** to access information regarding the state of <u>messages related to corresponding conversations **104**</u>.

('981, 12:63-13:6 (underlining added).)

58. The specification of the related U.S. Patent No. 7,945,860 **[Ex. 1003]**, which is incorporated-by-reference into the '981 patent (at '981, 1:7-12), provides a similar, though circular, description of a "conversation":

> <u>The term "conversation" is a set of related messages sent and received by a particular conversation</u>. Conversations **104**, **106** are typically invoked by other resources, such as Web services (not shown). The messages received by a particular conversation **104**, **106** may be sent by more than one other conversation, and a particular resource, such as a Web service, can invoke multiple conversations that may or may not be related to the resource's other conversations.

('860, Ex. 1003, 4:45-52.)

59. Consistent with the above descriptions, I interpret "**conversation**" under its broadest reasonable construction to mean "**a set of related messages for exchange of information.**"

## VI. APPLICATION OF THE PRIOR ART TO THE CLAIMS OF THE '981 PATENT

60. I have reviewed and analyzed the prior art references and materials listed in **Part I.B** above. In my opinion, each and every limitation of claims 1, 22 and 23 is disclosed by three documents that describe various features and aspects of "WebLogic Collaborate" software by BEA Systems, Inc.: **(1)** "Introducing BEA WebLogic Collaborate" ("Introducing Collaborate") **[Ex. 1004]**; **(2)** "Administering BEA WebLogic Collaborate" ("Administering Collaborate") **[Ex. 1005]**; and **(3)** "Programming BEA WebLogic Collaborate Management Applications" ("Programming Collaborate") **[Ex. 1006]**. I refer to these references collectively as the "**Collaborate References**." Certain limitations are also disclosed by the combination of the Collaborate References with David Fox & Troy Downing, Web Publisher's Construction Kit with HTML 3.2 (1995) ("Fox") **[Ex. 1008]**.

61. I am informed that the Collaborate References qualify as prior art because they bear a date of July 2001, which is before the May 2003 date applicable to the '981 patent. I provide further analysis in **Part VI.D** below

regarding the public accessibility of these references. Fox qualifies as prior art because it bears a date of publication of 1996. Before explaining how the prior art applies to the claim limitations, I will briefly summarize it.

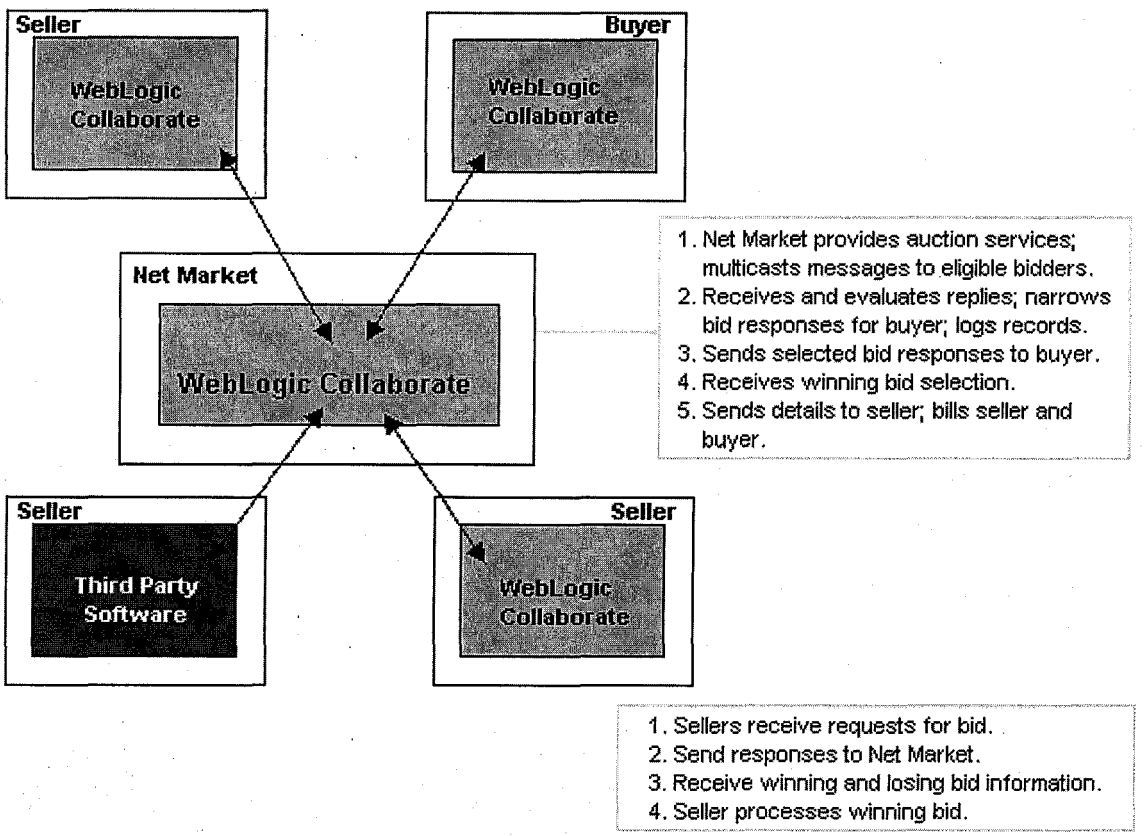
### A. Brief Description and Summary of the Prior Art

62. The "Collaborate References" collectively describe certain operation and management features of a software program called "**WebLogic Collaborate**," Release 2.0, from BEA Systems, Inc. In broad overview, WebLogic Collaborate is a software program designed to facilitate an exchange of messages (*e.g.* through "conversations") between an entities and their "**trading partners**." The "trading partners" of an entity include other entities with which it conducts business, such as its customers and suppliers. (Introducing Collaborate, Ex. 1004, at 1-4, 1-6 (Fig. 1-1), 1-7 (Fig. 1-2).) "Business partners in an e-commerce community can range in size from large enterprises to small divisions within an enterprise." (*Id.* at 1-4.)

63. As explained in Introducing Collaborate, the term "trading partner" refers to "an entity that has an agreement with another entity to participate in a specific business exchange, or conversation, in a specific role that is defined for the conversation." (*Id.*, Ex. 1004, at 1-5.) For example, an organization could set

up a community for inventory management in which its "trading partners" consisted of corporate departments within the organization. (*Id.* at 1-4.)

64. The "trading partners" involved run the WebLogic Collaborate software (or compatible collaboration software provided by a third party) to form an e-community. An example of such an "e-community" is shown in Figure 1-2 below, which shows use of the WebLogic Collaborate for an auction service:




(*Id.* at 1-7 (Fig. 1-2).) Figure 1-2 above shows use of WebLogic Collaborate for an auction service in which a "Net Market" (the center box) serves as an auction

broker between a Buyer and Sellers. (*Id.*)

65. Once joined into an e-community, these trading partners may participate in a "conversation." Within WebLogic Collaborate, a "conversation" is "a series of business messages exchanged between trading partners." (*Id.* at 1-7 to 1-8; *see also id.* ("The business messages that can be exchanged between participants in the conversation are determined by the roles the trading partners play in the conversation. The roles and other details of a conversation are specified in a *conversation definition* using the WebLogic Collaborate Administration Console. A conversation is an active instance of a *conversation definition*." (emphasis in original).)

66. A conversation "[m]ay be complex and long-running, or short-lived." (*Id.*) WebLogic Collaborate operates as a "Web service" in that messages exchanged between trading partners take the form of eXtensible Markup Language (XML) messages using standard Web protocols such as HTTP. (*Id.* at 1-13 ("A business message is the basic unit of communication among trading partners and is exchanged as part of a conversation. A <u>business message contains one or more XML business documents, one or more attachments, or a combination of both</u>." (emphasis added)), 1-1 ("WebLogic Collaborate supports

HTTP because the World Wide Web is the ubiquitous communication medium for e-business.").)

67. Finally, WebLogic Collaborate provides an "Administration Console" that provides a user interface allowing a user to manage various aspects of the collaboration system. (*Id.* at 1-30 to 1-31.) Additional detail regarding the disclosures of the Collaborate References is provided below.

68. Fox is a 1996 web programming textbook describing various well-known Web technologies such as using the Common Gateway Interface (CGI). My Declaration relies on Fox in combination with the Collaborate References for the "interface" limitations of the challenged claims. A further discussion of Fox is provided below in the discussion of the claim limitations for I have cited it.

**B. The Collaborate References Are Combinable**

69. Before turning to the disclosures of the Collaborate References as they apply to the '981 patent, I have been asked to provide my opinion as to whether those three references can properly be combined to establish obviousness. In my opinion, they clearly can.

70. In my opinion, one of ordinary skill in the art would have found the Collaborate References to be entirely combinable. All three documents describe

aspects of the same software program, BEA WebLogic Collaborate, and even the same version of that software (Release 2.0). The Collaborate References were authored by BEA Systems, Inc. and bear the same date (July 2001).

71. One of ordinary skill in the art would naturally have treated the Collaborate References as a group of related documents and consulted them together to ascertain the various features of WebLogic Collaborate.

72. In fact, the Collaborate References cite and make express references to each other. For example, Introducing Collaborate includes a section entitled "**Document Roadmap for WebLogic Collaborate**" that lists a number of other documents that the reader can consult to "find more detailed information about various features of your WebLogic Collaborate." (Introducing Collaborate, Ex. 1004, at 1-32.) That section specifically lists the other two Collaborate References. (*Id.* at 1-32 (listing Administering Collaborate), 1-34 (listing Programming Collaborate).)

73. Similarly, Administering Collaborate describes a "**WebLogic Collaborate document set**" that includes all three of the Collaborate References, among others. (Administering Collaborate, Ex. 1005, at xi-xii.) The Collaborate References are also replete with specific cross-references to each other

throughout various sections. (*See, for example*, Introducing Collaborate, Ex. 1004, at 1-14 ("For details about administering both system and custom logic plug-ins, see *Administering BEA WebLogic Collaborate*"); *id.* at 1-31 ("For complete details about using the WebLogic Collaborate Administration Console, see *Administering BEA WebLogic Collaborate*"), *id.* ("For details about programming management applications, see *Programming BEA WebLogic Collaborate Management Applications.*").)

74. The Collaborate References, in fact, specifically encourage the reader to consult each other. (*See, for example*, Administering Collaborate, Ex. 1005, at x ("Before reading this document, we recommend that you read the *Introducing BEA WebLogic Collaborate* document.").) A person of ordinary skill in the art, therefore, would have been amply motivated to combine the disclosures of the Collaborate References and would have considered them part-and-parcel of a single disclosure.

75. With respect to some claim limitations relating to interfaces, I have applied the teachings of the Fox reference (Ex. 1008) in combination with the Collaborate References. Because I have applied Fox only with respect to certain

limitations, I will address the rationale and motivation to come it with the Collaborate References in the claim limitations to which I have applied it.

76. My declaration has applied the teachings of the Collaborate References and Fox to claims 1, 22 and 23 of the '981 patent. In my opinion, the Collaborate References provide sufficient detail to enable a person of ordinary skill in the art to practice the limitations of the claims to which they apply without undue experimentation. Although the Collaborate References describe a complex web services platform, WebLogic Collaborate, claims 1, 22 and 23 describe only a tiny fraction of its functionality.

77. As explained in **Part IV** above, web services were not an invention of the '981 patent, and their underpinnings were firmly in place by May 2003. Claims 1, 22 and 23 describe one aspect of managing a web services, identifying and monitoring conversations. The subject matter of these claims could have been implemented using known technologies such as HTML, XML and Java.

78. These technologies existed long before May 2003, and documentation about web technologies (such as the Fox reference that predates the patent by more than six years) was widely-available. In light of the information available about web services and their underlying technologies, a

person of ordinary skill in the art in May 2003 could have implemented what is described in claims 1, 22 and 23 based on the disclosures in the Collaborate References, without undue experimentation.

### C. Each Limitation of Claims 1, 22 and 23 is Disclosed by the Collaborate References and Fox

#### 1. Claim 1

79. The preamble of claim 1 recites, "[a] system for managing a Web service." The Collaborate References disclose this. The "**Web service**" in the Collaborate References takes the form of the **WebLogic Collaborate system**.

80. As explained above, the term "Web service" should be understood as "a service or system that interacts with another system through the exchange of eXtensible Markup Language (XML) messages." As explained in the "Overview" section of Introducing Collaborate:

> The BEA WebLogic Collaborate™ product is an XML- and Java-based e-commerce platform that enables you to implement complex e-commerce systems on the Web. . . XML is used as a standard format for documents exchanged by business partners. WebLogic Collaborate supports HTTP because the World Wide Web is the ubiquitous communication medium for e-business.

(Introducing Collaborate, Ex. 1004, at 1-1 (emphasis added).) The WebLogic Collaborate system therefore clearly qualifies as a "Web service."

81. The Collaborate References also disclose "a system for managing a Web service." In particular, they describe a series of "administrative services" for managing the WebLogic Collaborate service. (*Id.* at 1-30 to 1-31.)

> The WebLogic Collaborate administration services support multiple system management functions, including configuring, administering, and monitoring trading partners, conversations, collaboration agreements, and more.
> Through these services, a WebLogic Collaborate administrator can create, configure, and manage the components of the WebLogic Collaborate system.

(Introducing Collaborate, Ex. 1004, at 1-30 (under "Administration Services") (emphasis added).) The Collaborate References therefore clearly disclose "a system for managing a Web service."

**a. "a computer processor" (Claim 1[a])**

82. The first limitation of claim 1 recites "a computer processor." The Collaborate References disclose or suggest this limitation.

83. In particular, the Collaborate References explain that BEA WebLogic Collaborate is a software product that runs, for example, on the Microsoft

Windows and UNIX operating systems. (*See* Administering Collaborate, Ex. 1005, at 1-5 ("On a Windows system, you can start WebLogic Collaborate with the program icons or from the command line."); *id.* at 1-7 (describing starting WebLogic Collaborate in UNIX).) Additionally, the Collaborate References explain that "WebLogic Collaborate is implemented entirely in Java and leverages the J2EE standard APIs." (Introducing Collaborate, Ex. 1004, at 1-1.) The term "Java" and "J2EE" generally refer to an object-oriented programming language and platform originally developed by Sun Microsystems. One of ordinary skill in the art would have understood that, in order for a system to execute WebLogic Collaborate on a Microsoft Windows or Unix system, or to run a Java-based program, the system would have included at least one computer processor.

**b. "a service managed object executable on the computer processor" (Claim 1[b])**

84. The second limitation of claim 1 recites "<u>a service managed object executable on the computer processor</u>." The "service managed object" in the Collaborate References take the form of a collection of software programs known as the "**BEA WebLogic Collaborate Managed Beans**," also referred as "**MBeans**."

85. These "Managed Beans" or "MBeans" are what is known as "JavaBeans." Java, as I mentioned earlier, refers a programming language and

software development platform. The basic unit of Java program is known as a Java "class." The term "**JavaBean**" describes a particular way to encapsulate a Java class that, generally speaking, was intended to make software components in Java more reusable from one program to another. Accordingly, the "Managed Beans" or "MBeans" in the Collaborate References describe software functionality encapsulated as "Java Beans" that, when executed by a processor in a computer, perform certain functions.

86. The collection of these "Managed Beans" corresponds to the "service managed object" recited in the claim. The term "service managed object," as explained above, means "<u>an object for managing a resource that is associated with a service</u>." The "**service**" here is WebLogic Collaborate, the "Web service" as discussed previously. The "**resource**" being managed is an aspect of the WebLogic Collaborate service, such as a run-time instance, delivery channel or trading partner session. (Programming Collaborate, Ex. 1006, at 2-3 to 2-4, Table 2-1 (listing WebLogic resources managed by MBeans).) As explained below, each MBean manages at least one of these resources.

87. The Collaborate References confirm that the Managed Beans (the "service managed object") qualify as objects for managing resources associated with the service (e.g. WebLogic Collaborate):

> This document introduces WebLogic Collaborate MBeans and management applications. MBeans are one of three types of component applications available within WebLogic Collaborate. Management applications use MBeans to monitor WebLogic Collaborate.

(Programming Collaborate, Ex. 1006, at 1-1 (under "WebLogic Collaborate Applications") (underlining added).) As noted previously, the specification of the '981 patent specifically lists "monitoring" as an activity that qualifies as "managing" a resource. ('981, 5:8-11.) The Collaborate References further explain:

> For all management applications, WebLogic Collaborate provides a set of Managed Beans, or *MBeans*, which are special JavaBeans with attributes and methods for management operations.

(Programming Collaborate, Ex. 1006, at 1-3 (italics in original; underlining added).)

88. The Collaborate References describe at least six different MBeans that are associated with WebLogic Collaborate. (*Id.* at 2-3, Table 2-1.) For

example, the MBean called "WLCMBean" is "[u]sed for monitoring a WebLogic Collaborate instance at run time." (*Id.* (first item in Table 2-1; underlining added).) The other MBeans monitor other resources for WebLogic Collaborate such as delivery channels, active transactions, trading partners, and messages. (*Id.* at 2-3 to 2-4, Table 2-1.) Again, as noted previously, "monitoring" qualifies as "managing a resource" under the '981 patent. ('981, 5:8-11.)

89. These MBeans, individually or as a group, clearly qualify as "an object for managing a resource that is associated with a service," *i.e.* the WebLogic Collaborate service. Each MBean manages a resource associated with the WebLogic Collaborate service including its run-time instance, delivery channels, and so forth. (Programming Collaborate, Ex. 1006, at 2-3 to 2-4, Table 2-1.) The Collaborate References therefore disclose "a service managed object executable on the computer processor," as recited in the claim.

**c. "the service managed object is associated with the Web service and includes at least one interface configured to allow a manager to access management features for the Web service" (Claim 1[c])**

90. For convenience and ease of reading, I have broken this claim limitation into two parts and will address each portion separately.

**i. "the service managed object is associated with the Web service" (Claim 1[c], first part)**

91. The "Managed Beans" or "MBeans" in the Collaborate References satisfy the first part of claim 1[c], "the service managed object is associated with the Web service." As noted above, the Managed Beans are associated with the Web service, *i.e.* WebLogic Collaborate. (Programming Collaborate, Ex. 1006, at 1-1 ("Management applications use MBeans to monitor WebLogic Collaborate."), 1-3.) For example, the Managed Bean called "WLCMBean" "[r]epresents a WebLogic Collaborate instance" and is "[u]sed for monitoring a WebLogic Collaborate instance at run time." (Programming Collaborate, Ex. 1006, at 2-3, Table 2-1 (first item in table; underlining added).) The Collaborate References therefore disclose that "the service managed object is associated with the Web service," as recited in the first part of claim 1[c].

**ii. "the service managed object . . . includes at least one interface configured to allow a manager to access management features for the Web service" (Claim 1[c], second part)**

92. The Collaborate References also disclose that the Managed Beans (the "service managed object") include "at least one interface configured to allow a manager to access management features for the Web service," as recited in the

second part of claim 1[c]. Because the "**interface**" recited in the claim language must provide a "**manager**" with access to management features, I will first address how the Collaborate References disclose the claimed "manager."

93. The "**manager**" in the Collaborate References takes the form of an **Administration Console**, a web-based user interface that can be accessed by an administrator using an Internet web browser such as Internet Explorer. (*See* Administering Collaborate, Ex. 1005, at 1-8 to 1-9, Figure 1-1.) The Administration Console provides access to management features of the Web service, *i.e.* WebLogic Collaborate:

> <u>You can use the WebLogic Collaborate Administration Console to</u>:
>
> - <u>Configure WebLogic Collaborate preferences</u>, trading partners, conversation definitions, collaboration agreements, business protocol definitions, and logic plug-ins
> - Export and import configured elements
> - <u>Monitor and control WebLogic Collaborate</u>, trading partner sessions, conversations, and collaboration agreements

(Administering Collaborate, Ex. 1005, at 1-7 (underlining added).) Having addressed the "**manager,**" I now turn to the claimed "**interface.**"

94. The Collaborate References also disclose the claimed "<u>at least one interface</u>," which is "configured to allow" the Administration Console ("manager")

to the access management features for WebLogic Collaborate (the "Web Service"). There are at least two separate and independent ways in which the claimed "**interface**" can be mapped to the prior art.

95. First, the "**interface**" can take the form of **Application Programming Interfaces (APIs) provided by the Managed Beans or MBeans**, which provide management features. The Collaborate References state that the Administration Console (the claimed "**manager**") uses these MBeans APIs to access management features:

> WebLogic Collaborate provides the application programming interfaces (APIs) needed to create custom management applications that monitor run-time activity on WebLogic Collaborate nodes. The WebLogic Collaborate Administration Console tools also use these APIs to provide real-time monitoring information.
> These APIs consist of sets of Java Management Extensions (JMX) Managed Beans, or MBeans, which are special JavaBeans with attributes and methods for management operations.

(Programming Collaborate, Ex. 1006, at 2-2 (under "MBeans and the MBean Server") (underlining added); *see also id.* at 2-5 ("The WebLogic Collaborate Administration Console uses the JMX API and WebLogic Collaborate MBeans to monitor running WebLogic Collaborate instances.") (underlining added).)

96. As I discussed in the previous claim limitations, these Managed Beans or MBeans provide various management features, including monitoring WebLogic Collaborate Instances and monitoring conversations. (Programming Collaborate, Ex. 1006, at 2-3 to 2-4, Table 2-1.)

97. An Application Programming Interface or "API," including the MBeans API discussed above, satisfies the "**interface**" claim limitation, as I explained in **Part V.C** above, because it provides the connection point for communication and/or exchange of information between the Administration Console and the MBeans. Accordingly, in my opinion, the Collaborate References disclose that the Managed Beans or MBeans (the "service managed object") include "<u>at least one interface configured to allow a manager to access management features for the Web service</u>," as recited in claim 1[c].

98. As noted previously, there is a second way in which the prior art discloses the claimed "**interface**." The "Administration Console," as noted above, is a web-based user interface that allows an administrator to access management features. (*See* Administering Collaborate, Ex. 1005, at 1-8 to 1-9, Figure 1-1.) One of ordinary skill in the art would have appreciated that the web server that provides the web pages for the Administration Console would have included an

"**interface**" to receive user selections through the Administration Console, and then use that input to interact with other software (such as the MBeans), for example, to retrieve requested information.

99. Such interfaces were well known to persons of ordinary skill in the art by May 2003. An example of such an interface is the **Common Gateway Interface (CGI)** disclosed in Fox. Fox, which was published more than six years before the filing date of the patent, describes various technologies for allowing any web site (such as the Administration Console discussed above) to interact with external programs. One such technology was known as the "Common Gateway Interface," which is an interface that allows a web server to communicate with other software programs. As explained in Fox:

> The Common Gateway Interface—or CGI—is a method that lets you access external programs on a Web server and usually send the results to a Web browser. . . These programs can be any executable code, script or program supported by the operating system that runs your server.

(Fox, Ex. 1008, at 482 (under "Lesson #1: What is CGI?") (underlining added).) These programs that can be accessed using CGI are called "CGI scripts," and they can be written in any programming language supported by the server. (*Id.*) Fox

further explains that CGI scripts could be used for various purposes including accessing databases:

> Why is it called the Common "Gateway" Interface? Well, the answer is simple: The Common Gateway Interface was originally intended as a "gateway" between WWW clients and other programs that could be run remotely on your server. Many CGI scripts, especially those that access databases, simply execute another application on the server and redirect its output with whatever formatting changes are required to the HTTP server, and then to the client that requested the script.

(*Id.* at 490 (under "Lesson #4: How to Use a Script to Access Other Applications") (underlining added).) Fox further explains that a program accessed through the CGI interface can perform a wide range of tasks. "It can access other programs, open files, read from files, create graphics, dial your modem, call your mother, do database searches, send e-mail, you name it." (*Id.* at 484 (under "What Can I Do with a CGI Script?") (underlining added).)

100. Although I believe that the Collaborate References alone disclose the claimed "**interface**," the claimed interface would also have been obvious over the Collaborate References in view of the interface teachings in Fox. This provides a

second and independent ground for showing the obviousness of the claimed "interface."

101. It would have been obvious to one of ordinary skill in the art to adapt Fox's teachings about web server interfaces such as the Common Gateway Interface (CGI) to the Collaborate References, with no change in their respective functions. This would have predictably resulted in a system in which the Administration Console in WebLogic Collaborate (the claimed "**manager**") used a web server interface, such as the Common Gateway Interface (CGI) in Fox, as an interface between the web server that receives user input and external programs (such as the Managed Beans or MBeans) that provide functionality in response to that user input. This would have allowed the Administration Console (the claimed "**manager**") to access management features for the WebLogic Collaborate system (the claimed "**Web service**") service using its web-based user interface. (Administering Collaborate, Ex. 1005, at 3-2 ("The WebLogic Collaborate Administration Console is used to . . . [m]onitor WebLogic Collaborate, trading partner sessions, conversations, and collaboration agreements.").)

102. One of ordinary skill in the art would have had many motivations to combine. The technology was not complex. Web server interfaces to external

programs (CGI being just one example) were part of the basic repertoire of knowledge possessed by persons of ordinary skill in the art well before May 2003. As explained in Fox in 1996, "[a]s you read this, CGI scripts are coming to life all over the world." (Fox, Ex. 1008, at 485.) CGI scripts were so pervasive that, as explained in Fox, "CGI scripts are used for doing all of the 'cool' stuff on the Net." (*Id.*) One of ordinary skill in the art would certainly have recognized that in order to implement dynamic web pages such as the Administration Console pages, which are generated in response to requests and queries, one could employ an interface such as CGI to facilitate interaction with external programs.

**d. "the at least one interface is configured to provide a list of conversations associated with the Web service." (Claim 1[d])**

103. The final limitation of claim 1 requires that "<u>the at least one interface is configured to provide a list of conversations associated with the Web service</u>." I explained in **Part V.C** that the term "**conversation**" means "a set of related messages for exchange of information." The Collaborate References disclose this limitation.

104. As I explained above for claim 1[c], the Administration Console uses the interface (API) provided by the MBeans to access management features for

the WebLogic Collaborate system. These features include retrieving a list of conversations associated with the WebLogic Collaborate system (the "Web service"). As noted above, the Collaborate References explain that a "conversation" is "a series of business messages exchanged between trading partners." (Introducing Collaborate, Ex. 1004, at 1-7 to 1-8, *see also* ¶¶ 65-66 above.) This language indicates that "conversations" in the Collaborate References are equivalent to the "conversations" recited in the claims.

105. The Collaborate References disclose several ways in which "a list of conversations associated with the Web service," *i.e.*, WebLogic Collaborate, may be provided:

There are three options for listing active XOCP conversations:

- *List the conversations for a selected conversation definition*
  To list the conversations for a selected conversation definition, select a conversation definition from the navigation tree or Conversations page, then select the Monitoring tab to <u>view the list of active conversations for that conversation definition</u>.
- *List the conversations for a selected delivery channel*
  To list the conversations for a selected delivery channel, list delivery channels as described in Monitoring Delivery Channels" on page 6-9. When you select a delivery channel from the list, the

number of conversations for that delivery channel is displayed as part of the summary statistics for the delivery channel. You can view a list of the conversations for that delivery channel by clicking the number.

- *List the conversations for a trading partner session*

  To list the conversations for a selected trading partner session, list trading partner sessions as described in "Monitoring Trading Partner Sessions" on page 6-7. When you select a trading partner session from the list, the number of conversations for that trading partner session is displayed as part of the summary statistics for the trading partner session. You can view a list of the conversations for that trading partner session by clicking the number.

(Administering Collaborate, Ex. 1005, at 6-10 (under "Monitoring Conversations") (italics in original; underlining added).) Selecting a conversation from a conversation list will bring up a window showing information about the selected conversation such as the conversation identifier, start time of the conversation, and other information. (*Id.* at 6-11 (Figure 6-5).)

106. The quoted text above refers to "XOCP," which stands for the eXtensible Open Collaboration Protocol (XOCP). (Introducing Collaborate, Ex. 1004, at 1-14 (under "XOCP").) XOCP is "a BEA-specific business protocol" for

providing messaging services. (*Id.*) WebLogic Collaborate supports several different messaging protocols, including XOCP. (*Id.* at 1-13 ("WebLogic Collaborate supports the following business protocols: ▪ XOCP (eXtensible Open Collaboration Protocol). . .") (middle of page).)

107. The prior art also makes clear that the conversation lists discussed above were provided by the claimed "**interface**." As noted for the preceding claim limitation, I have described at least independent two ways in which the prior art discloses the "interface" recited in the claim: **(1)** application programming interfaces (APIs) provided by the Managed Beans or MBeans; and **(2)** a web server interface (such as the Common Gateway Interface (CGI)) in Fox. Either theory fully discloses an interface that "is configured to provide a list of conversations associated with the Web service," as recited in the claim.

108. With respect to the first theory in which the "interface" takes the form of MBeans APIs, one those APIs called "`getActiveConversations`" provides the list of the conversations in a WebLogic Collaborate session:

> MBeans that are logically related have accessor methods to retrieve references to each other. These methods are strongly typed and return exact MBean types. For example, the `WLCMBean.getActiveDeliveryChannels()` method returns an array of type DeliveryChannelMBean that represents all the

> active delivery channels in the system. Similarly, the `TradingPartnerSessionMBean.getActiveConversations()` method returns an array of type `ConversationMBean` that represents all the active conversations in this session.

(Programming Collaborate, Ex. 1006 at 2-10 (under "Step 6: Navigate Across MBeans") (underlining added).)

109. The paragraph above, in plain English, describes services provided by MBean APIs referred to above as "accessor methods." Generally speaking, a "method" in Java (and in object oriented programming in general) refers to software or instructions that can be invoked (or "called") from another software process by using the method's name. An "accessor method" generally refers to a method that retrieves ("gets") data from an object.

110. In the text quoted above, "getActiveConversations()" refers to an accessor method that can be invoked by an application – such as the Administration Console – to retrieve a list of active conversations in the WebLogic Collaborate session. As explained above, this method "returns an array of type ConversationMBean that represents all the active conversations in this session." (*Id.* at 2-10 (underlining added).)

111. This accessor method is part of the MBean API (the claimed "**interface**") as previously mentioned. The Collaborate References further disclose that the Administration Console uses the MBeans APIs to provide the list of conversations. (*Id.* at 2-2 ("The WebLogic Collaborate Administration Console tools also use these APIs to provide real-time monitoring information.") (under "MBeans and the MBean Server"), 2-5 ("The WebLogic Collaborate Administration Console uses the JMX API and WebLogic Collaborate MBeans to monitor running WebLogic Collaborate instances.").) In my opinion, therefore, the Collaborate References disclose that "the at least one interface is configured to provide a list of conversations associated with the Web service," as recited in claim 1[d].

112. With respect to the second theory I outlined above in which the "**interface**" takes the form of a web server interface (such as the Common Gateway Interface (CGI) of Fox), one of ordinary skill in the art would have recognized that an interface such as CGI could also have been "configured to provide a list of conversations associated with the Web service." This is because the list of conversations is reported back to the user through a web page accessible through the Administration Console, as explained above. (*See also* Administering Collaborate, Ex. 1005, at 3-12 (Figure 3-8).) The ability to

dynamically generate a web page in response to user input, such as a list of conversations in the Administration Console, is a key reason to use a web server interface such as the Common Gateway Interface (CGI). (Fox, Ex. 1008, at 485 (describing possible uses of CGI including ability to "create graphic images on-the-fly," "serve maps," "send you live video feeds" and "access huge databases").)

113. Under either theory, the prior art discloses that "the at least one interface is configured to provide a list of conversations associated with the Web service," as recited in the claim. For all of these reasons, therefore, the Collaborate References and Fox disclose each element of claim 1.

**2. The Collaborate References and Fox Disclose Claim 22**

114. Claim 22 is similar in many respects to claim 1, but lists the elements in slightly different order. The Collaborate References disclose all limitations of claim 22 for many of the same reasons as claim 1.

115. The preamble of claim 22 recites, "[a] computer program product tangibly embodied in a computer readable storage medium." As I explained in connection with claim 1[a] above, BEA WebLogic Collaborate is a Java-based computer program product that runs on a Windows- or UNIX-based computer system. (*See* Administering Collaborate, Ex. 1005, at 1-5 ("On a Windows system,

you can start WebLogic Collaborate with the program icons or from the command line."); *id.* at 1-7 (describing starting WebLogic Collaborate in UNIX); Introducing Collaborate, Ex. 1004, at 1-1 ("WebLogic Collaborate is implemented entirely in Java and leverages the J2EE standard APIs.").)

116. The Collaborate References make clear that the WebLogic Collaborate is stored on a computer readable storage medium. For example, the Collaborate References explain that the WebLogic Collaborate software is installed in particular directories. For example, in order to start WebLogic Collaborate from Windows, the Collaborate References tell the user to navigate to a directory called "\bea\wlintegration2.0\collaborate\config\mydomain" and execute a command called "startWeblogic." (*Id.* at 1-6 (under "Starting WebLogic Collaborate from the Command Line).)

117. One of ordinary skill in the art would understand that in order for the software to be embodied in the form of files contained in directories in a computer filing system, the computer includes a "computer readable storage medium" such as a hard disk drive or other type of digital storage device. The Collaborate References therefore disclose "[a] computer program product tangibly embodied in a computer readable storage medium."

### a. "service interface" limitations (Claims 22[a], 22[b])

118. Claim 22 recites a "service interface," followed by a "managed object interface," and then ends with lengthy "wherein" clause that refers to the "service interface." In the interest of ease of reading, and to keep the discussion of common claim limitations on one place, I will first address the "service interface" limitations, and then address the "managed object" term.

#### i. "a service interface" (Claim 22[a])

119. The first limitation of claim 22 recites "a service interface." I explained above that the term "service interface" means "an interface associated with a service." The "**service interface**" described in claim 22 is substantially similar, for purposes of my application of the claims to the Collaborate References, to the "**interface**" recited in claim 1 that I discussed above.

120. As with claim 1, there are at least two separate and independent ways in which the prior art discloses the claimed "service interface." First, the Collaborate References disclose a "**service interface**" in the form of **Application Programming Interfaces (APIs) provided by the Managed Beans or MBeans,** which provide management features:

> WebLogic Collaborate provides the <u>application programming interfaces (APIs)</u> needed to create custom management applications

> that monitor run-time activity on WebLogic Collaborate nodes. The WebLogic Collaborate Administration Console tools also use these APIs to provide real-time monitoring information.
>
> These APIs consist of sets of Java Management Extensions (JMX) Managed Beans, or MBeans, which are special JavaBeans with attributes and methods for management operations.

(Programming Collaborate, Ex. 1006, at 2-2 (under "MBeans and the MBean Server"); *see also id.* at 2-5 ("The WebLogic Collaborate Administration Console uses the JMX API and WebLogic Collaborate MBeans to monitor running WebLogic Collaborate instances.").)

121. The Managed Beans or MBeans API is associated with a service, *i.e.* the WebLogic Collaborate service. (Programming Collaborate, Ex. 1006, at 1-1 ("Management applications use MBeans to monitor WebLogic Collaborate."), 1-3.) For example, the Managed Bean called "WLCMBean" "[r]epresents a WebLogic Collaborate instance" and is "[u]sed for monitoring a WebLogic Collaborate instance at run time." (Programming Collaborate, Ex. 1006, at 2-3, Table 2-1 (first item in table; underlining added).) The Collaborate References therefore disclose "a service interface," as recited in claim 1[a].

122. Under the second theory that I discussed above, the "service interface" takes the form of a web interface such as the **Common Gateway Interface (CGI)** disclosed in Fox. This interface, as discussed previously, provides a "gateway" in which a web server (such as the one providing the Administration Console) can interact with another application program to perform certain tasks, such as querying databases. (Fox, Ex. 1008, at 482 (under "Lesson #1: What is CGI?"); *id.* at 490 (under "Lesson #4: How to Use a Script to Access Other Applications"); *id.* at 494 (under "What Can I Do with a CGI Script?").)

123. As explained for claim 1[d] above, it would have been obvious to one of ordinary skill in the art to adapt Fox's teachings about the Common Gateway Interface (CGI) to the Collaborate References, with no change in their respective functions. This would have predictably resulted in a system in which the Administration Console in WebLogic Collaborate (the claimed "**manager**") used a web server interface, such as the Common Gateway Interface (CGI) in Fox, as an interface associated with a service, *i.e.* the WebLogic Collaborate system. (Administering Collaborate, Ex. 1005, at 3-2 ("The WebLogic Collaborate Administration Console is used to . . . [m]onitor WebLogic Collaborate, trading partner sessions, conversations, and collaboration agreements.").)

124. As I explained for claim 1[d] above, one of ordinary skill in the art would have had many motivations to combine. Web server interfaces to external programs (CGI being just one example) were part of the basic repertoire of knowledge possessed by persons of ordinary skill in the art well before May 2003. As explained in Fox in 1996, "[a]s you read this, CGI scripts are coming to life all over the world." (*Id.* at 485.) CGI scripts were so pervasive that, as explained in Fox, "CGI scripts are used for doing all of the 'cool' stuff on the Net." (*Id.*) One of ordinary skill in the art would certainly have recognized that in order to implement dynamic web pages such as the Administration Console pages, which are generated in response to requests and queries, one could employ an interface such as CGI to facilitate interaction with external programs. In my opinion, therefore, the Collaborate References (alone or in combination with Fox) disclose the "**service interface**" of claim 22[a].

**ii. "wherein the service interface is configured to include information for managing a Web service, including information indicating conversations associated with the service that are in progress" (Claim 22[b])**

125. Both of the interfaces that I described above for the claimed "**service interface**," *i.e.* the MBeans API and the web server interface (such as CGI), include

"information for managing a Web service, including information indicating conversations associated with the service that are in progress."

126. With respect to the first theory that I outlined above in which the claimed "**service interface**" comprises the MBeans APIs, as I explained for claim 1[d] above, the Managed Beans or MBeans provide information for monitoring WebLogic Collaborate conversations. For example, the Managed Beans can produce lists of conversations for display in the WebLogic Collaborate Administration Console. (See **Part VI.C.1** above; *see also* Administering Collaborate, Ex. 1005, at 6-10 (under "Monitoring Conversations").) The Collaborate References also explain that the APIs provided by the Managed Beans or MBeans (the "service interface"), such as the getActiveConversations() method discussed for claim 1[d] above, "<u>return[] an array</u> of type ConversationMBean that <u>represents all the active conversations in this session</u>." (Programming Collaborate, Ex. 1006 at 2-10 (under "Step 6: Navigate Across MBeans") (underlining added).)

127. With respect to the second theory that I outlined above in which the claimed "**service interface**" comprises a web server interface such as the Common Gateway Interface (CGI), as I explained for claim 1[d] above, an interface

such as CGI could also have included "information for managing a Web service, including information indicating conversations associated with the service that are in progress." This is because the list of conversations is reported back to the user through a web page accessible through the Administration Console, as explained above. The web server interface would have included "information indicating conversations associated with the service that are in progress" because, as noted previously, that interface was used to obtain and deliver the web page for the Administration Console listing the active conversations. (Fox, Ex. 1008, at 490 (under "Lesson #4: How to Use a Script to Access Other Applications").)

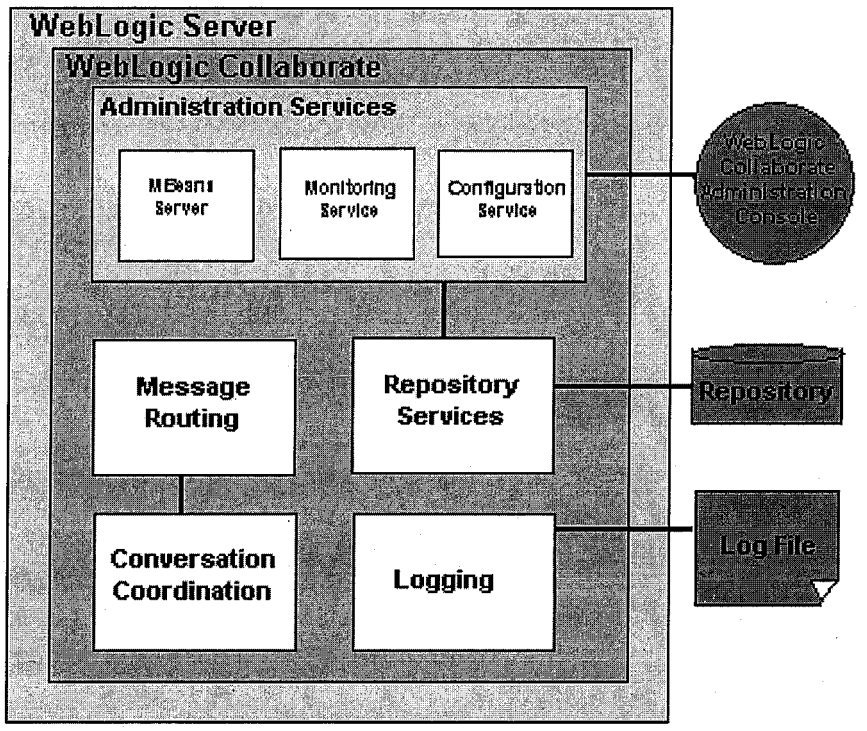
128. As I explained previously, one of ordinary skill in the art would have had ample motivation to combine. The ability to dynamically generate a web page in response to user input, such as a list of conversations in the Administration Console, is a key reason to use a web server interface such as CGI. (Fox, Ex. 1008, at 485.) Therefore, for the reasons stated in claim 1[d] and above, the Collaborate References alone, or in combination with Fox, disclose that the service interface "is configured to include information for managing a Web service, including information indicating conversations associated with the service that are in progress."

**b. "a managed object interface associated with the service interface" (Claim 22[b])**

129. The second limitation of claim 22 recites "a managed object interface associated with the service interface." There is no further mention of this interface in claim 22. The claim does not expressly recite any particular function this interface must perform, it simply requires that it exist. In any event, the Collaborate References disclose this limitation.

130. As explained in **Part V.C** above, a "managed object interface" is "an interface associated with a managed object (i.e. an object for managing a resource)." For purposes of claim 22, the "**managed object**" takes the form of **a repository service** that manages a resource, *e.g.* configuration and other information for WebLogic Collaborate. (Introducing Collaborate, Ex. 1004, at 1-29 ("The repository service stores data into the repository."); Administering Collaborate, Ex. 1005, at 7-1 ("The repository is a database that stores configuration information for WebLogic Collaborate.") (under "Working with the Repository").) The repository, among other things, supports importing and exporting data and other system administration tasks using the Administration Console. (Introducing Collaborate, Ex. 1004, at 1-29 to 1-30, 2-19 to 2-21.)

131. As with the "service interface" discussed above, there are at least two ways in which the prior art discloses the claimed "managed object interface." First, the "**managed object interface**" takes the form of the interface that facilitates the connection between the repository services and the Administration Console. The relationship between these components is shown in Figure 1-9 of Introducing Collaborate:




(Introducing Collaborate, Ex. 1004, at 1-29 (Figure 1-9: "WebLogic Collaborate Services").) As shown in Figure 1-29 above, the Repository Services (shown in the middle row) interacts with the WebLogic Collaborate Administration Console

(shown on the top right) through the "Administration Services." (*Id.* at 1-28 ("WebLogic Collaborate system components are configured and managed primarily through the WebLogic Collaborate Administration Console, which works together with a repository service.") (underlining added).)

132. The Collaborate References explain that "the repository [i]s accessible through the WebLogic Collaborate Administration Console for system administration, creation of collaboration agreements by business developers, and monitoring of trading partners, conversations, collaboration agreements, and so on." (Introducing Collaborate, Ex. 1004, at 1-30.) The Collaborate references therefore disclose an "interface" associated with the repository service.

133. The Collaborate References disclose that this managed object interface is "associated with the service interface" under both of the theories that I outlined above. In the case in which the "service interface" comprises the Managed Beans or MBeans APIs, this association is clearly shown in Figure 1-9 above. That figure shows "Administrative Services" facilitating connection from the Administration Console, for both the MBeans Server (which provides the MBeans and their associated APIs) and the Repository Services. The repository service is associated with the MBeans APIs that comprise the "service interface"

because, among other things, both interfaces work together to provide management features for the WebLogic Collaborate Administration Console.

134. This managed object interface is also "associated with the service interface" under my second theory in which the service interface comprises a web server interface such as CGI used to obtain conversation information. Under this theory, the "managed object interface" is associated with the web server interface because they both interact with the WebLogic Collaborate Administration Console.

135. Second, although I believe that the Collaborate References alone disclose the "**managed object interface**," this interface could also be found through the combination of the Collaborate References and Fox. As I have explained at length above, web server interfaces such as CGI provide a mechanism for a web server to receive input from a web browser through a web page (such as the Administration Console), and interface with an external application to create a customized web page in response to the user's request. (Fox, Ex. 1008, at 484-85.) It would have been obvious to one of ordinary skill in the art to adapt the web server interface and CGI teachings of Fox to the Collaborate References, with no change in their respective functions.

136. This would predictably have resulted in a system in which the Administration Console interfaced with a web server interface, such as the Common Gateway Interface (CGI) in Fox, to invoke a process for accessing the WebLogic Collaborate repository. Fox expressly discloses that one of the purposes for which CGI is used is to "do database searches" (*id.* at 484), or "access huge databases" (*id.* at 485). The WebLogic repository, as noted above, comprises a database. (Administering Collaborate, Ex. 1005, at 7-1 ("The repository is a database that stores configuration information for WebLogic Collaborate.") (under "Working with the Repository").) One of ordinary skill in the art would therefore have been motivated to use a web server interface such as CGI to provide an interface for the managed object, *i.e.* the WebLogic repository. Finally, as noted above, the managed object interface is "associated with the service interface" because both interfaces work together to provide management features for the WebLogic Collaborate Administration Console.

### 3. The Collaborate References and Fox Disclose Claim 23

137. Claim 23 depends from independent claim 22 and recites, "[t]he computer program product of claim 22, wherein the service interface is further configured to include information regarding at least one of the group of: **[i]** the

last requested message received by the service; **[ii]** the last fault message returned from the service; and **[iii]** the execution environment for the service." I explained above in **Part VI.C.2** that the Collaborate References alone, or in combination with Fox, disclose or suggest every limitation of claim 22.

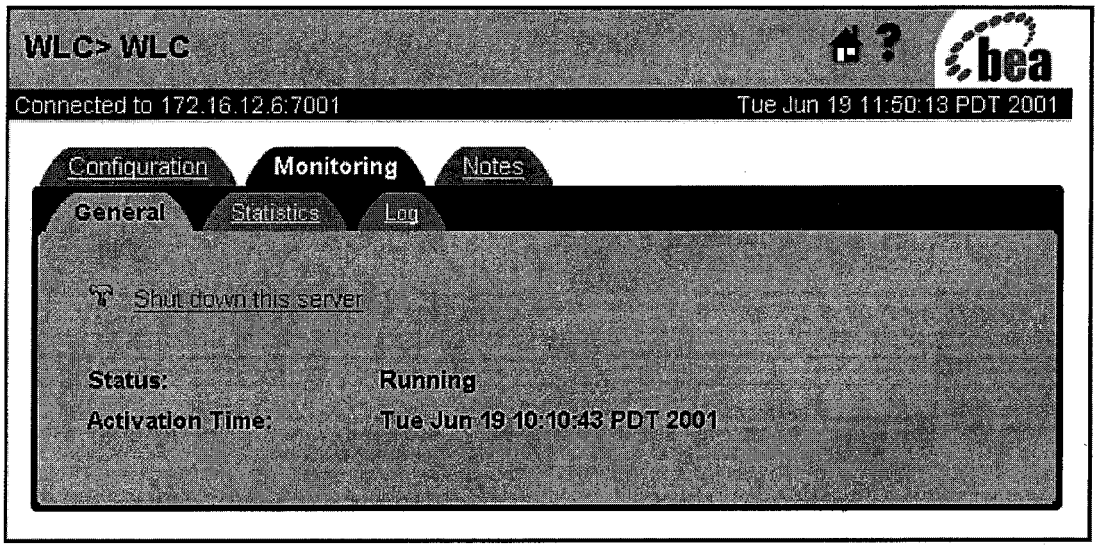
138. The Collaborate References also disclose the additional features in claim 23, which requires that the service interface is configured to provide information falling into at least one of the three categories in the claim. As shown below, although the claims appear to require that only one of these be met, the Collaborate References disclose all three.

139. First, the Collaborate References disclose that the service interface could include information regarding "<u>the last requested message received by the service</u>." In particular, the Collaborate References explain that the Administration Console includes a number of "Monitoring Pages." (Administering Collaborate, Ex. 1005, at 6-3 to 6-4 (under "WebLogic Administration Console Monitoring Pages").) The monitoring page for WebLogic Collaborate can display summary statistics including the "number of messages sent and received, <u>time of last message sent and received</u> . . ." (*Id.* at 6-4 (Table 6-1, "WebLogic Collaborate (WLC) page") (underlining added).) Because messages are part of a conversation

that is initiated and carried about among trading partners, the messages are "requested."

140. Second, the Collaborate References disclose that the service interface could include information regarding "the last fault message returned from the service." For example, in the monitoring page for trading partners, the Administration Console can display "details of a selected session," such as "time of last sent and received message, time of first and last failed message. . ." (*Id.* at 6-4, Table 6-1 "Trading partner page") (underlining added).)

141. Third, the Collaborate References disclose that the service interface could include information regarding "the execution environment for the service." For example, the WebLogic Collaborate (WLC) monitoring page in the Administration Console can show execution environment information such as "[s]tatus (running or inactive)," and "[s]hutdown options (immediate or terminate)." (*Id.* at 6-4 (Table 6-1, "WebLogic Collaborate (WLC) page") (underlining added).) For example, Figure 6-1 shows a page for monitoring the status of the WebLogic Collaborate server:




(*Id.* at 6-6 (Figure 6-1).) For all of these reasons, therefore, the Collaborate References disclose all limitations of claim 23.

### D. The Collaborate References Are Properly Prior Art

142. Finally, I have been asked to provide my opinions on whether the Collaborate References can properly be considered prior art to the '981 patent. I am informed by counsel that, in order for a document to qualify a prior art printed publication under the patent laws, the document has to have been disseminated or otherwise made available so persons of ordinary skill in the art, exercising reasonable diligence, could locate it. In my opinion, the Collaborate References satisfy this requirement.

143. The Collaborate References show a date of "July 2011" on their face page and on the copyright page. I have reviewed the "Affidavit of Christopher Butler" from the Internet Archive, which I understand is being submitted as Exhibit 1014 with the Petition for *Inter Partes* Review. I am generally familiar with the Internet Archive and have used it since at least 2005. As explained in Mr. Butler's declaration, the Internet Archive includes a service known as the "Wayback Machine" that allows users to browse from more than 400 billion archived web pages.

144. His Declaration attaches a webpage entitled "BEA WebLogic Collaborate 2.0: PDF" that on its face shows a listing of download links to various documents (in PDF form), including the Collaborate References. (Ex. 1014.) The Uniform Resource Locator (URL) shown at the bottom of that web page, "https://web.archive.org/web/20010829204911/http://e-docs.bea.com/ wlintegration/v2_0/collaborate/interm/pdf.htm," based on the URL encoding rules for the Internet Archive described in the Butler Affidavit, indicates that the HTML on the web page was accessible through the web at least by August 29, 2001 at 08:49:11 PM. (*See* Butler Affidavit, Ex. 1014, ¶ 5 (explaining how the digits in the URL indicate the date and time the page was archived).)

145. In my opinion, the Collaborate References were sufficiently disseminated or otherwise available so persons of ordinary skill in the art exercising reasonable diligence could have located them. The documents were, based on the Butler Affidavit, available on the web through BEA's corporate website by at least August 2001. Based on my review of the documents related to BEA, persons of ordinary skill in the art could have readily located the Collaborate References.

146. The Collaborate References themselves make clear that BEA made an effort to promote its "e-docs Web Site" (e-docs.bea.com) as a central source of documentation about its products. (Introducing Collaborate, Ex. 1004, at vi ("The WebLogic Collaborate product documentation is available on the BEA Systems, Inc. corporate Web site."); Administering Collaborate, Ex. 1005, at x ("From the BEA Home page, click on Product Documentation or go directly to the e-docs" Product Documentation page at http://e-docs.bea.com."); *id.* at xi ("A PDF version of this document is available from the BEA WebLogic Collaborate documentation Home page, which is available on the documentation CD and on the e-docs Web site at http://e-docs.bea.com.").)

147. The download page that contained links to PDF versions of the Collaborate References (Exhibit A to the Butler Affidavit) also included detailed instructions for users on downloading and using the PDF documents using a web browser, and a link to download Adobe Acrobat if the user did not already have it installed on his or her computer. (Ex. 1014, Ex. A, at 1-2.) I have seen no evidence suggesting that a person of ordinary skill in the art would have experienced difficulty locating the Collaborate References from BEA's website.

148. BEA Systems, Inc., the company that produced the Collaborate References, was a known provider of web services products in the early 2000s. In a press release dated August 27, 2001 (approximately the same time period as the date shown on the face of the Collaborate References), BEA stated that it had more than 11,000 customers throughout the world and that its e-business platform was "the de facto standard for more than 2,100 systems integrators, independent software vendors (ISVs) and application service providers (ASPs)..." (Ex. 1013.) Additionally, by the July 2001 date listed on the face of the Collaborate References, Internet search engines such as Google, Alta Vista and Yahoo! were available and capable of indexing web pages and retrieving them in response to simple keyword queries. I personally used these Internet search

engines in the early 2000s, including using them to locate and retrieve technical information and product documentation. In light of all of the facts above, in my opinion, the Collaborate References were sufficiently available that persons of ordinary skill in the art could have located them with reasonable diligence.

## VII. CONCLUSION

149. In my opinion, each element of claims 1, 22 and 23 is disclosed or suggested by the prior art references described above, and a person of ordinary skill in the art would have had ample motivation to combine those references. Each challenged claim, therefore, would have been obvious to a person of ordinary skill in the art.

150. In signing this Declaration, I recognize that the Declaration will be filed as evidence in a contested case before the Patent Trial and Appeal Board of the United States Patent and Trademark Office. I also recognize that I may be subject to cross-examination in this proceeding. If required, I will appear for cross-examination at the appropriate time. I reserve the right to offer opinions relevant to the invalidity of the '981 patent claims at issue and/or offer testimony in support of this Declaration.

151. I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true, and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 28 U.S.C. § 1001.

Dated: February 5, 2015

Respectfully submitted,

Tal Lavian 2/5/2015

Tal Lavian, Ph.D.

# EXHIBIT A

# Tal Lavian, Ph.D.

http://telecommnet.com
http://cs.berkeley.edu/~tlavian
tlavian@telecommnet.com

1640 Mariani Dr.
Sunnyvale, CA 94087
(408)-209-9112

## Research and Consulting: Network Communications, Telecommunications, and Internet Protocols

- Scientist, educator, and technologist with over 25 years of experience
- Co-author of over 25 scientific publications, journal articles, and peer-reviewed papers
- Named inventor on over 80 issued and filed patents
- Industry Fellow and Lecturer at UC Berkeley Engineering –Center for Entrepreneurship and Technology (CET)

### EDUCATION

- **Ph.D.**, Computer Science specializing in networking and communications, UC Berkeley
- **M.Sc.**, Electrical Engineering, Tel Aviv University
- **B.Sc.**, Mathematics and Computer Science, Tel Aviv University

### EXPERTISE

Network communications, telecommunications, mobile wireless and Internet protocols:

- **Communication networks:** Internet Protocols; TCP/IP suite; TCP; UDP; IP; VoIP; Ethernet; network protocols; network software applications; Data Link, Network, and Transport Layers (L2, L3, L4)
- **Routing/switching:** LAN; WAN; VPN; routing protocols; RIP; BGP; MPLS;OSPF; IS-IS;DNS; QoS; switching; packet switching; network infrastructure; network communication architectures
- **Mobile Wireless:** Wireless LAN; 802.11; cellular systems; mobile devices; smartphone technologies
- **Internet Software:** Internet software applications; Internet protocols; distributed computing; Web applications; FTP; HTTP; Java; C; C++; client server; file transfer; multicast; streaming media

### LITIGATION SUPPORT SERVICES

- Expert witness in Federal courts, USPTO, and the ITC (over 30 cases)
- Expert witness in USPTO reexamination, interference, and numerous PTAB - IPR
- Expert reports, depositions, and courtroom testimonies
- Skilled articulation of technical material for both technical and non-technical audiences
- Product and technology analysis, patent portfolios, claim charts, patentability research
- Litigation support and technology education in patent disputes, patent reexaminations
- Past cases involved Google, Cisco, Juniper, HP, Ericsson, Microsoft, Apple, and Samsung

## ACCOMPLISHMENTS

- Selected as Principal Investigator for three US Department of Defense (DARPA) projects
- Led research project on networking computation for the US Air Force Research Lab (AFRL)
- Led and developed the first network resource scheduling service for grid computing
- Led wireless research project for an undisclosed US federal agency
- Managed and engineered the first demonstrated transatlantic dynamic allocation of 10Gbs Lambdas as a grid service
- Spearheaded and planned the first demonstrated wire-speed active network on commercial hardware
- Invented over 80 patents; over 50 prosecuted *pro se* in front of the USPTO
- Created and chaired Nortel Networks' EDN Patent Committee
- IEEE Senior Member

## PROFESSIONAL EXPERIENCE

**VisuMenu, Inc.** – Sunnyvale, CA 2010-Present

**Co- Founder and Chief Technology Officer (CTO)**

- Design and develop architecture of visual IVR technologies for smartphones and wireless mobile devices in the area of network communications
- Design crawler/spider system for IVR / PBX using Asterisk, SIP and VoIP
- Deploy the system as cloud networking and cloud computing utilizing Amazon Web Services (EC2, S3, VPC, DNS, and RDS)

**Telecomm Net Consulting, Inc.** (Innovations-IP) Sunnyvale, CA 2006-Present

**Principal Scientist**

- Consult in the areas of network communications, telecommunications, Internet protocols, and smartphone mobile wireless devices
- Provide architecture and system consultation for software projects relating to computer networks, mobile wireless devices, Internet web technologies.
- Expert witness in network communications patent infringement suits

**University of California Berkeley**, Berkeley, CA 2000-Present

**Berkeley Industry Fellow, Lecturer, Visiting Scientist, Ph.D. Candidate, Nortel's Scientist Liaison**

*Some positions and projects were concurrent, others sequential*

- Serve as Industry Fellow and Lecturer at the Center for Entrepreneurship and Technology (CET). Studied the areas of network services, telecommunication systems and software, communications infrastructure, and data centers

- Developed long-term technology for the enterprise market, integrating communication and computing technologies
- Conducted research projects in data centers (RAD Labs), telecommunication infrastructure (SAHARA), and wireless systems (ICEBERG)
- Acted as scientific liaison between Nortel Research Lab and UC Berkeley, providing tangible value in advanced technologies
- Earned Ph.D. in Computer Science, specializing in communications and networking

**Nortel Networks**, Santa Clara, CA 1996 - 2007

*Originally employed by Bay Networks, which was acquired by Nortel Networks*

**Principal Scientist, Principal Architect, Principal Engineer, Senior Software Engineer**

- Held scientific and research roles at Nortel Labs, Bay Architecture Labs, and CTO Office

**Principal Investigator for US Department of Defense (DARPA) Projects**

- Conceived, proposed, and completed three research projects: Active Networks, DWDM-RAM, and a networking computation project for Air Force Research Lab (AFRL)
- Led a wireless research project for an undisclosed US federal agency

**Academic and Industrial Researcher**

- Analyzed new technologies to reduce risks associated with R&D investment
- Spearheaded research collaboration with leading universities and professors at UC Berkeley, Northwestern University, University of Amsterdam, and University of Technology Sydney
- Evaluated competitive products relative to Nortel's products and technology
- Proactively identified prospective business ideas, leading to new networking products
- Predicted technological trends in advance through researching the technological horizon and academic sphere
- Developed software for switches, routers and network communications devices
- Developed systems and architectures for switches, routers, and network management
- Researched and developed the following projects:
  - Data-Center Communications: network and server orchestration 2006-2007
  - DRAC: SOA-facilitated L1/L2/L3 network dynamic controller 2003-2007
  - Omega: classified wireless project for undisclosed US Federal Agency 2006
  - Open Platform: project for the US Air Force Research Laboratory (AFRL) 2005
  - Network Resource Orchestration for Web Services Workflows 2004-2005

- Proxy Study between Web/Grids Services and Network Services 2004
- Streaming Content Replication: real-time A/V media multicast at edge 2003-2004
- DWDM-RAM: US DARPA-funded program on agile optical transport 2003-2004
- Packet Capturing and Forwarding Service on IP and Ethernet traffic 2002-2003
- CO2: content-aware agile networking 2001-2003
- Active Networks: US DARPA-funded research program 1999-2002
- ORE: programmable network service platform 1998-2002
- JVM Platform: Java on network devices 1998-2001
- Web-Based Device Management: network device management 1996-1997

**Technology Innovator and Patent Leader**

- Created and chaired Nortel Networks' EDN Patent Committee
- Facilitated continuous stream of innovative ideas and their conversion into intellectual property rights
- Developed intellectual property assets through invention and analysis of existing technology portfolios

**Aptel Communications**, Netanya, Israel 1994-1995

**Software Engineer, Team Leader**

*Start-up company focused on mobile wireless CDMA spread spectrum PCN/PCS*

- Developed mobile wireless device using an unlicensed band, Direct Sequence Spread Spectrum (DSSS)
- Designed and managed a personal communication network (PCN) and personal communication system (PCS), the precursors of short text messages (SMS)
- Responsible for the design and development of network software products
- Developed software network communications mainly in C/C++
- Brought two-way paging product from concept to development

**Scitex Ltd.**, Herzeliya, Israel 1990-1993

**Software Engineer, Team Leader**

*Software and hardware company acquired by Hewlett Packard (HP)*

- Developed system and network communications mainly in C/C++
- Invented Parallel SIMD Architecture
- Participated in the Technology Innovation group

**Shalev**, Ramat-HaSharon, Israel 1987-1990

*Start-up company*

**Software Engineer**

- Developed real-time software and algorithms mainly in C/C++ and Pascal

## PROFESSIONAL ASSOCIATIONS

- IEEE Senior Member
- IEEE CNSV co-chair Intellectual Property SIG (2013)
- President Next Step Toastmasters (an advanced TM club in the Silicon Valley) (2013)
- Technical Co-Chair, IEEE Hot Interconnects 2005 at Stanford University
- Member, IEEE Communications Society (COMMSOC)
- Member, IEEE Computer Society
- Member, IEEE Systems, Man, and Cybernetics Society
- Member, IEEE-USA Intellectual Property Committee
- Member, ACM, ACM Special Interest Group on Data Communication (SIGCOM)
- Member, ACM Special Interest Group on Hypertext, Hypermedia and Web (SIGWEB)
- Member, IEEE Consultants' Network (CNSV)
- Global Member, Internet Society (ISOC)
- President Java Users Group – Silicon Valley Mountain View, CA,1999-2000
- Toastmasters International

### ADVISORY BOARDS

- Quixey – (present) search engine for wireless mobile apps
- Mytopia – mobile social games
- iLeverage – Israeli Innovations

### PROFESSIONAL AWARDS

- Top Talent Award – Nortel
- Top Inventors Award – Nortel EDN
- Certified IEEE-WCET - Wireless Communications Engineering Technologies
- Toastmasters International - Competent Communicator (twice)
- Toastmasters International - Advanced Communicator Bronze

## Patents and Publications

*(notan exhaustive list)*

### Patents Issued:

- **US 8,688,796** Rating system for determining whether to accept or reject objection raised by user in social network
- **US 8,572,303** Portable universal communication device
- **US 8,553,859** Device and method for providing enhanced telephony
- **US 8,548,131** Systems and methods for communicating with an interactive voice response system
- **US 8,537,989** Device and method for providing enhanced telephony
- **US 8,341,257** Grid proxy architecture for network resources
- **US8,161,139** Method and apparatus for intelligent management of a network element
- **US 8,146,090** Time-value curves to provide dynamic QoS for time sensitive file transfer
- **US 8,078,708** Grid proxy architecture for network resources
- **US 7,944,827** Content-aware dynamic network resource allocation
- **US7,860,999** Distributed computation in network devices
- **US 7,734,748** Method and apparatus for intelligent management of a network element
- **US 7,710,871** Dynamic assignment of traffic classes to a priority queue in a packet forwarding device
- **US 7,580,349** Content-aware dynamic network resource allocation
- **US 7,433,941** Method and apparatus for accessing network information on a network device
- **US 7,359,993** Method and apparatus for interfacing external resources with a network element
- **US 7,313,608** Method and apparatus for using documents written in a markup language to access and configure network elements
- **US 7,260,621** Object-oriented network management interface

- **US 7,237,012** Method and apparatus for classifying Java remote method invocation transport traffic
- **US 7,127,526** Method and apparatus for dynamically loading and managing software services on a network device
- **US7,047,536** Method and apparatus for classifying remote procedure call transport traffic
- **US7,039,724** Programmable command-line interface API for managing operation of a network device
- **US6,976,054** Method and system for accessing low-level resources in a network device
- **US6,970,943** Routing architecture including a compute plane configured for high-speed processing of packets to provide application layer support
- **US6,950,932** Security association mediator for Java-enabled devices
- **US6,850,989** Method and apparatus for automatically configuring a network switch
- **US6,845,397** Interface method and system for accessing inner layers of a network protocol
- **US6,842,781** Download and processing of a network management application on a network device
- **US6,772,205** Executing applications on a target network device using a proxy network device
- **US6,564,325** Method of and apparatus for providing multi-level security access to system
- **US6,175,868** Method and apparatus for automatically configuring a network switch
- **US6,170,015** Network apparatus with Java co-processor
- **US 8,619,793** Dynamic assignment of traffic classes to a priority queue in a packet forwarding device
- **US 8687,777** Systems and methods for visual presentation and selection of IVR menu
- **US 8,681,951** Systems and methods for visual presentation and selection of IVR menu

- **US 8,625,756** Systems and methods for visual presentation and selection of IVR menu
- **US 8,594,280** Systems and methods for visual presentation and selection of IVR menu
- **US 8,548,135** Systems and methods for visual presentation and selection of IVR menu
- **US 8,406,388** Systems and methods for visual presentation and selection of IVR menu
- **US 8,345,835** Systems and methods for visual presentation and selection of IVR menu
- **US 8,223,931** Systems and methods for visual presentation and selection of IVR menu
- **US 8,160,215** Systems and methods for visual presentation and selection of IVR menu
- **US 8,155,280** Systems and methods for visual presentation and selection of IVR menu
- **US 8,054,952** Systems and methods for visual presentation and selection of IVR menu
- **US 8,000,454** Systems and methods for visual presentation and selection of IVR menu
- **EP 1,905,211** Technique for authenticating network users
- **EP 1,142,213** Dynamic assignment of traffic classes to a priority queue in a packet forwarding device
- **EP 1,671,460** Method and apparatus for scheduling resources on a switched underlay network
- **CA 2,358,525** Dynamic assignment of traffic classes to a priority queue in a packet forwarding device

**Patent Applications Published and Pending:**

- **US 20140105025** Dynamic Assignment of Traffic Classes to a Priority Queue in a Packet Forwarding Device
- **US 20140105012** Dynamic Assignment of Traffic Classes to a Priority Queue in a Packet Forwarding Device
- **US 20140012991** Grid Proxy Architecture for Network Resources
- **US 20130080898** Systems and Methods for Electronic Communications
- **US 20130022191** Systems and Methods for Visual Presentation and Selection of IVR Menu
- **US 20130022183** Systems and Methods for Visual Presentation and Selection of IVR Menu
- **US 20130022181** Systems and Methods for Visual Presentation and Selection of IVR Menu
- **US 20120180059** Time-Value Curves to Provide Dynamic QOS for Time Sensitive File Transfers
- **US 20120063574** Systems and Methods for Visual Presentation and Selection of IVR Menu
- **US 20110225330** Portable Universal Communication Device
- **US 20100220616** Optimizing Network Connections
- **US 20100217854** Method and Apparatus for Intelligent Management of a Network Element
- **US 20100146492** Translation of Programming Code
- **US 20100146112** Efficient Communication Techniques
- **US 20100146111** Efficient Communication in a Network
- **US 20090313613** Methods and Apparatus for Automatic Translation of a Computer Program Language Code
- **US 20090313004** Platform-Independent Application Development Framework
- **US 20090279562** Content-aware dynamic network resource allocation
- **US 20080040630** Time-Value Curves to Provide Dynamic QoS for Time Sensitive File Transfers
- **US 20070169171** Technique for authenticating network users
- **US 20060123481** Method and apparatus for network immunization
- **US 20060075042** Extensible Resource Messaging Between User Applications and Network Elements in a Communication Network

- **US 20050083960** Method and Apparatus for Transporting Parcels of Data Using Network Elements with Network Element Storage
- **US 20050076339** Method and Apparatus for Automated Negotiation for Resources on a Switched Underlay Network
- **US 20050076336** Method and Apparatus for Scheduling Resources on a Switched Underlay Network
- **US 20050076173** Method And Apparatus for Preconditioning Data to Be Transferred on a Switched Underlay Network
- **US 20050076099** Method and Apparatus for Live Streaming Media Replication in a Communication Network
- **US 20050074529** Method and apparatus for transporting visualization information on a switched underlay network
- **US 20040076161** Dynamic Assignment of Traffic Classes to a Priority Queue in a Packet Forwarding Device
- **US 20020021701** Dynamic Assignment of Traffic Classes to a Priority Queue in a Packet Forwarding Device
- **WO 2007/008976** Technique for Authenticating Network Users
- **WO 2006/063052** Method and apparatus for network immunization
- **WO2000/0054460** Method and apparatus for accessing network information on a network device

## Publications

*(not an exhaustive list)*

- "Communications Architecture in Support of Grid Computing",Tal Lavian,Scholar's Press 2013ISBN 978-3-639-51098-0.
- "Applications Drive Secure Lightpath Creation across Heterogeneous Domains, Feature Topic Optical Control Planes for Grid Networks: Opportunities, Challenges and the Vision." Gommans L.; Van Oudenaarde B.; Dijkstra F.; De Laat C.; Lavian T.; Monga I.; Taal A.; Travostino F.; Wan A.; IEEE *Communications Magazine*, vol. 44, no. 3, March 2006, pp. 100-106.
- *Lambda Data Grid: Communications Architecture in Support of Grid Computing*. Tal I. Lavian, Randy H. Katz; Doctoral Thesis, University of California at Berkeley. January 2006.
- "Information Switching Networks." Hoang D.B.; T. Lavian;*The 4th Workshop on the Internet, Telecommunications and Signal Processing, WITSP2005*, December 19-21, 2005, Sunshine Coast, Australia.
- "Impact of Grid Computing on Network Operators and HW Vendors." Allcock B.; Arnaud B.; Lavian T.; Papadopoulos P.B.; Hasan M.Z.; Kaplow W.; *IEEE Hot Interconnects at Stanford University 2005*, pp.89-90.
- *DWDM-RAM: AData Intensive Grid Service Architecture Enabled by Dynamic Optical Networks*. Lavian T.; Mambretti J.; Cutrell D.; Cohen H.J; Merrill S.; Durairaj R.; Daspit P.; Monga I.; Naiksatam S.; Figueira S.; Gutierrez D.; Hoang D.B., Travostino F.; CCGRID 2004, pp. 762-764.
- *DWDM-RAM: AnArchitecture for Data Intensive Service Enabled by Next Generation Dynamic Optical Networks*. Hoang D.B.; Cohen H.; Cutrell D.; Figueira S.; Lavian T.; Mambretti J.; Monga I.; Naiksatam S.; Travostino F.; Proceedings IEEE Globecom 2004, Workshop on High-Performance Global Grid Networks, Houston, 29 Nov. to 3 Dec. 2004, pp.400-409.
- *Implementation of a Quality of Service Feedback Control Loop on Programmable Routers*. Nguyen C.; Hoang D.B.; Zhao, I.L.; Lavian, T.; Proceedings, 12th IEEE International Conference on Networks 2004. (ICON 2004) Singapore, Volume 2, 16-19 Nov. 2004, pp.578-582.
- *A Platform for Large-Scale Grid Data Service on Dynamic High-Performance Networks*. Lavian T.; Hoang D.B.; Mambretti J.; Figueira S.; Naiksatam S.; Kaushil N.; Monga I.; Durairaj R.; Cutrell D.; Merrill S.; Cohen H.; Daspit P.; Travostino F; GridNets 2004, San Jose, CA., October 2004.
- *DWDM-RAM: Enabling Grid Services with Dynamic Optical Networks*. Figueira S.; Naiksatam S.; Cohen H.; Cutrell D.; Daspit, P.; Gutierrez D.; Hoang D. B.; Lavian T.; Mambretti J.; Merrill S.; Travostino F; Proceedings, 4th IEEE/ACM International Symposium on Cluster Computing and the Grid, Chicago, USA, April 2004, pp. 707-714.
- *DWDM-RAM: Enabling Grid Services with Dynamic Optical Networks*. Figueira S.; Naiksatam S.; Cohen H.; Cutrell D.; Gutierrez D.; Hoang D.B.; Lavian T.; Mambretti J.; Merrill S.; Travostino F.; 4th IEEE/ACM International Symposium on Cluster Computing and the Grid, Chicago, USA, April 2004.
- *An Extensible, Programmable, Commercial-Grade Platform for Internet Service Architecture*. Lavian T.; Hoang D.B.; Travostino F.; Wang P.Y.; Subramanian S.; Monga I.; IEEE Transactions on Systems, Man, and

Cybernetics on Technologies Promoting Computational Intelligence, Openness and Programmability in Networks and Internet Services Volume 34, Issue 1, Feb. 2004, pp.58-68.

- *DWDM-RAM: An Architecture for Data Intensive Service Enabled by Next Generation Dynamic Optical Networks.* Lavian T.; Cutrell D.; Mambretti J.; Weinberger J.; Gutierrez D.; Naiksatam S.; Figueira S.; Hoang D. B.; Supercomputing Conference, SC2003 Igniting Innovation, Phoenix, November 2003.
- *Edge Device Multi-Unicasting for Video Streaming.* Lavian T.; Wang P.; Durairaj R.; Hoang D.; Travostino F.; Telecommunications, 2003. ICT 2003. 10th International Conference on Telecommunications, Tahiti, Volume 2, 23 Feb.-1 March, 2003 pp. 1441-1447.
- The SAHARA Model for Service Composition Across Multiple Providers. Raman B.; Agarwal S.; Chen Y.; Caesar M.; Cui W.; Lai K.; Lavian T.; Machiraju S.; Mao Z. M.; Porter G.; Roscoe T.; Subramanian L.; Suzuki T.; Zhuang S.; Joseph A. D.; Katz Y.H.; Stoica I.; Proceedings of the First International Conference on Pervasive Computing. ACM Pervasive 2002, pp. 1-14.
- *Enabling Active Flow Manipulation in Silicon-Based Network Forwarding Engines.* Lavian T.; Wang P.; Travostino F.; Subramanian S.; Duraraj R.; Hoang D.B.; Sethaput V.; Culler D.; Proceeding of the Active Networks Conference and Exposition, 2002.(DANCE) 29-30 May 2002, pp. 65-76.
- *Practical Active Network Services within Content-Aware Gateways.* Subramanian S.; Wang P.; Durairaj R.; Rasimas J.; Travostino F.; Lavian T.; Hoang D.B.; Proceeding of the DARPA Active Networks Conference and Exposition, 2002.(DANCE) 29-30 May 2002, pp. 344-354.
- *Active Networking on a Programmable Network Platform.* Wang P.Y.; Lavian T.; Duncan R.; Jaeger R.; Fourth IEEE Conference on Open Architectures and Network Programming (OPENARCH), Anchorage, April 2002.
- *Intelligent Network Services through Active Flow Manipulation.* Lavian T.; Wang P.; Travostino F.; Subramanian S.; Hoang D.B.; Sethaput V.; IEEE Intelligent Networks 2001 Workshop (IN2001), Boston, May 2001.
- Intelligent Network Services through Active Flow Manipulation. Lavian T.; Wang P.; Travostino F.; Subramanian S.; Hoang D.B.; Sethaput V.; Intelligent Network Workshop, 2001 IEEE 6-9 May 2001, pp.73 - 82.
- *Enabling Active Flow Manipulation in Silicon-based Network Forwarding Engine.* Lavian, T.; Wang, P.; Travostino, F.; Subramanian S.; Hoang D.B.; Sethaput V.; Culler D.; Journal of Communications and Networks, March 2001, pp.78-87.
- *Active Networking on a Programmable Networking Platform.* Lavian T.; Wang P.Y.; IEEE Open Architectures and Network Programming, 2001, pp. 95-103.
- *Enabling Active Networks Services on a Gigabit Routing Switch.* Wang P.; Jaeger R.; Duncan R.; Lavian T.; Travostino F.; 2nd Workshop on Active Middleware Services, 2000.
- *Dynamic Classification in Silicon-Based Forwarding Engine Environments.* Jaeger R.; Duncan R.; Travostino F.; Lavian T.; Hollingsworth J.; Selected Papers. 10th IEEE Workshop on Metropolitan Area and Local Networks, 1999. 21-24 Nov. 1999, pp.103-109.

- *Open Programmable Architecture for Java-Enabled Network Devices*. Lavian, T.; Jaeger, R. F.; Hollingsworth, J. K.; IEEE Hot Interconnects Stanford University, August 1999, pp. 265-277.
- *Open Java SNMP MIB API*. Rob Duncan, Tal Lavian, Roy Lee, Jason Zhou, Bay Architecture Lab Technical Report TR98-038, December 1998.
- *Java-Based Open Service Interface Architecture*. Lavian T.; Lau S.; BAL TR98-010 Bay Architecture Lab Technical Report, March 1998.
- *Parallel SIMD Architecture for Color Image Processing*. Lavian T. Tel – Aviv University, Tel – Aviv, Israel, November 1995.
- *Grid Network Services, Draft-ggf-ghpn-netservices-1.0.*George Clapp, Tiziana Ferrari, Doan B. Hoang, Gigi Karmous-Edwards, Tal Lavian, Mark J. Leese, Paul Mealor, Inder Monga, Volker Sander, Franco Travostino, Global Grid Forum(GGF).
- *Project DRAC: Creating an applications-aware network.*Travostino F.; Keates R.; Lavian T.; Monga I.; Schofield B.; Nortel Technical Journal, February 2005, pp. 23-26.
- *Optical Network Infrastructure for Grid, Draft-ggf-ghpn-opticalnets-1.*Dimitra Simeonidou, Reza Nejabati, Bill St. Arnaud, Micah Beck, Peter Clarke, Doan B. Hoang, David Hutchison, Gigi Karmous-Edwards, Tal Lavian, Jason Leigh, Joe Mambretti, Volker Sander, John Strand, Franco Travostino, Global Grid Forum(GGF) GHPN Standard GFD-I.036 August 2004.
- *Popeye - Using Fine-grained Network Access Control to Support Mobile Users and Protect Intranet Hosts.*Mike Chen, Barbara Hohlt, Tal Lavian, December 2000.

## Presentations and Talks

*(not an exhaustive list)*

- Lambda Data Grid: An Agile Optical Platform for Grid Computing and Data-intensive Applications.
- Web Services and OGSA
- WINER Workflow Integrated Network Resource Orchestration.
- Technology & Society.
- Abundant Bandwidth and how it affects us?
- Active Content Networking(ACN).
- DWDM-RAM:Enabling Grid Services with Dynamic Optical Networks .
- Application-engaged Dynamic Orchestration of Optical Network Resources .
- A Platform for Data Intensive Services Enabled by Next Generation Dynamic Optical Networks .
- Optical Networks.
- Grid Optical Network Service Architecture for Data Intensive Applications.
- Optical Networking & DWDM.
- OptiCal Inc.
- OptiCal & LUMOS Networks.
- Optical Networking Services.
- Business Models for Dynamically Provisioned Optical Networks.
- Business Model Concepts for Dynamically Provisioned Optical Networks.
- Optical Networks Infrastructure.
- Research Challenges in agile optical networks.
- Services and Applications' infrastructure for agile optical networks.
- Impact on Society.
- TeraGrid Communication and Computation.
- Unified Device Management via Java-enabled Network Devices.
- Active Network Node in Silicon-Based L3 Gigabit Routing Switch.
- Active Nets Technology Transfer through High-Performance Network Devices.
- Programmable Network Node: Applications.
- Open Innovation via Java-enabled Network Devices.
- Practical Considerations for Deploying a Java Active Networking Platform.
- Open Java-Based Intelligent Agent Architecture for Adaptive Networking Devices.
- Java SNMP Oplet.
- Open Distributed Networking Intelligence: A New Java Paradigm.
- Open Programmability.
- Active Networking On A Programmable Networking Platform.
- Open Networking through Programmability.
- Open Programmable Architecture for Java-enabled Network Devices.

- Integrating Active Networking and Commercial-Grade Routing Platforms.
- Programmable Network Devices.
- To be smart or not to be?