(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
13 November 2003 (13.11.2003)

PCT

(10) International Publication Number
WO 03/094041 A1

(51) International Patent Classification⁷: G06F 17/00

(21) International Application Number: PCT/US03/13586

(22) International Filing Date: 30 April 2003 (30.04.2003)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/376,773 1 May 2002 (01.05.2002) US
10/404,865 1 April 2003 (01.04.2003) US

(71) Applicant: BEA SYSTEMS, INC. [US/US]; 2315 North First Street, San Jose, CA 95131 (US).

(72) Inventors: WISER, David; 6392 Slida Drive, San Jose, CA 95129 (US). DALAL, Sanjay; 575 E. Remington Drive, #12M, Sunnyvale, CA 94087 (US). HOEBANX, Pascal; 2896 Pruneridge Avenue, Santa Clara, CA 95051 (US).

(74) Agent: MEYER, Sheldon, R.; Fliesler Dubb Meyer & Lovejoy LLP, Four Embarcadero Center, Suite 400, San Francisco, CA 94111-4156 (US).

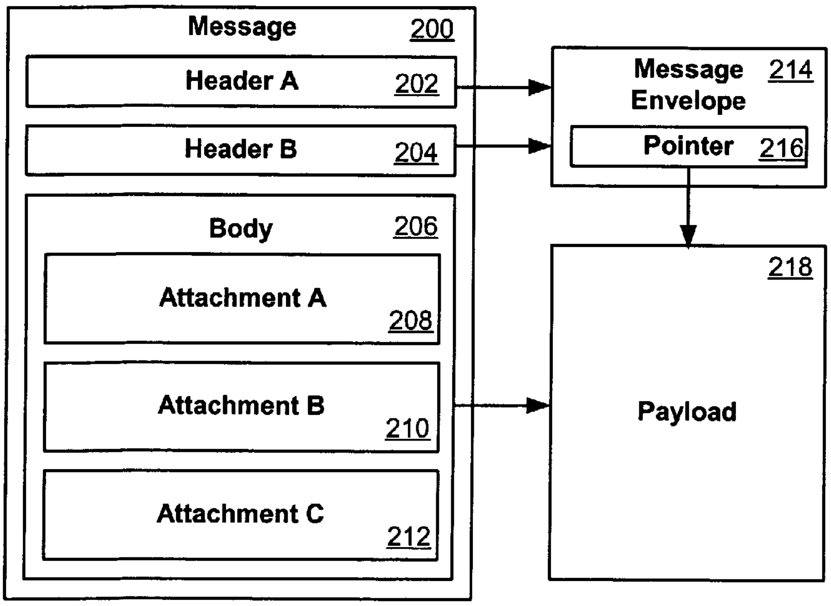
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NI, NO, NZ, OM, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

[Continued on next page]

(54) Title: SYSTEM AND METHOD FOR STORING LARGE MESSAGES



(57) Abstract: A large message (200) can be stored by separating the message (200) into an envelope portion containing information such as headers (202, 204), protocols, and addresses, and a payload portion (218) containing items such as file attachments. The envelope portion (214) can be stored in local storage, while the payload (218) can be stored to a persistent store. The message (200) can be processed incrementally, such that the entire message (200) is never in system memory. Once the envelope portion (214) is processed, the payload portion (218) can be read in increments without being processed, and those increments written directly to the persistent store. Alternatively, the payload (218) can be streamed to the persistent store. A pointer (216) in the envelope (214) can then be used to locate and retrieve attachments (208, 210, 212) from persistent storage.

System and Method for Storing Large Messages

CLAIM OF PRIORITY

This application claims priority to U.S. Provisional Patent Application No. 60/376,773, filed May 1, 2002, entitled "System and Method for Storing Large Messages," which is hereby incorporated herein by reference.

COPYRIGHT NOTICE

A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document of the patent disclosure, as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.

CROSS-REFERENCED CASES

The following applications are cross-referenced and incorporated herein by reference:

U.S. Patent Application No. 10/404,552 filed April 1, 2003, to Mike Blevins et al. and entitled, "COLLABORATIVE BUSINESS PLUG-IN FRAMEWORK";

U.S. Patent Application No. 10/404,684 filed April 1, 2003, to Mike Blevins et al. and entitled, "SYSTEMS AND METHODS FOR BUSINESS PROCESS PLUG-IN DEVELOPMENT"; and

U.S. Patent Application No. 10/404,666 filed April 1, 2003, to David Wiser et al. and entitled "Single Servlets for B2B Message Routing."

FIELD OF THE INVENTION

The present invention relates to the storage of large messages in a computer system or on a computer network.

BACKGROUND

Existing integration and messaging systems have problems handling large messages. Incoming messages are read into memory in their entirety, such that when a number of large messages are received a system can crash due to a lack of available memory. Some systems try to prevent these problems by limiting the size of messages that can be processed through a system, but this approach is undesirable to users needing to send messages that may occasionally exceed that limitation.

Another existing approach utilizes in-database persistence and in-memory caching on a hub. Persistence saves enough data for recovery purposes, and caching allows messages to be serialized to a Java Message Service (JMS). This allows JMS to enqueue faster, and allows a JMS dequeue to request the message from a cache without having to redo expensive operations like deserialization, decryption, and XML parsing. The problem still exists in that it is necessary to read an entire message into memory in order to process the message.

BRIEF SUMMARY

Systems and methods in accordance with embodiments of the present invention can overcome deficiencies in existing messaging systems by changing the way in which messages are processed and stored. An integration component can receive an incoming message, such as from a Web server. The integration component can separate the message into an "envelope" portion, which can contain information such as headers, protocols, and addresses, and a "payload" portion, which can contain items such as file attachments. The integration component can write the envelope portion to local memory, and can write the payload portion to at least one persistent store. A pointer can be placed in the envelope to identify the location of the payload in the persistent store. Applications can then use the envelope to locate the payload in a persistent store.

An integration component can also process a message incrementally. The integration component can process portions of the message until the payload portion is reached. The integration component can then stop processing the message, but can continue to read the message in increments and write those increments to a persistent store. Parsers such as MIME parsers and XML parsers can be used by the integration component to process the message. Alternatively, the integration component can process the message as a stream, or at least write the payload portion to the persistent store as a stream.

Other features, aspects, and objects of the invention can be obtained from a review of the specification, the figures, and the claims.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a diagram of a system in accordance with one embodiment of the present invention.

Figure 2 is a diagram of a message that can be processed using the system of Figure 1.

DETAILED DESCRIPTION

In systems and methods in accordance with embodiments of the present invention, "large" messages, such as large business messages in XML format, an be processed in a Web server or integration application. These business messages can be Java Message Service (JMS) messages, for example, which can utilize distributed destinations in a cluster. A large business message can be any message that may have an attachment or a large amount of text, for example, which can have an overall message size of at or above 1 MB, at or above 10MB, at or above 50 MB, or even at or above 100MB. For example, company A can send a message to company B that has a file size of 100 MB. The integration system receiving that message will have to process and resend the entire message. In existing systems, it is necessary to read the entire message into memory before writing the message to disk. The read and write are each done in one complete step. Present systems also have to parse the entire message.

As shown in **Figure 1**, when a message from a company A **100** is first received to an integration component **104** from a Web server **102**, the message can be read into local memory **106**. Local memory can be any appropriate storage medium, such as may be located on the Web server itself, in a cluster containing the Web server, or on a network node accessible to the Web server. If several large messages are received by the Web server **102**, the server may eventually run out of memory. In a system in accordance with one embodiment of the present invention, portions of the body of each message can be stored in persistent storage **108** instead of being completely stored in local memory **106**. There are at least two types of persistent storage, including file-based persistence stores and data-based persistence stores. Continuing with the example, company B **110** can be working with an integration application. When the message arrives at the Web server **102** for company B **110**, the message can arrive on a socket on the network. Portions of the message can be stored somewhat directly to the persistent message store **108** instead of being read entirely into local memory **106**. One way to do this is to read the message in increments, or small portions, and write those small portions to storage. For example, the 100 MB message could have a 4 MB portion read into local memory **106**, then have that 4 MB portion written to persistent storage **108**. Then another 4 MB portion could be read into local memory and written to persistent storage. This process could continue until the entire body portion of the message is in persistent storage **108**. Although portions of the entire message may be in memory at one point or another, there would only be up to 4 MB of the message in local memory at any given time. The user can configure the persistent store **108** so that the message is sent to a file or to a database, for example. The portion size can be any size appropriate for the size of the message or the capacity of the system, such as portions of 1 MB, 5 MB, or 10 MB. The portion size can also be a percentage of the overall file size, such as 1%, 5%, 10%, or 25%, for example.

When a message is processed using an integration application or integration server, for example, the message can use a storage method referred to herein as "envelope plus payload." The message can be processed in the server to separate the contents to be placed in the "envelope" from contents to be placed in the "payload." This is shown, for example, in the diagram of **Figure 2**. Headers **202**, **204** of a message **200** can be extracted by an integration server, as the headers may be all the server requires to process the message **200**. A header can identify the protocol under which the message is sent, such as an XOCP protocol. The protocol can be used to help identify the headers **202, 204** and the body **206** of the message. It can be important in certain systems to identify the message protocol, as protocols such as RosettaNet and ebXML have different packaging semantics than a protocol such as XOCP. The headers can be placed in the envelope **214**, which can be stored in local memory. The body **206** of the message **200**, which can contain several attachments **208, 210, 212**, for example, can be placed into the payload **218**. The payload can be stored in persistent storage on the server, in the cluster, or on the network. The envelope **214** can contain a pointer **216** to the location of the payload **218**.

Since a message can contain a body with multiple parts, the payload can be designed to contain multiple parts as well. While processing a message in the server, however, only the envelope may be needed. The payload can belong to the user of a B2B server, for example, or an application riding on top of an integration server. The payload can be stored to persistent storage, so that the full payload is never stored in memory. A server or any application can simply deal with the envelope, which can contain pointers to the payload. When an application wants to access any portion of the message, the application can view information contained in the envelope, which can include identification information for the payload parts.

An application can use any pointers in an envelope to extract portions of the body of the message stored in the payload. As the application can retrieve the data from this persistence store, it is not necessary to accumulate everything in local memory on the integration server. A message envelope can contain a pointer to the body of the message, whether there is a single message body or a number of portions, or can contain a pointer for each portion of the body in persistent storage. The number of portions can include a number of attachments, for example. It is not necessary for the integration system to process the attachments to a message, so the system can simply write the attachments to storage, either all together in one block of memory or individually. The pointer can point to the location at which a portion of the message body begins in memory, or can point to the boundaries of a given body portion in memory, for example.

An envelope can contain other useful information about a message, such as the address of the sender and/or the address of the recipient. Each of these addresses can each be a URL, for example. The envelope can also contain the protocol of the message and possibly the protocol of any body portion, if applicable. The envelope can contain message text. The envelope can also contain information about each attachment in the body, such as title, file type, and historical information.

At least two levels of parsing can be used to process a message. A low-level parsing mechanism can be used to decode transfer protocols such as MIME or UUENCODE. The low-level parser can receive the byte stream and identify the parts of the message, such as a text portion and a binary attachment. A second level of parsing, such as XML parsing, can be used to read headers and body portions, which can be in XML or another appropriate messaging or mark-up language.

A message can arrive from the Internet, for example, and can pass through the Web server into an integration transport layer. First, the message can pass through a MIME parser. Second, the message can be decoded using a second processing layer to determine the appropriate business protocol. The envelope can be created in this transport layer. In the decoding process, which can use the XML parser, the envelope can be filled with headers and other appropriate information. After the headers, a pointer can be placed in the envelope and the MIME parser can stop parsing the message. The MIME parser can know to stop parsing when it hits attachments, for example.

The remainder of the message, which can include at least a portion of the body and any attachments, can then be written directly to persistent storage, either in small data "dumps" or on a data stream. Once the entire message is processed, such that the envelope and payload are created, an application can determine where the message portions reside using pointers in the message envelope. The envelope can be thought of as an "abstract" of the message. Once a user or application gets this abstract, that user or application can extract any portion of the message that is needed. For instance, if there are three attachments, the user or application can choose to extract one or two of the attachments from the persistent storage. When the user deletes the message, the envelope can be used, such as by an integration server or B2B server, to delete the associated portions in the persistent storage.

The foregoing description of preferred embodiments of the present invention has been provided for the purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise forms disclosed. Many modifications and variations will be apparent to one of ordinary skill in the art. The embodiments were chosen and described in order to best explain the principles of the invention and its practical application, thereby enabling others skilled in the art to understand the invention for various embodiments and with various modifications that are suited to the particular use contemplated. It is intended that the scope of the invention be defined by the following claims and their equivalence.

CLAIMS

What is claimed is:

1. A system for storing a message, comprising:

a memory component adapted to temporarily store information for the message;

a persistent store adapted to persistently store information for the message; and

an integration component for receiving the message, the integration component adapted to separate the message into an envelope portion and a payload portion, the integration component adapted to store the envelope portion in the memory component and adapted to store the payload portion in the persistent store.

2. A system according to claim 1, wherein:

the integration component is adapted to separate the message into an envelope portion containing information selected from the group consisting of headers, protocols, addresses, and message text.

3. A system according to claim 1, wherein:

the integration component is adapted to separate the message into a payload portion containing message attachments.

4. A system according to claim 1, further comprising:

a Web server adapted to receive the message and direct the message to the integration component

5. A system according to claim 1, wherein:

the persistent store is selected from group consisting of file-based persistent stores and data-based persistent stores.

6. A system according to claim 1, wherein:

the integration component is further adapted to process message in increments of the overall message size.

7. A system according to claim 5, wherein:

the integration component is further adapted to write the payload to the persistent store in increments.

8. A system according to claim 1, wherein:

the integration component is further adapted to process message in increments each having a size selected from the group consisting of 1 MB, 5 MB, or 10 MB.

9. A system according to claim 1, wherein:

the integration component is further adapted to process message in increments each having a size selected from the group consisting of 1%, 5%, 10%, and 25% of the overall file size.

10. A system according to claim 1, wherein:

the integration component is selected from the group consisting of integration applications, integration servers, integration transport layers, and integration plug-ins.

11. A system according to claim 1, wherein:

the integration component is further adapted to place a pointer in the envelope, the pointer indicating the location of the payload portion in the persistent store.

12. A system according to claim 1, wherein:

the integration component is adapted to store the payload portion in the persistent store, wherein the payload is stored in multiple locations in the persistent store.

13. A system according to claim 1, wherein:

the integration component is adapted to store the payload portion in the persistent store and any other persistent store, wherein the payload portion can be stored to more than one persistent store.

14. A system according to claim 1, further comprising:

a protocol parser adapted to identify portions of the message to the integration component.

15. A system according to claim 14, wherein:

the protocol parser is further adapted to stop parsing the protocol parser when reaches payload portion of the message.

16. A system according to claim 1, further comprising:

an XML parser for reading header and body portions of the message.

17. A system according to claim 1, wherein:

the integration component is further adapted to store the payload portion to the persistent store in a stream without processing the payload portion.

18. A system according to claim 1, wherein:

the integration component is adapted to receive messages in XML.

19. A system according to claim 1, wherein:

the integration component is adapted to receive messages having a file size selected from the group consisting of at or above 1MB, at or above 10 MB, at or above 50 MB, and at or above 100 MB.

20. A method for storing a message, comprising:

receiving the message to an integration component;

separating the message into an envelope portion and a payload portion;

storing the envelope portion to local storage; and

storing the payload to at least one persistent store.

21. A method according to claim 20, further comprising:

parsing the message with a protocol parser to identify portions of the message to the integration component.

22. A method according to claim 20, further comprising:

parsing the message with an XML parser to read header and body portions of the message.

23. A method according to claim 20, further comprising:

processing the message in increments.

24. A method according to claim 20, further comprising:

processing the message as a stream.

25. A method according to claim 20, further comprising:

placing a pointer in the envelope portion to identify the location of the payload portion in the persistent store.

26. A computer-readable medium, comprising:

means for receiving the message to an integration component;

means for separating the message into an envelope portion and a payload portion;

means for storing the envelope portion to local storage; and

means for storing the payload to at least one persistent store.

27. A computer program product for execution by a server computer for storing a message, comprising:

computer code for receiving the message to an integration component;

computer code for separating the message into an envelope portion and a payload portion;

computer code for storing the envelope portion to local storage; and

computer code for storing the payload to at least one persistent store.

28. A system for storing a message, comprising:

means for receiving the message to an integration component;

means for separating the message into an envelope portion and a payload portion;

means for storing the envelope portion to local storage; and

means for storing the payload to at least one persistent store.

29. A computer system comprising: a processor;

object code executed by said processor, said object code configured to:

receive the message to an integration component;

separate the message into an envelope portion and a payload portion;

store the envelope portion to local storage; and

store the payload to at least one persistent store.

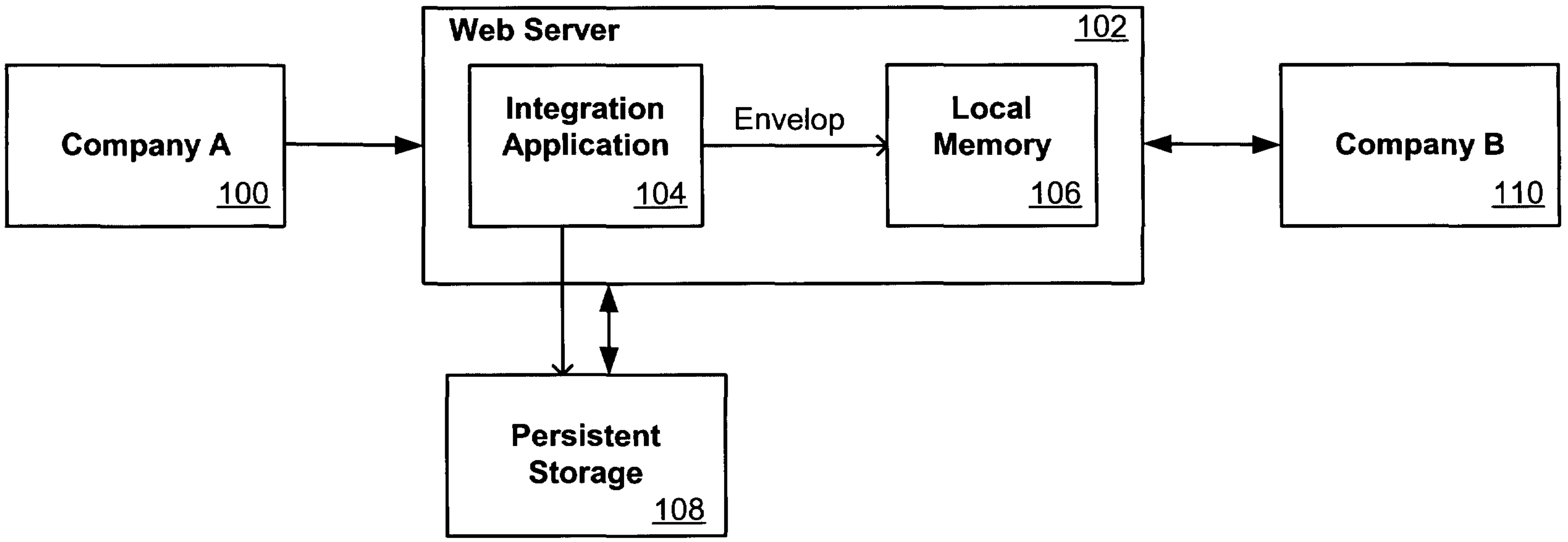

Company A
100
Web Server
102
Integration Application
104
Envelop
Local Memory
106
Company B
110
Persistent Storage
108

Figure 1

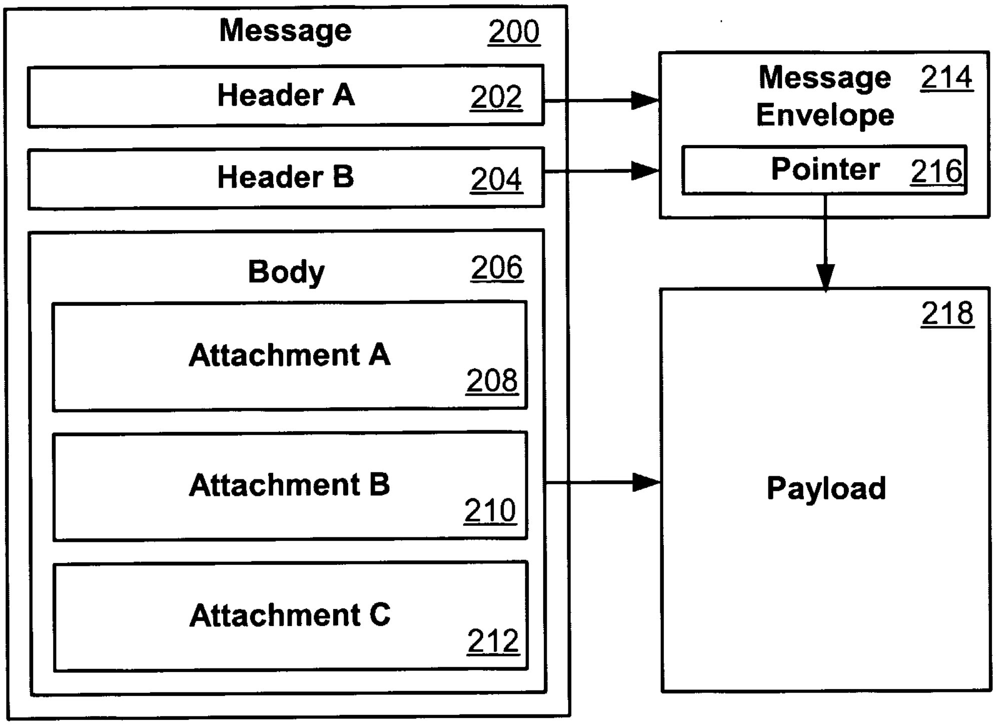

Message 200
Header A 202
Header B 204
Body 206
Attachment A 208
Attachment B 210
Attachment C 212
Message Envelope 214
Pointer 216
218
Payload

Figure 2

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US03/13586

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) :G06F 17/00
US CL : 707/101
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 707/1-10. 100, 101-104.1, 200-206

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

WEST search terms: XML, markup language, message, size, network

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 6,345,283 B1 (ANDERSON) 05 FEBRUARY 2002, ABSTRACT	1-29
A	US 6,348,970 B1 (MARX) 19 FEBRUARY 2002, ABSTRACT	1-29
A	US 6,360,221 B1 (GOUGH ET AL.) 19 MARCH 2002, ABSTRACT	1-29

□ Further documents are listed in the continuation of Box C. □ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

18 JULY 2003

Date of mailing of the international search report

12 AUG 2003

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231

Facsimile No. (703) 305-3230

Authorized officer
Peggy Harrod
DAVID Y. JUNG

Telephone No. (703) 308-5262