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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
22 March 2001 (22.03.2001)

PCT

(10) International Publication Number
WO 01/19159 A2

(51) International Patent Classification: Not classified

(21) International Application Number: PCT/GB99/04415

(22) International Filing Date:
23 December 1999 (23.12.1999)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
9921720.0 14 September 1999 (14.09.1999) GB

(71) Applicant *(for all designated States except US)*: **TAO GROUP LIMITED** [GB/GB]; 62/63 Suttons Business Park, Earley, Reading, Berkshire RG6 1AZ (GB).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: **RENOUF, Timothy** [GB/GB]; 20 Chesterfield Road, Newbury, Berks RG14 7QB (GB).

(74) Agents: **MAGGS, Michael, Norman** et al.; Kilburn & Strode, 20 Red Lion Street, London WC1R 4PJ (GB).

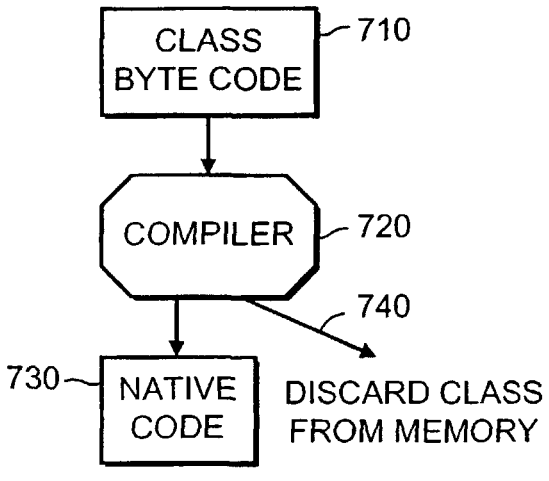
(81) Designated States *(national)*: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL,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

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

Published:
— *Without international search report and to be republished upon receipt of that report.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: LOADING OBJECT-ORIENTED COMPUTER PROGRAMS



(57) Abstract: The method of loading an object-oriented computer program comprises translating the program into compiled code and holding that code, ready for execution, in an execution environment which does not necessarily also hold the original program code. In a preferred embodiment, the original class file is discarded from memory prior to execution of the program. The invention is particularly applicable to wireless communications networks consisting of a plurality of client devices, such as mobile phones, each of which typically has only a small amount of memory available. Using the present invention, the original class file is discarded after compilation, and does not need to be retained in the client device memories.

WO 01/19159 A2

LOADING OBJECT-ORIENTED COMPUTER PROGRAMS

The invention relates generally to methods of and computer systems for loading object-oriented computer programs. More specifically, although not exclusively the invention is concerned with object-oriented programs in which the code is provided in the form of class files, each containing a plurality of methods.

A well known example of an object-oriented programming language is "Java" (a trade mark of Sun Microsystems Inc.). A "Java implementation" is a software system which allows a software application consisting of one or more class files to be run. These class files must conform to some version of the standard Java Virtual Machine Specification, as published by Sun Microsystems Inc. A class file defines the data and program code required for a particular class.

Although there is some interaction, the Java implementation can conceptually be divided into two parts:

- The Java Virtual Machine (JVM). This reads the class files and executes the instructions contained within them in a way which conforms to some version of Sun's Java Virtual Machine Specification. Some of the instructions contained within a class may reference the data or program code of other classes; the JVM also manages such relationships between classes.
- The Java Class Library. This is a set of predefined classes with data and program code which act in a way conforming to Sun's Java Class Library Specification. These library classes could be implemented in some way other than as real class files, for example using the C or assembler programming languages, in which case the JVM must ensure that references