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

(19) World Intellectual Property Organization
International Bureau

PCT

(43) International Publication Date
3 April 2008 (03.04.2008)

(10) International Publication Number
WO 2008/038265 A4

(51) **International Patent Classification:**
*G06F 9144* **(2006.01)** *G06F 19/00* **(2006.01)**

(21) **International Application Number:**
**PCT/IL2007/001152**

**(22) International Filing Date:**
20 September 2007 (20.09.2007)

**(25) Filing Language:** English

**(26) Publication Language:** English

**(30) Priority Data:**
60/826,749 25 September 2006 (25.09.2006) US

(71) **Applicant** *(for all designated States except US):* **TYPEMOCK LTD.** [IL/IL]; 17 Mapu Street, 63435 Tel Aviv (IL).

**(72) Inventor; and**
**(75) Inventor/Applicant** *(for US only):* **LOPIAN, Eli** [IL/IL]; 17 Mapu Street, 63435 Tel Aviv (IL).

**(74) Agent: REINHOLD COHN AND PARTNERS;** Ro.b. 4060, 61040 Tel-aviv (IL).

(81) **Designated States** *(unless otherwise indicated, for every kind of national protection available):* AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KR KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW

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

**Declaration under Rule** 4.17:
— *of inventorship (Rule 4.17(iv))*

**Published:**
— *with international search report*
— *with amended claims*

**(88) Date of publication of the international search report:**
18 December 2008

**Date of publication of the amended claims:** 5 February 2009

WO 2008/038265 A4

**(54) Title:** METHOD AND SYSTEM FOR ISOLATING SOFTWARE COMPONENTS

**(57) Abstract:** A software testing system operative to test a software application comprising a plurality of software components, at least some of which are highly coupled hence unable to support a dependency injection, each software component operative to perform a function, the system comprising apparatus for at least partially isolating, from within the software application, at least one highly coupled software component which performs a given function, and apparatus for testing at least the at least partially isolated highly coupled software component.

AMENDED CLAIMS
received by the International Bureau on 06 OCTOBER 2008

1. A software testing system operative to test a software application comprising a plurality of software components, at least some of which are highly coupled hence unable to support a dependency injection, each software component operative to perform a function, the software application lacking a built-in byte-code modification functionality, the system comprising:

apparatus for at least partially isolating, from within the software application, at least one highly coupled software component which performs a given function, without utilizing built-in byte-code modification functionality; and

apparatus for testing logic of at least said at least partially isolated highly coupled software component, without dependency injection.

2. A system according to claim 1 wherein said highly coupled software component is operative to call at least one additional software component and wherein said apparatus for testing is operative to test whether said at least partially isolated highly coupled software component calls said at least one additional software component correctly.

3. A system according to claim 1 wherein the plurality of software components comprises a set of at least one pairs of utilizing-utilized software components each including a utilized software component and a utilizing software component which utilizes said utilized software component, said apparatus for at least partially isolating comprises apparatus for adding access controlling code between each pair of utilizing-utilized software components, said access controlling code being operative to control access of the utilizing software component to the utilized software component.

4. A system according to claim 3 wherein said set comprises at least one utilizing software component which calls its corresponding utilized software component.

5. A system according to claim 3 wherein said set comprises at least one utilizing software component which accesses at least one data element belonging to its corresponding utilized software component.

6. A system according to claim 3 wherein said utilizing software component comprises test code and wherein said access controlling code is operative to generate a plurality of testing scenarios for said test code by suitably controlling access of the test code to the utilized software component.

7. A system according to claim 3 wherein the software application comprises at least one source file and wherein said apparatus for adding code is operative, before compilation of the software application, to add the access controlling code to the at least one source file thereby to provide, upon compilation of the source file, an at least partially isolatable weaved application.

8. A system according to claim 3 wherein the software application is stored in at least one executable file and wherein said apparatus for adding code comprises apparatus for parsing said at least one executable file, adding said access controlling code to the parsed executable file, and saving, thereby to provide an at least partially isolatable weaved executable file.

9. A system according to claim 3 wherein said apparatus for adding code is operative, after the software application has been loaded into directly accessible memory by an operating system from an executable file on disk and before the software application has been run, to parse the software application and add said access controlling code to the parsed software application, thereby to provide an at least partially isolatable weaved application.

10. A system according to claim 1 wherein the plurality of software components comprises a set of at least one pairs of utilizing-utilized software components each including a utilized software component and a utilizing software component which utilizes said utilized software component, and wherein said apparatus for at least partially isolating comprises access controlling code external of the software application for anticipating forthcoming utilization of utilized software components by utilizing software components and for selectively preventing said

utilization by controlling access of the utilizing software component to the utilized software component.

11. A system according to claim 1 wherein said apparatus for at least partially isolating is operative, upon occurrence of a call by a first component from among the plurality of software components to a second component from among the plurality of software components, to intervene to ensure mat the second component does not run.

12. A system according to claim 11 wherein said call, absent operation of said apparatus for at least partially isolating, results in data being returned by the second component to the first component, and wherein said apparatus for at least partially isolating is operative, instead, to inject fake data into the first component.

13. A system according to claim 11 wherein said apparatus for at least partially isolating is operative to fake a failure of the second component.

14. A system according to claim 3 or claim 10 wherein said access controlling code is controlled by an application-specific test protocol.

15. A system according to claim 1 wherein said apparatus for at least partially isolating is operative, upon occurrence of a call by a first component from among the plurality of software components to a second component from among the plurality of software components, the second component operating upon at least one argument, to intervene by providing the second component with at least one fake argument.

16. A system according to claim 3 or claim 10 wherein at least one utilized software component comprises initialization code and wherein said apparatus for testing is operative to record instances of at least partial isolation of said initialization code and, if said utilized software component is called subsequent to termination of said at least partial isolation of the initialization code, to artificially execute said initialization code.

17. A system according to claim 14 wherein said access controlling code is controlled by the protocol via the apparatus for testing.

18. A system according to claim 1 wherein said apparatus for testing is operative to select at least one software component for said apparatus for at least partially isolating to at least partially isolate from within the software application.

19. A system according to claim 1 wherein said apparatus for testing is operative to generate a plurality of expectations each of which comprises an identity of an individual component from among the plurality of software components and an associated behavior inducing message inducing said apparatus for at least partially isolating, when said individual component is called, to selectively at least partially isolate, and to impose a fake behavior upon, the individual component.

20. A system according to claim 19 wherein the fake behavior imposed upon the individual component, when called, in accordance with contents of the associated behavior inducing message, comprises one of the following:

preventing the called component from running and, if results are to be returned, returning fake results;

faking a failure of the called component;

providing a fake argument to the called component; and

none of the above.

21. A system according to claim 19 and also comprising access controlling code, weaved into at least one location in said software application, which is operative to query said apparatus for testing as to which operation, if any, said apparatus for at least partially isolating is to perform, to receive an expectation, responsively, from the apparatus for testing, and to activate said apparatus for at least partially isolating accordingly.

22. A system according to claim 19 wherein at least one of said expectations also comprises an indication of circumstances under which said individual component is to act upon said behavior inducing message.

23. A system according to claim 19 wherein at least an individual one of said expectations also comprises an indication of at least one expected arguments which are expected to be passed to said individual component.

24. A system according to claim 19 wherein said identity of an individual component comprises a string identifying the component and stored within the expectation.

25. A system according to claim 19 wherein at least one expectation is generating by recording an actual call to at least said individual component.

26. A system according to claim 23 wherein said individual expectation is generating by recording an actual call to said individual component, in the course of which call at least one actual argument is actually passed to said individual component, and wherein the indication of at least one argument in said expectation comprises said at least one actual argument.

27. A system according to claim 1 wherein the plurality of software components comprises a set of at least one pairs of utilizing-utilized software components each including a utilized software component and a utilizing software component which utilizes said utilized software component and which includes meta-data pointing to the utilized software component, and said apparatus for at least partially isolating comprises apparatus for modifying said meta-data to point to access control code, said access controlling code being operative to control access of the utilizing software component to the utilized software component.

28. A system according to claim 11 wherein the second component has yet to be created.

29. A system according to claim 25 wherein said call comprises a chain of n calls and wherein n expectations are generated by recording said chain of n calls.

30. A system according to claim 23 wherein said apparatus for testing is operative, when said individual component is called with at least one actual arguments, to test said actual arguments in comparison to said expected arguments.

31. A software testing system operative to test a software application comprising a plurality of software components, at least some of which are highly coupled hence unable to support a dependency injection, each software component

operative to perform a function, the software application lacking a built-in byte-code modification functionality, the system comprising:

apparatus for at least partially isolating, from within the software application, at least one highly coupled software component which performs a given function, without utilizing built-in byte-code modification functionality; and

apparatus for testing logic of at least said at least partially isolated highly coupled software component, without dependency injection,

wherein said apparatus for testing is operative to generate a plurality of expectations each of which comprises an identity of an individual component from among the plurality of software components and an associated behavior inducing message inducing said apparatus for at least partially isolating, when said individual component is called, to selectively at least partially isolate, and to impose a fake behavior upon, the individual component.

32. A system according to claim 31 wherein said highly coupled software component is operative to call at least one additional software component and wherein said apparatus for testing is operative to test whether said at least partially isolated highly coupled software component calls said at least one additional software component correctly.

33. A system according to claim 31 wherein the plurality of software components comprises a set of at least one pairs of utilizing-utilized software components each including a utilized software component and a utilizing software component which utilizes said utilized software component, said apparatus for at least partially isolating comprises apparatus for adding access controlling code between each pair of utilizing-utilized software components, said access controlling code being operative to control access of the utilizing software component to the utilized software component.

34. A system according to claim 33 wherein said set comprises at least one utilizing software component which calls its corresponding utilized software component.

35. A system according to claim 33 wherein said set comprises at least one utilizing software component which accesses at least one data element belonging to its corresponding utilized software component.

36. A system according to claim 33 wherein said utilizing software component comprises test code and wherein said access controlling code is operative to generate a plurality of testing scenarios for said test code by suitably controlling access of the test code to the utilized software component.

37. A system according to claim 33 wherein the software application comprises at least one source file and wherein said apparatus for adding code is operative, before compilation of the software application, to add the access controlling code to the at least one source file thereby to provide, upon compilation of the source file, an at least partially isolatable weaved application.

38. A system according to claim 33 wherein the software application is stored in at least one executable file and wherein said apparatus for adding code comprises apparatus for parsing said at least one executable file, adding said access controlling code to the parsed executable file, and saving, thereby to provide an at least partially isolatable weaved executable file.

39. A system according to claim 33 wherein said apparatus for adding code is operative, after the software application has been loaded into directly accessible memory by an operating system from an executable file on disk and before the software application has been run, to parse the software application and add said access controlling code to the parsed software application, thereby to provide an at least.partially isolatable weaved application.

40. A system according to claim 31 wherein the plurality of software components comprises a set of at least one pairs of utilizing-utilized software components each including a utilized software component and a utilizing software

component which utilizes said utilized software component, and wherein said apparatus for at least partially isolating comprises access controlling code external of the software application for anticipating forthcoming utilization of utilized software components by utilizing software components and for selectively preventing said utilization by controlling access of the utilizing software component to the utilized software component.

41. A system according to claim 31 wherein said apparatus for at least partially isolating is operative, upon occurrence of a call by a first component from among the plurality of software components to a second component from among the plurality of software components, to intervene to ensure that the second component does not run.

42. A system according to claim 41 wherein said call, absent operation of said apparatus for at least partially isolating, results in data being returned by the second component to the first component, and wherein said apparatus for at least partially isolating is operative, instead, to inject fake data into the first component.

43. A system according to claim 41 wherein said apparatus for at least partially isolating is operative to fake a failure of the second component.

44. A system according to claim 33 or claim 40 wherein said access controlling code is controlled by an application-specific test protocol.

45. A system according to claim 31 wherein said apparatus for at least partially isolating is operative, upon occurrence of a call by a first component from among the plurality of software components to a second component from among the plurality of software components, the second component operating upon at least one argument, to intervene by providing the second component with at least one fake argument.

46. A system according to claim 33 or claim 40 wherein at least one utilized software component comprises initialization code and wherein said apparatus for testing is operative to record instances of at least partial isolation of said initialization code and, if said utilized software component is called subsequent to termination of

said at least partial isolation of the initialization code, to artificially execute said initialization code.

47. A system according to claim 44 wherein said access controlling code is controlled by the protocol via the apparatus for testing.

48. A system according to claim 31 wherein said apparatus for testing is operative to select at least one software component for said apparatus for at least partially isolating to at least partially isolate from within the software application.

49. A system according to claim 31 wherein the plurality of software components comprises a set of at least one pairs of utilizing-utilized software components each including a utilized software component and a utilizing software component which utilizes said utilized software component and which includes meta¬data pointing to the utilized software component, and said apparatus for at least partially isolating comprises apparatus for modifying said meta-data to point to access control code, said access controlling code being operative to control access of the utilizing software component to the utilized software component.

50. A system according to claim 41 wherein the second component has yet to be created.

51. A system for testing a software component of a software application, the system comprising:

a dynamic isolator module adapted to dynamically and at least partially isolate the software component from the software application.

52. A software testing method operative to test a software application comprising a plurality of software components, at least some of which are highly coupled hence unable to support a dependency injection, each software component operative to perform a function, the method comprising:

at least partially isolating, from within the software application, at least one highly coupled software component which performs a given function; and

testing at least said at least partially isolated highly coupled software component.

53. A method for testing a software component of a software application, the method comprising:

dynamically and at least partially isolating the software component from the software application.

54. A system according to claim 51, the software application lacking a built-in byte-code modification functionality, the module comprising:

apparatus for at least partially isolating, from within the software application, at least one highly coupled software component which performs a given function, without utilizing built-in byte-code modification functionality, the system also comprising apparatus for testing logic of at least said at least partially isolated highly coupled software component, without dependency injection.

55. A system according to claim 52, said isolating comprising at least partially isolating, from within the software application, at least one highly coupled software component which performs a given function, without utilizing built-in byte-code modification functionality, said testing comprising testing logic of at least said at least partially isolated highly coupled software component, without dependency injection.

56. A method according to claim 53, said isolating comprising at least partially isolating, from within the software application, at least one highly coupled software component which performs a given function, without utilizing built-in byte-

code modification functionality, said method also comprising testing logic of at least said at least partially isolated highly coupled software component, without dependency injection.

57. A system according to claim 51 and also comprising apparatus for testing operative to generate a plurality of expectations each of which comprises an identity of the software component and an associated behavior inducing message inducing said module, when said component is called, to selectively at least partially isolate, and to impose a fake behavior upon, the component.

58. A method according to claim 52 wherein said testing is operative to generate a plurality of expectations each of which comprises an identity of an individual component from among the plurality of software components and an associated behavior inducing message operative, when said individual component is called, to induce an operation of selectively at least partially isolating, and imposing a fake behavior upon, the individual component.

59. A method according to claim 53 and also comprising testing to generate a plurality of expectations each of which comprises an identity of the software component and an associated behavior inducing message operative, when said component is called, to induce an operation of selectively at least partially isolating, and imposing a fake behavior upon, the individual component.