CLAIMS

1. An apparatus for testing applications, said apparatus including: a data-processing machine including a memory and a processor operably coupled to said memory, said data-processing machine having been configured to implement a data-driven testing framework that includes a data engineering module, a computational environment manager, and a result analysis module,

wherein said data engineering module is configured to create engineered test data based at least in part on said application to be tested,

wherein said computational environment manager is configured to control a computational environment in which said application is to operate on said engineered test data, and

wherein said result analysis module is configured to compare engineered test data operated upon by said application with an expected output.

2. The apparatus of claim 1,

wherein said data engineering module is configured to extract a subset of production data,

wherein said subset is selected to achieve a specified code coverage, and

wherein said engineered test data includes said subset of said production data.

3. The apparatus of claim 1, wherein said data engineering module includes a data still for generating distilled data from production data.

4. The apparatus of claim 1, wherein said data engineering module is configured to extract a subset of production data, wherein said data engineering module is further configured to augment said subset with additional data, thereby generating augmented data, wherein said additional data is selected to achieve specified code coverage, and wherein said engineered test data includes said augmented data.

5. The apparatus of claim 1, wherein said data engineering module includes a data enhancer for receiving distilled data from said data still and enhancing said distilled data.

6. The apparatus of claim 1, wherein said data engineering module is configured to generate data based at least in part on said application to be tested, wherein said generated data is selected to achieve specified code coverage, and wherein said engineered test data includes said generated data.

7. The apparatus of claim 1, wherein said data engineering module further includes a positive-data manufacturer for generating positive data.

8. The apparatus of claim 1, wherein said data engineering module is configured to generate data based at least in part on said application to be

tested, and wherein said data is absent from production data.

9. The apparatus of claim 1, wherein said data engineering module further includes a negative-data manufacturer for generating negative data.

10. The apparatus of claim 1, wherein said data engineering module includes means for generating engineered test data.

11. The apparatus of claim 1, wherein said data engineering module includes an integrity checker for determining referential integrity of said engineered test data.

12. The apparatus of claim 1, wherein said data engineering module is further configured to detect errors in referential integrity.

13. The apparatus of claim 1, wherein said data engineering module includes a re-referencer for correcting a loss of referential integrity in data prior to outputting said data as engineered test data.

14. The apparatus of claim 1, wherein said data engineering module is further configured to correct a loss of referential integrity in data.

15. The apparatus of claim 1, wherein said data engineering module includes an inspection unit for receiving said engineered test data and enabling a user to at least one of view said

engineered test data and profile said engineered test data.

16. The apparatus of claim 1, wherein said data engineering module includes a data-inspection unit for receiving said engineered test data and enabling a user to view said engineered test data.

17. The apparatus of claim 1, wherein said data engineering module includes a profiler for receiving said engineered test data and enabling a user to profile said engineered test data.

18. The apparatus of claim 1, wherein said data engineering module is further configured to enable a user to profile said engineered test data.

19. The apparatus of claim 1, wherein said data engineering module is further configured to enable a user to view said engineered test data.

20. The apparatus of claim 1, wherein said data engineering module includes plural means for generating engineered test data, and wherein a particular means for generating engineered test data is generated based at least in part on information concerning said application to be tested.

21. The apparatus of claim 1, wherein said data engineering module includes a data enhancer, a data still, a negative-data manufacturer, and a

positive-data manufacturer, each of which is configured to provide data that forms a basis for said engineered test data.

22. The apparatus of claim 1, wherein said data engineering module includes logic extractor configured to identify logical functions within said application to be tested that are to be tested and provides those logical functions to a data still.

23. The apparatus of claim 1, wherein said data engineering module is further configured to identify logical functions within said application to be tested that are to be tested and provides those logical functions to be used as a basis for obtaining a subset of production data.

24. The apparatus of claim 1, wherein said computational environment manager includes means for automatically setting up and taking down a computational environment in which said application to be tested that is to be tested is to be tested.

25. The apparatus of claim 1, wherein said computational environment manager includes an environmental-transition machine, wherein said environmental-transition machine is configured to identify a source of said engineered test data, and wherein said environmental-transition machine is further configured to identify a target in

which to place data that results from processing of said engineered test data by said application to be tested.

26. The apparatus of claim 1, wherein said environmental-transition machine is further configured to copy engineered test data from a first repository to said source.

27. The apparatus of claim 26, wherein said environmental-transition machine is further configured to copy engineered test data from said target to a second repository.

28. The apparatus of claim 1, wherein said computational environment manager includes an environmental-backup machine, and a restoration machine, wherein said environmental-backup machine is configured for backing up a first environment prior to transforming said first environment into a second environment, wherein said restoration machine is configured for replacing said second environment with said first environment, and wherein said second environment is an environment in which testing of said application to be tested is to take place.

29. The apparatus of claim 1, wherein said computational environment manager includes an executioner, wherein said executioner is configured to cause execution of said application to be tested.

30. The apparatus of claim 29, wherein said

executioner is configured to automatically execute a script when causing execution of said application.

31. The apparatus of claim 1, wherein said computational environment manager includes an environmental-transition machine, an environmental-backup machine, a restoration machine, and an executioner, wherein said environmental-transition machine is configured to identify a source of said engineered test data, wherein said environmental-transition machine is further configured to identify a target in which to place data that results from processing of said engineered test data by said application to be tested, wherein said environmental-backup machine is configured for backing up a first environment prior to transforming said first environment into a second environment, wherein said restoration machine is configured for replacing said second environment with said first environment, wherein said second environment is an environment in which testing of said application to be tested is to take place, and wherein said executioner is configured to cause execution of said application to be tested.

32. A method for processing data in a computing system, said method including: testing applications, wherein testing applications includes receiving information representative of an application to be tested over one of an input

device and a port of a data processing system, and processing said received information, wherein processing said received information includes creating engineered test data based at least in part on said information, controlling a computational environment in which said application is to operate on said engineered test data, and comparing engineered test data operated upon by said application with an expected output, said method further including outputting a result indicative of said comparison.

33. Software stored in a non-transitory form on a computer-readable medium for managing testing of applications, said software including instructions for causing a computing system to execute processing steps that include: creating engineered test data based at least in part on an application to be tested; controlling a computational environment in which said application is to operate on said engineered test data; comparing engineered test data operated upon by said application with an expected output; and outputting an analysis of said comparison.

AMENDED CLAIMS
received by the International Bureau on 06 January 2016 (06.01.2016)

1. An apparatus for testing applications, said apparatus including: a data-processing machine including a memory and a processor operably coupled to said memory, said data-processing machine having been configured to implement a data-driven testing framework that includes a data engineering module, a computational environment manager, and a result analysis module,

wherein said data engineering module is configured to create engineered test data based at least in part on an application to be tested,

wherein said computational environment manager is configured to control a computational environment in which said application is to operate on said engineered test data, and

wherein said result analysis module is configured to compare engineered test data operated upon by said application with an expected output.

2. The apparatus of claim **1**,

wherein said data engineering module is configured to extract a subset of production data,

wherein said subset is selected to achieve a specified code coverage, and

wherein said engineered test data includes said subset of said production data.

3. The apparatus of claim **1**, wherein said data engineering module includes a data still for generating distilled data from production data.

4. The apparatus of claim claim **34**, wherein said additional data is selected to achieve specified code coverage.

5. The apparatus of claim **1**, wherein said data engineering module includes a data enhancer for receiving distilled data from said data still and enhancing said distilled data.

6. The apparatus of claim **1**, wherein said data engineering module is configured to

generate data based at least in part on said application to be tested, wherein said generated data is selected to achieve specified code coverage, and wherein said engineered test data includes said generated data.

7. The apparatus of claim 1, wherein said data engineering module further includes a positive-data manufacturer for generating positive data.

8. The apparatus of claim 1, wherein said data engineering module is configured to generate data based at least in part on said application to be tested, and wherein said data is absent from production data.

9. The apparatus of claim 1, wherein said data engineering module further includes a negative-data manufacturer for generating negative data.

10. The apparatus of claim 1, wherein said data engineering module includes means for generating engineered test data.

11. The apparatus of claim 1, wherein said data engineering module includes an integrity checker for determining referential integrity of said engineered test data.

12. The apparatus of claim 1, wherein said data engineering module is further configured to detect errors in referential integrity.

13. The apparatus of claim 1, wherein said data engineering module includes a re-referencer for correcting a loss of referential integrity in data prior to outputting said data as engineered test data.

14. The apparatus of claim 1, wherein said data engineering module is further configured to correct a loss of referential integrity in data.

15. The apparatus of claim 1, wherein said data engineering module includes an inspection unit for receiving said engineered test data and enabling a user to at least one of view said engineered test data and profile said engineered test data.

16. (Original) The apparatus of claim 1, wherein said data engineering module

includes a data-inspection unit for receiving said engineered test data and enabling a user to view said engineered test data.

17. The apparatus of claim 1, wherein said data engineering module includes a profiler for receiving said engineered test data and enabling a user to profile said engineered test data.

18. The apparatus of claim 1, wherein said data engineering module is further configured to enable a user to profile said engineered test data.

19. The apparatus of claim 1, wherein said data engineering module is further configured to enable a user to view said engineered test data.

20. The apparatus of claim 1, wherein said data engineering module includes plural means for generating engineered test data, and wherein a particular means for generating engineered test data is generated based at least in part on information concerning said application to be tested.

21. The apparatus of claim 1, wherein said data engineering module includes a data enhancer, a data still, a negative-data manufacturer, and a positive-data manufacturer, each of which is configured to provide data that forms a basis for said engineered test data.

22. The apparatus of claim 1, wherein said data engineering module includes logic extractor configured to identify logical functions within said application to be tested that are to be tested and provides those logical functions to a data still.

23. The apparatus of claim 1, wherein said data engineering module is further configured to identify logical functions within said application to be tested that are to be tested and provides those logical functions to be used as a basis for obtaining a subset of production data.

24. The apparatus of claim 1, wherein said computational environment manager includes means for automatically setting up and taking down a computational environment in which said application to be tested that is to be tested is to be

tested.

25. The apparatus of claim 1, wherein said computational environment manager includes an environmental-transition machine, wherein said environmental-transition machine is configured to identify a source of said engineered test data, and wherein said environmental-transition machine is further configured to identify a target in which to place data that results from processing of said engineered test data by said application to be tested.

26. The apparatus of claim 1, wherein said environmental-transition machine is further configured to copy engineered test data from a first repository to said source.

27. The apparatus of claim 26, wherein said environmental-transition machine is further configured to copy engineered test data from said target to a second repository.

28. The apparatus of claim 1, wherein said computational environment manager includes an environmental-backup machine, and a restoration machine, wherein said environmental-backup machine is configured for backing up a first environment prior to transforming said first environment into a second environment, wherein said restoration machine is configured for replacing said second environment with said first environment, and wherein said second environment is an environment in which testing of said application to be tested is to take place.

29. The apparatus of claim 1, wherein said computational environment manager includes an executioner, wherein said executioner is configured to cause execution of said application to be tested.

30. The apparatus of claim 29, wherein said executioner is configured to automatically execute a script when causing execution of said application.

31. The apparatus of claim 1, wherein said computational environment manager

includes an environmental-transition machine, an environmental-backup machine, a restoration machine, and an executioner, wherein said environmental-transition machine is configured to identify a source of said engineered test data, wherein said environmental-transition machine is further configured to identify a target in which to place data that results from processing of said engineered test data by said application to be tested, wherein said environmental-backup machine is configured for backing up a first environment prior to transforming said first environment into a second environment, wherein said restoration machine is configured for replacing said second environment with said first environment, wherein said second environment is an environment in which testing of said application to be tested is to take place, and wherein said executioner is configured to cause execution of said application to be tested.

32. A method for processing data in a computing system, said method including: testing applications, wherein testing applications includes receiving information representative of an application to be tested over one of an input device and a port of a data processing system, and processing said received information, wherein processing said received information includes creating engineered test data based at least in part on said information, controlling a computational environment in which said application is to operate on said engineered test data, and comparing engineered test data operated upon by said application with an expected output, said method further including outputting a result indicative of said comparison.

33. Software stored in a non-transitory form on a computer-readable medium for managing testing of applications, said software including instructions for causing a computing system to execute processing steps that include: creating engineered test data based at least in part on an application to be tested; controlling a computational environment in which said application is to operate on said engineered test data; comparing engineered test data operated upon by said application with an expected output; and outputting an analysis of said comparison.

34. The apparatus of claim 1, wherein said data engineering module is configured to extract a subset of existing data, wherein said data engineering module is further configured to augment said subset, thereby generating augmented data, and wherein said engineered test data includes said augmented data.

35. The apparatus of claim 34, wherein said augmented data includes one or more fields added to one or more records of said subset.

36. The apparatus of claim 35, wherein said data engineering module is further configured to generate data to fill the added one or more fields based on one or more supplied rules.

37. The apparatus of claim 1, wherein said data engineering module is configured to create engineered test data by distillation of existing data, wherein said engineered test data has a higher logic concentration than said existing data.

CLAIMS

1. An apparatus for testing applications, said apparatus including: a data-processing machine including a memory and a processor operably coupled to said memory, said data-processing machine having been configured to implement a data-driven testing framework that includes a data engineering module, a computational environment manager, and a result analysis module,

wherein said data engineering module is configured to create engineered test data based at least in part on an application to be tested, including

using data-lineage to track derivation of at least one derived variable through logic of the application to be tested,

identifying at least one input variable from which said derived variable is derived, and

including in said engineered test data a first value for said input variable that corresponds to a second value for said derived variable,

wherein said computational environment manager is configured to control a computational environment in which said application is to operate on said engineered test data, wherein said input variable and said derived variable are accessed within said computational environment, and

wherein said result analysis module is configured to compare an output resulting from said engineered test

data being operated upon by said application with an expected output.

2. The apparatus of claim **1**,

wherein said data engineering module is configured to extract a subset of production data,

wherein said subset is selected to achieve a specified code coverage, and

wherein said engineered test data includes said subset of said production data.

3. The apparatus of claim **1**, wherein said data engineering module includes a data still for generating distilled data from production data.

4. The apparatus of claim **34**, wherein said additional data is selected to achieve specified code coverage.

5. The apparatus of claim **3**, wherein said data engineering module includes a data enhancer for receiving distilled data from said data still and enhancing said distilled data.

6. The apparatus of claim **1**, wherein said data engineering module is configured to generate data based at least in part on said application to be tested, wherein said generated data is selected to achieve specified code coverage, and wherein said engineered test data includes said generated data.

7. The apparatus of claim **1**, wherein said data engineering module further includes a positive-data manufacturer for

generating positive data.

8. The apparatus of claim 1, wherein said data engineering module is configured to generate data based at least in part on said application to be tested, and wherein said data is absent from production data.

9. The apparatus of claim 1, wherein said data engineering module further includes a negative-data manufacturer for generating negative data.

10. The apparatus of claim 1, wherein said data engineering module includes means for generating engineered test data.

11. The apparatus of claim 1, wherein said data engineering module includes an integrity checker for determining referential integrity of said engineered test data.

12. The apparatus of claim 1, wherein said data engineering module is further configured to detect errors in referential integrity.

13. The apparatus of claim 1, wherein said data engineering module includes a re-referencer for correcting a loss of referential integrity in data prior to outputting said data as engineered test data.

14. The apparatus of claim 1, wherein said data engineering module is further configured to correct a loss of referential integrity in data.

15. The apparatus of claim 1, wherein said data engineering module includes an inspection unit for receiving said engineered test data and enabling a user to at least one of view said engineered test data and profile said engineered

test data.

16. The apparatus of claim 1, wherein said data engineering module includes a data-inspection unit for receiving said engineered test data and enabling a user to view said engineered test data.

17. The apparatus of claim 1, wherein said data engineering module includes a profiler for receiving said engineered test data and enabling a user to profile said engineered test data.

18. The apparatus of claim 1, wherein said data engineering module is further configured to enable a user to profile said engineered test data.

19. The apparatus of claim 1, wherein said data engineering module is further configured to enable a user to view said engineered test data.

20. The apparatus of claim 1, wherein said data engineering module includes plural means for generating engineered test data, and wherein a particular means for generating engineered test data is generated based at least in part on information concerning said application to be tested.

21. The apparatus of claim 1, wherein said data engineering module includes a data enhancer, a data still, a negative-data manufacturer, and a positive-data manufacturer, each of which is configured to provide data that forms a basis for said engineered test data.

22. The apparatus of claim 1, wherein said data engineering module includes logic extractor configured to identify

logical functions within said application to be tested that are to be tested and provides those logical functions to a data still.

23. The apparatus of claim **1**, wherein said data engineering module is further configured to identify logical functions within said application to be tested that are to be tested and provides those logical functions to be used as a basis for obtaining a subset of production data.

24. The apparatus of claim **1**, wherein said computational environment manager includes means for automatically setting up and taking down a computational environment in which said application to be tested that is to be tested is to be tested.

25. The apparatus of claim **1**, wherein said computational environment manager includes an environmental-transition machine, wherein said environmental-transition machine is configured to identify a source of said engineered test data, and wherein said environmental-transition machine is further configured to identify a target in which to place data that results from processing of said engineered test data by said application to be tested.

26. The apparatus of claim **25**, wherein said environmental-transition machine is further configured to copy engineered test data from a first repository to said source.

27. The apparatus of claim **26**, wherein said environmental-transition machine is further configured to copy engineered test data from said target to a second repository.

28. The apparatus of claim **1**, wherein said computational

environment manager includes an environmental-backup machine, and a restoration machine, wherein said environmental-backup machine is configured for backing up a first environment prior to transforming said first environment into a second environment, wherein said restoration machine is configured for replacing said second environment with said first environment, and wherein said second environment is an environment in which testing of said application to be tested is to take place.

29. The apparatus of claim **1**, wherein said computational environment manager includes an executioner, wherein said executioner is configured to cause execution of said application to be tested.

30. The apparatus of claim **29**, wherein said executioner is configured to automatically execute a script when causing execution of said application.

31. The apparatus of claim **1**, wherein said computational environment manager includes an environmental-transition machine, an environmental-backup machine, a restoration machine, and an executioner, wherein said environmental-transition machine is configured to identify a source of said engineered test data, wherein said environmental-transition machine is further configured to identify a target in which to place data that results from processing of said engineered test data by said application to be tested, wherein said environmental-backup machine is configured for backing up a first environment prior to transforming said first environment into a second environment, wherein said restoration machine is configured for replacing said second environment with said first

environment, wherein said second environment is an environment in which testing of said application to be tested is to take place, and wherein said executioner is configured to cause execution of said application to be tested.

32. A method for processing data in a computing system, said method including:

testing applications, wherein testing applications includes receiving information representative of an application to be tested over one of an input device and a port of a data processing system, and processing said received information, wherein processing said received information includes

creating engineered test data based at least in part on said information, including

using data-lineage to track derivation of at least one derived variable through logic of the application to be tested,

identifying at least one input variable from which said derived variable is derived, and

including in said engineered test data a first value for said input variable that corresponds to a second value for said derived variable,

controlling a computational environment in which said application is to operate on said engineered test data, wherein said input variable and said derived variable are accessed within said computational environment, and

comparing an output resulting from said engineered test data being operated upon by said application with an expected output, and

outputting a result indicative of said comparison.

33. Software stored in a non-transitory form on a computer-readable medium for managing testing of applications, said software including instructions for causing a computing system to execute processing steps that include:

creating engineered test data based at least in part on an application to be tested, including

using data-lineage to track derivation of at least one derived variable through logic of the application to be tested,

identifying at least one input variable from which said derived variable is derived, and

including in said engineered test data a first value for said input variable that corresponds to a second value for said derived variable;

controlling a computational environment in which said application is to operate on said engineered test data, wherein said input variable and said derived variable are accessed within said computational environment;

comparing an output resulting from said engineered test data being operated upon by said application with an expected output; and

outputting an analysis of said comparison.

34. The apparatus of claim **1**, wherein said data engineering module is configured to extract a subset of existing data, wherein said data engineering module is further configured to augment said subset, thereby generating augmented data, and wherein said engineered test data includes said augmented data.

35. The apparatus of claim **34**, wherein said augmented data includes one or more fields added to one or more records of said subset.

36. The apparatus of claim **35**, wherein said data engineering module is further configured to generate data to fill the added one or more fields based on one or more supplied rules.

37. The apparatus of claim **1**, wherein said data engineering module is configured to create engineered test data by distillation of existing data, wherein said engineered test data has a higher logic concentration than said existing data.