

21

tem to gather information about the self-describing software artifacts and the manifests wherein:
 each software artifact comprises an offline manifestation of an executable entity; each executable entity is one of: a process, an application, or a component of an operating system,
 the manifests comprise metadata declarative descriptions of the multiple self-describing software artifacts and associated executable entities:
 in an event that multiple manifestations of a particular executable entity exist, a separate manifest is associated with each manifestation of the particular executable entity;
 each manifest exists in either a static manifest form or a dynamic manifest form such that:
 a manifest in the static manifest form is stored in association with a software artifact: and
 a manifest in the dynamic manifest form is employed during runtime of each associated executable entity,
 such that the manifest in the dynamic manifest form comprises dynamic metadata which is constructed at runtime to connect a plurality of runtime system elements; and;
 performing verification on the multiple self-describing software artifacts; and reporting results of the verification.

20. One or more media as recited in claim 19, wherein the acts further comprise verifying that dependencies of software components installed on a computer are met, the software components being composed of one or more self-describing software artifacts.

21. One or more media as recited in claim 19, wherein the acts further comprise performing act comprises-verifying that processor-executable instructions represented by one or more self-describing artifacts remains unaltered.

22. One or more media as recited in claim 19, wherein the acts further comprise verifying that an application is correctly installed in a computer, the application being composed of one or more self-describing software artifacts.

23. One or more media as recited in claim 19, wherein the acts further comprise verifying that a known faulty or malicious program is not installed on a computer.

24. One or more media as recited in claim 19, wherein the acts further comprise verifying that an application and all of its constituent components and dependencies are accessible from a computer before the application is installed on the computer, the application being composed of one or more self-describing software artifacts.

25. One or more media as recited in claim 19, wherein the acts further comprise verifying that an application is installable on the computer before loading components of the application onto the computer.

26. One or more media as recited in claim 19, wherein the acts further comprise verifying that installation of a new application will not conflict with existing applications, the applications being composed of one or more self-describing software artifacts.

27. One or more media as recited in claim 19, wherein the acts further comprise verifying that an application can be removed without breaking dependencies from other applications, the applications being composed of one or more self-describing software artifacts.

28. One or more media as recited in claim 19, wherein the acts further comprise verifying that an application conforms to a predefined local policy, the application being composed of one or more self-describing software artifacts.

22

29. One or more media as recited in claim 19, wherein one or more self-describing artifacts compose an application, having process-executable instructions that may be executed on a computer when the application is loaded into a memory of a computer, the acts further comprising optimizing the process-executable instructions of the application to improve the performance of the instructions of the application.

30. One or more media as recited in claim 19, wherein one or more self-describing artifacts compose an application, having process-executable instructions that may be executed on a computer when the application is loaded into a memory of a computer, the acts further comprising analyzing the process-executable instructions of the application to detect potential errors in execution behavior or operation of the instructions of the application.

31. A data structure embodied on one or more computer storage media, the data structure comprising:

multiple self-describing software artifacts comprising an offline manifestation of an executable entity and processor-executable instructions representing operating system components or applications, such components and applications being installed on a computing system;

manifests associated with the multiple self-describing software artifacts, the manifests being comprising metadata declarative descriptions of the multiple self-describing software artifacts and metadata declarative descriptions of associated executable entities, wherein in an event that multiple manifestations of a particular executable entity exist, a separate manifest is associated with each manifestation of the particular executable entity, and each manifest exists in one of a plurality of forms, the plurality of forms comprising:

a static manifest form stored in association with software artifacts and a dynamic manifest form employed during runtime of each associated executable entity, such that the dynamic manifests comprise dynamic metadata which is constructed at runtime to collect a plurality of runtime system elements.

32. One or more media as recited in claim 31, wherein one of the manifests is a system manifest comprising declarative descriptions of the multiple self-describing software artifacts.

33. One or more computer storage media having processor-executable instructions that, when executed by a processor, perform acts comprising:

obtaining a copy of an offline "system image" of a software-based computer, the system image representing the content and configuration of software components installed on the computer, wherein the installed software components are represented on the system image as self-describing software artifacts wherein each software artifact comprises an offline manifestation of an executable entity, each executable entity is one of: a process, an application, or a component of an operating system.

and each self-describing software artifact has an associated persistently saved manifest comprising a metadata declarative description of the manifest's software artifact and a metadata declarative description of an associated executable entity, such that in an event that multiple manifestations of a particular executable entity exist, a separate manifest is associated with each manifestation of the particular executable entity; and each manifest exists in either a static manifest form or a dynamic manifest form;

in an event that a first manifest exists in the static manifest form, store the first manifest in association with software artifacts; and,