

## SOFTWARE DELIVERY MANAGER

### FIELD OF THE INVENTION

[0001] The present invention relates to the deployment of software and software components.

### BACKGROUND OF THE INVENTION

[0002] A goal of information technology (IT) is to support business processes. Business processes may include a set of web services, which are provided by applications running on different components from multiple vendors.

[0003] Applications within an IT landscape can include test systems, quality assurance systems, and production systems. Each such system provides one or more computer program applications (“application programs” or simply “applications”) running on a computer platform based on computer hardware and peripherals, on which are generally built several layers of software to support the execution and use of application programs. One example of such a platform is a web application server that provides a J2EE (Java 2 Platform, Enterprise Edition) platform for Java programming language applications.

[0004] Application requirements within the IT landscape change over time as companies institute business or organizational changes, as they implement addition functions. For this and other reasons, vendors of application programs generally generate and provide to their customers patches, updates, and upgrades to correct errors, to satisfy new technical requirements, and to satisfy new business requirements. Customers may also generate patches and updates, as well as customer-specific extensions to vendor-supplied applications. Sometimes an update or upgrade involves changes in customer configuration settings for particular applications or the migration of particular software (i.e., computer program) components to newer software releases. In addition, as IT landscapes change, intersystem and inter-application dependencies must be managed. A change in one component, for example, may require adaptations in another component.

[0005] Patches, updates, and upgrades must generally be deployed. In simple systems, deployment may involve no more than copying an updated component, for example, to a directory in a file system. Generally, however, deployment involves many steps of selecting, configuring, removing, and installing multiple components to achieve a desired application and system state. In enterprise and mission critical contexts, a customer will generally deploy and test new application and system elements on a test system or a quality assurance system, or both, before the deployment is performed on a production system, that is, on a system that is used as a live data processing production environment.

[0006] The technical effect of a deployment of software into a computer system is to change the functional or operational characteristics of the system, or both, or of one or more applications or other services running on the system or in conjunction with the system.

### SUMMARY OF THE INVENTION

[0007] The invention provides software deployment tools and methods and data structures performed by or used in conjunction with such tools. An implementation of a tool in

accordance with the invention will be described. It is called the Software Delivery Manager (SDM).

[0008] In general, in one aspect, the invention features a software deployment tool that uses an abstract, extensible notion of software server types. Instances of these software server types are used as targets systems for software delivery and may form alone or combined a production runtime environment. Examples are J2EE-servers, database instances or file systems.

[0009] In general, in another aspect, the invention features a software deployment tool that uses an abstract, extensible notion of software types. All software deliveries are typed and, through a type mapping, target systems within a local deployment environment are found.

[0010] In general, in another aspect, the invention features a software deployment tool that uses a uniform delivery archive format for all software types. In the SDM implementation, the format is ZIP- and JAR-compatible and self-contained. It includes the software to be delivered and additional versioning-, dependency- and deployment-information. What additional information is included can be dependent on the specific software type.

[0011] In general, in another aspect, the invention features a delivery archive carrying deployment information that can be used to transport knowledge from a development to a production system, or from a software producer to software consumer. This can ease the manual tasks involved during deployment of software or even allows for full automation of the deployment process. In the case of updates of already deployed archives, the deployment tool allows the reuse of the deployment parameters of a previous version.

[0012] In general, in another aspect, the invention features a delivery archive that, through the formulation and delivery of dependencies, can cause a software deployment tool to require specific versions of other software packages to be present within a production runtime environment. The deployment tool resolves the dependency information during deployment preparation and ensures that all dependencies are met or that the needed packages are deployed together.

[0013] In general, in another aspect, the invention features a software deployment tool that uses fingerprints on all delivered files within a delivery archive to check for modifications of previous versions. It supports the detection of collisions and the deployment of modified archives.

[0014] In general, in another aspect, the invention features a software deployment tool that provides the bundling of archives into larger units, which will be called catalogs. Because both catalogs and archives are provided, a software producer can create and deliver large logical units of software and at the same time retain the ability to deliver patches on a small, archive level.

[0015] In another aspect, the invention features a graphical user interface (GUI) and a programming interface (API), by which the functionality of the software deployment tool can be used at a customer site. Thus, configuration of target systems, configuration of deployments, processing software deployments and updates, and so on, may be done by a human deployer using the graphical user interface, or—in