Use Cases

James Falkner, Sun Microsystems, Inc.
Charlotte Allen, Sun Microsystems, Inc.
Gowri Sivaprasad, Sun Microsystems, Inc.

June 15, 2005

Table of Contents:

I Glossary:

- Initial Configuration - Setting initial parameters needed for operation
- Ongoing Configuration Changes - modifying initial parameters at any future point in time
- Component - A collection of software, data, some configuration. That which is to be installed. Examples: Microsoft Word, Spell Checker
- Install - Putting new software components on a target.
- Installable unit - The smallest unit that can be installed. SVr4 pkgs, RPMs, zip files, directories of file or even a single file are examples of installable units. Some units have meta-data associated with them and can perform intelligent operations with them. Installable units are based on different packaging formats. Common acronym for installable unit is "IU".
- Enterprise - The organization that employs the install operator, which,may set general policies. Don't use "installation" for this as it's,too easy to confuse with the act of installing.
- Install operator or installing operator - A person who interacts with, the installer or the provisioning application to perform the installation process on one or more hosts
- Installer - A program (piece of software) that is executed to guide the operator and perform the installation process on a given target
- Instance - An executable copy of a component as installed on a target. Multiple instances on a given system may share some files and configuration information, or may be completely separate copies.
- Migration - The conversion of customer data and customization from the format used by the previous version of a component to the newer version of a component. Updates should not require migrations. Upgrades may, but it is generally best if the component will continue to function (perhaps without new features) with the old data formats so that the upgraded system may be brought up before a possibly lengthy migration is performed.
- Provisioning Application - A network-wide piece of software that directs installation across hosts and maintains a database of available services.
- Service - A set of services available generally across the network. Examples: J2EE Application Containers; LDAP services; SMTP services.
- Target - Where a component gets installed. May be a system or a part of a system. The part of a system may be very formal (an instance of an OS or other container running on a given host) or rather informal (a directory root in a file system.)
- Uninstall - Removing existing software components from a system.
- Update - Replacement of limited parts of installed components with newer versions to correct errors or add very minimal new functionality.. Updates may need to be installed on short notice to correct pressing problems, thus updates should not require reconfiguration or migration.
- Upgrade - Replacement of entire installed components with newer versions to provide enhanced features Upgrades are installed infrequently and in a well-planned way, reconfiguration and migration may be required as part of an upgrade (of course, they are better avoided when possible.)

II. Introduction

In general, use cases and/or scenarios can be categorized based on a number of axes describing the characteristics in which the use case is found.

Use case axes:

- Local vs. Remote:

A local install is done while "logged onto" or "at the console" of the machine. A remote install is done while interacting with a provisioning application, which performs the install on individual machines through an agent or other bootstrap system.

An explicit assumption here is that local installs are always on just one install target/OS instance, while remote installs can be multiple (a single remote install is just a multiple remote install where the number of targets happens to be one, that's just a limiting case and not a particularly interesting one, so we don't have separate use case scenarios for "remote single" and "remote multiple.")

- Install vs. Update vs. Upgrade vs. Uninstall

See above Glossary

- Evaluation vs. Development vs. Deployment

Evaluation: Putting the software on just to play around with it, review it, determine if the feature set is acceptable, run the demos, etc.

Development: Putting the software on to develop more software that will run on top. In particular, a tools package would want to do development installs of underlying infrastructure like an application server.

Deployment: Putting the software on to actually use it in production.

III. Install/Uninstall Use Cases

- 1.Legacy packaging support:

Who: Administrator with native packaging skills
What: A product that uses SDD to 'wrap around' existing packaging technology
Why: A transition period will exist where products are produced using SDD, but underneath, still using native packaging systems
How:
1. Deploy solution based on SDD toolset
2. Manage deployed system using native packaging tools

- 2. Globalization support

Who: Administrator deploying a solution based on SDD
What: A product that uses SDD to describe a bundled product
Why: Not everyone speaks a common language
How:
1. User deploys SDD-based product in a number of languages
2. All output from the procedure must be localized, including user interface and log files.

- 3. Accessibility support

Who: Administrator deploying a solution based on SDD who has limited use of eyes, ears, hands, etc.
What: A product that uses SDD to describe a bundled product
Why: Support of disabilities and "Section 508 compliance"
How:
1. User deploys SDD-based product in a number of languages
2. All interfaces must comply with section 508 http://www.section508.gov/

- 4. Automation

Who: Administrator who wishes to deploy without interaction
What: A product that uses SDD to describe a bundled product
Why: Automated data centers
How:
1. User describes the goal of the deployment
2. User initiates deployment
3. Deployment proceeds unattended, and finishes, or stops with an error

- 5. Repeatable Automation

Who: Administrator who wishes to deploy N number of times without interaction
What: A product that uses SDD to describe a bundled product
Why: Automated data centers
How:
1. User captures output of a single deployment session
2. User initiates deployment on multiple different environments using base 'template'
3. Deployment proceeds unattended, and finishes, or stops with an error, on each target.

- 6. Network connectivity

Who: Administrator
What: A product that uses SDD to describe a bundled product
Why: Security of network connectivity
How:
1. User deploys onto a disconnected target, from local media. Any remote dependencies that cannot be detected and/or resolved produce errors/warnings.

- 7. Cancel

Who: Administrator
What: A product that uses SDD to describe a bundled product
Why: Ability to cancel a deployment and return to previous state.
How:
1. User begins deployment.
2. Deployment detects a state in which it cannot continue.
3. User stops deployment, and reverts to previous state.

- 8. Sequential Install/Uninstall

Who: Administrator
What: A product that uses SDD to describe a bundled product
Why: Ability to add onto a previous deployment
How:
1. User deploys part of a product.
2. Later, user (or automated service) discovers the need for additional components.
3. User (or automated service) installs the additional components from the same distribution.

- 9. Detection of previously installed components/Multiple instances

Who: Administrator
What: A product that has previously been installed
Why: Multiple copies of identical products lead to patching/update complexity.
How:
1. Admin installs product
2. Later, admin installs a later revision of said product.
3. Admin is given a choice to use some previous components if they are backward compatible, otherwise the admin is given the choice to install a new revision of such a component, or stop the installation.

- 10. Detection of remote dependencies

Who: Administrator
What: A solutions that has previously been installed on remote systems
Why: Some solutions are typically deployed across multiple (hetergenous) systems
How:
1. Admin installs middleware onto a server
2. Later, admin installs application from remote system, deploying its content to the remote system, after such services are discovered.

- 11. Detection of compatible but 3rd-party products

Who: Administrator
What: A solution that contains a dependency that can be satisfied by an external product.
Why: Solutions often adhere to an industry standard.
How:
1. Admin installs software onto a system.
2. software complies with some well-known standard, and can resolve its dependency using an external solution that provides such a standard interface.

- 12. Marketing bundles

Who: Marketing/Product management
What: A software offering containing multiple components that can be grouped based on marketing needs.
Why: Components of a product are often grouped together based on marketing needs, and those groupings are not obvious to the solution integration engineer.
How:
1. Solution integration (aka Release Engineering) is supplied a description of a marketing bundle.
2. Solution integration produces product that contains a user interface element allowing end users to 'select' such a bundle.

- 13. Uninstall

Who: Administrator
What: A software offering based on SDD.
Why: Clean uninstalls are often a sign of a robust product. Customers demand that products 'clean up after themselves' in a coherent way.
How:
1. Admin installs a product
2. Product is used in production, re-configured one or more times.

3. Admin uninstalls product. An option is given to completely erase the software, or leave behind data files and/or configuration.
4. Left-behind data/config can optionally be used in future re-installs to re-configure newly-installed software without intervention.

- 14. Dependency resolution

Who: Administrator
What: A software offering based on SDD.
Why: Dependency declarations must be granular enough to support advanced dependency detection and re-use.
How:
1. Admin installs a product
2. Admin installs a dependent product. Dependent product discovers compatible interfaces and those discoveries are used to calculate what needs to be installed, and what configuration needs to be put in place to use discovered dependency.
3. Uninstallation of dependency produces a warning.

- 15. Environmental checks

Who: Administrator
What: A software offering based on SDD.
Why: Environments change often and can sometimes prevent a given process from continuing.
How:
1. Admin installs a product
2. Installation detects the presence of some limiting factor (lack of disk space, OS patches are wrong, processor is too slow, etc).
3. Admin is given a choice to override, or fail the installation.

- 16. Status updates

Who: Administrator
What: A software offering based on SDD.
Why: Constant user feedback is important for admins to estimate required time and whether or not something is hung
How:
1. Admin installs a product
2. If, during any installation process (install, update, upgrade, uninstall), the local system is required to perform a task (ex. environmental checks, moving bits to the file system) it is made clear to the end user how long the expected task will take, the progress of the given task, and a description of the task and why it is needed.

- 17. Serviceability

Who: Administrator
What: A software offering based on SDD.
Why: Many solutions do not have sufficient capabilities to efficiently detect and diagnose errors in components.
How:
1. Admin installs product, in attended or unattended mode
2. A (not necessarily fatal) failure occurs
3. User examines log files to determine failure, and based on that content, is able to diagnose problem without calling support services.

- 18. Relocatability

Who: Administrator

What: A software offering based on SDD.
Why: Customer configurations are too varied to reliably know where to deploy components.
How:
1. Admin installs product into one of many non-local locations:
    a) an arbitrary local filesystem
    b) an arbitrary remote filesystem
    c) an arbitrary remote container
2. All relocated products are successfully linked to their relocated dependencies

- 19. Secure Install by default

Who: Administrator
What: A software offering based on SDD.
Why: Installing software onto a networked system requires that such default installation not be vulnerable to outside attacks
How:
1. Admin installs product
2. Product does not have any unsecured outward-facing services enabled by default
3. Admin configures outward-facing services and makes a deliberate attempt to start such services

- 20. User-based install

Who: User who does not have administrator privileges
What: A software offering based on SDD.
Why: For non-system software, any user should be able to install, configure, and use said software without requiring administrator privileges.
How:
1. User installs product (into a location into which they have "write access"
2. User configures and uses software. Software makes no use of (or disables certain features that require) administrator-only interfaces.

- 21. Multiple Instances

Who: Administrator and/or User
What: A software offering based on SDD.
Why: Software is often used in several different modes, such as in an evaluation mode, or testing mode, or deployment. Multiple, independent copies allows different "domains" to be created that can be individually configured, managed, and uninstalled without affecting other copies.
How:
1. User installs product that already exists on the system.
2. Rather than configuring multiple runtimes from a single install, user installs a complete copy in a different location. No dependencies exist between the multiple installs.

- 22. No down-time

Who: Administrator and/or User
What: A software offering based on SDD.
Why: It is often expensive to bring down a software service to perform maintenance and/or upgrade.
How:
1. User installs product
2. product is enabled in production systems
3. User upgrades production system using newer version (which has been validated offline).
4. New version of software (or its configuration) lies dormant until such a time as administrator makes a deliberate step to switch to the new software.
5. If new software does not behave as expected, it is disabled, and the service is reverted back to the old, known working service level.

- 23. Licensing

Who: Administrator
What: A software offering based on SDD
Why: Software is not free
How:
1. user installs product
2. Product requires activation license (either limited time span or perpetual use).
3. User enters required license information

- 24. Registration

Who: Administrator
What: A software offering based on SDD
Why: Marketing data
How:
1. user installs product
2. Installer presents registration option, noting what will be sent, and the benefits.
3. User proceeds with registration, or chooses to register later, or never.

- 25. Deployment help

Who: Administrator
What: A software offering based on SDD
Why: Customers might need additional information at the time of the install, and do not have documentation available.
How:
1. user installs product
2. During install, useful help messages (tooltips, popups, etc) are provided describing the products to install.

IV. Configuration Use Cases

- 26. Configuration Modes

Who: Administrator
What: A software offering based on SDD
Why: Ready-made configurations based on experience go a long way toward making the evaluation or initial impression experience better.
How:
1. user installs product
2. During install, several configuration templates are provided (such as “Evaluation”, “Typical”, “Express”, etc).
3. User selects one of the pre-defined configurations, which are known to work. If no such default configuration is possible for some components, then an informative message is provided and the user is required to fill in the custom configuration elements that couldn't be applied automatically.

- 27. Common configuration parameters

Who: Administrator
What: A software offering based on SDD
Why: It is annoying to have to enter the same value multiple times for identical configuration options.
How:
1. user installs product
2. If configuration at install time is selected, user is presented with the minimum number of configurable items. Any duplicate items are reduced to a single query. Any configuration items

which have constraints are checked at the time of entry.

- 28. Optional Configuration

Who: ISV/IHV
What: A product that uses SDD to describe a bundled product
Why: Ability to install without configuration (e.g. factory pre-install)
How:
1. ISV installs product onto system without configuration
2. ISV ships system to customer
3. Customer continues process by configuring pre-installed component(s)

- 29. Deployment validation

Who: Administrator
What: A software offering based on SDD.
Why: Customers are often not aware of invalid configurations they supply
How:
1. Admin installs a product
2. Each component selected for deployment has the opportunity to validate its future configuration, and present overrideable warnings.

- 29. Configuration Serialization

Who: Administrator
What: A software offering based on SDD.
Why: Serialized configuration is useful to describe a portable configuration for a component or product
How:
1. Admin installs a product
2. Admin serializes the applied configuration to multiple previously-installed products, and/or imports the configuration into a provisioning application.

V. Update/Upgrade Use Cases

- 30. Automatic detection of upgrades

Who: Administrator
What: A software offering based on SDD.
Why: Software is often obsolete the minute it leaves the factory.
How:
1. Admin installs a product
2. Installation or runtime update interface detects the presence of newer versions of components of installed or about-to-be-installed components or products from the local system.
3. Admin is given a choice to install discovered updates *that they are entitled to.*
4. Admin chooses to update, and updates the selected components, without breaking any single or inter-system dependencies.

- 31. Automatic detection of remote upgrades

Who: Administrator
What: A software offering based on SDD that has been installed on remote, managed systems
Why: Software is often obsolete the minute it leaves the factory.
How:
1. Admin installs a product onto remote system
2. Later, user is able to view and select updates for components or products installed on that remote system.

3. Admin is given a choice to install discovered updates *that they are entitled to.*
4. Admin chooses to update, and updates the selected components, without breaking any single or inter-system dependencies.

- 32. Automatic detection of update applicability

Who: Administrator
What: A software offering based on SDD that has been installed on remote, managed systems
Why: Some updates are not applicable to any and all instances of its base software
How:
1. Admin installs a product onto remote system
2. Later, user is able to view and select updates for components or products installed on that remote system.
3. Admin is given a choice to install discovered updates *that they are entitled to.*
4. Admin is not given a choice to update software for which updates are not applicable to.

VI. Developer/Assembler Use Cases

- 33. Embedding of 3rd-party components

Who: Assembler
What: A software offering packaged with SDD that fulfills a requirement in a larger, aggregated offering
Why: Off-the-shelf components are often cheaper to bundle rather than develop
How:
1. SDD-Based 3rd-party solution is aggregated into a larger distribution
2. When installed, 3rd-party solution appears alongside the 'native' components and can be linked to using metadata provided by 3rd-party bundle

- 34. Validation of bundled components

Who: User
What: A software offering packaged with SDD
Why: Metadata bundled with the product must enable validation (checksums, signatures, etc) to ensure deployed content is consistent with what the author intended.
How:
1. SDD-Based solution is deployed onto local and/or remote systems
2. Components are compared against their factory-prearranged validation parameters (checksums, signatures, etc).