

# FreshPatents.com: Data processing: software development, installation, and management - USPTO Class 717 Patent Applications Update

USPTO Class 717 - Data processing: software development, installation, and management

## **20090271760 - Method for application development**

A process and method for application development that conforms with Rapid Application Development (RAD) requirements and Software Engineering Institute (SEI) Software Capability Maturity Model Integrated (CMMI) Level 3 requirements. An application development process that conforms with Rapid Application Development (RAD) requirements and Software Engineering Institute (SEI) Software Capability Maturity Model Integrated (CMMI) Level 3 requirements includes one or more formally-named development cycles, one or more formally-named design cycles, in which each development cycle includes one or more formally-named design cycles, and one or more formally-named construction cycles, in which each design cycle includes one or more formally-named construction cycles....

---

## **20090271760 - Method for application development**

A process and method for application development that conforms with Rapid Application Development (RAD) requirements and Software Engineering Institute (SEI) Software Capability Maturity Model Integrated (CMMI) Level 3 requirements. An application development process that conforms with Rapid Application Development (RAD) requirements and Software Engineering Institute (SEI) Software Capability Maturity Model Integrated (CMMI) Level 3 requirements includes one or more formally-named development cycles, one or more formally-named design cycles, in which each development cycle includes one or more formally-named design cycles, and one or more formally-named construction cycles, in which each design cycle includes one or more formally-named construction cycles....

---

## **20090271761 - Generative programming system and method employing focused grammars**

A system and method may employ, focused grammars to facilitate automated generation of computer programs. Such implementation of focused grammars enables a new form of symbolic regression referred to as generative programming or automated programming. The search through the space of possible programs may be guided by a streak search method in accordance with which identified candidate programs that improve on the current streak may be used to create focused grammars for directing an additional localized search. In some embodiments, candidate programs are generated by randomly traversing focused grammars from the starting rule, and subsequently through the grammars, until a complete candidate program parse tree has been constructed. Candidate programs may then be executed, for example, by an evaluator, which may employ an interpreter adapted for use in conjunction with a Stack Manipulation Language or other interpreted language....

---

## **20090271761 - Generative programming system and method employing focused grammars**

A system and method may employ, focused grammars to facilitate automated generation of computer programs. Such implementation of focused grammars enables a new form of symbolic regression referred to as generative programming or automated programming. The search through the space of possible programs may be guided by a streak search method in accordance with which identified candidate programs that improve on the current streak may be used to create focused grammars for directing an additional localized search. In some embodiments, candidate programs are generated by randomly traversing focused grammars from the starting rule, and subsequently through the grammars, until a complete candidate program parse tree has been constructed. Candidate programs may then be executed, for example, by an evaluator, which may employ an interpreter adapted for use in conjunction with a Stack Manipulation Language or other interpreted language....

---

## **20090271762 - Business software application system and method**

A business software application system and method are provided....

---

## **20090271762 - Business software application system and method**

A business software application system and method are provided....

---

## **20090271763 - Method for protecting user-managed memory using an exception**

Techniques for providing safe user-managed memory are provided The techniques include performing memory allocation in constant time, performing memory de-allocation in constant time, performing memory access overhead for safety checking in constant time, and using the memory allocation, memory de-allocation and memory access overhead to protect at least one memory access, user-specified allocation and de-allocation with at least one exception for said user-managed memory....

---

## **20090271763 - Method for protecting user-managed memory using an exception**

Techniques for providing safe user-managed memory are provided The techniques include performing memory allocation in constant time, performing memory de-allocation in constant time, performing memory access overhead for safety checking in constant time, and using the memory allocation, memory de-allocation and memory access overhead to protect at least one memory access, user-specified allocation and de-allocation with at least one exception for said user-managed memory....

---

## **20090271764 - System and method for managing computer environment setup requests**

A system for managing requests for setting up computer test environments includes a setup request database storing the test environment setup requests and a request management module which provides a standardized user interface screen to submit the requests. If the request submitted by a user does not contain all of the required information necessary to set up the test environment, the user is prompted to provide the missing information, thereby minimizing a subsequent exchange of communications due to missing information....

---

## **20090271764 - System and method for managing computer environment setup requests**

A system for managing requests for setting up computer test environments includes a setup request database storing the test environment setup requests and a request management module which provides a standardized user interface screen to submit the requests. If the request submitted by a user does not contain all of the required information necessary to set up the test environment, the user is prompted to provide the missing information, thereby minimizing a subsequent exchange of communications due to missing information....

---

## **20090271765 - Consumer and producer specific semantics of shared object protocols**

Determining how to perform operations specified in executable code. A method may include accessing a language context. The language context is related to a consumer location in executable code. The language context specifies an operation to be performed on one or more objects. The method includes sending a message requesting information about how to perform the operation on the one or more objects. A meta-object is received. The meta-object includes or produces executable code, that when executed performs the operation on the one or more objects....

---

## **20090271765 - Consumer and producer specific semantics of shared object protocols**

Determining how to perform operations specified in executable code. A method may include accessing a language context. The language context is related to a consumer location in executable code. The language context specifies an operation to be performed on one or more objects. The method includes sending a message requesting information about how to perform the operation on the one or more objects. A meta-object is received. The meta-object includes or produces executable code, that when executed performs the operation on the one or more objects....

---

## **20090271767 - Method and an apparatus for evaluating a tool**

A method is disclosed for evaluating a tool used in a system including steps of providing top-level-challenges to be met by the tool in at least one life cycle phase of the system to enhance a productivity of the given system. In at least one embodiment, each top-level-challenge can be provided, each having a number of concepts or best practices with different numeric classification values. At least one tool profile of the tool is calculated for selected top-level-challenges by way of a function as a statistical function on the basis of numeric classification values assigned to sub-challenges of the top-level-challenges. The method and apparatus according to at least one embodiment of the present invention can be used for evaluating a software tool such as a service information system employed in an industrial system such as a power plant for one or several life cycle phases of the system including its engineering, commissioning, operation, service and modernization phase. The method and...

---

### **20090271767 - Method and an apparatus for evaluating a tool**

A method is disclosed for evaluating a tool used in a system including steps of providing top-level-challenges to be met by the tool in at least one life cycle phase of the system to enhance a productivity of the given system. In at least one embodiment, each top-level-challenge can be provided, each having a number of concepts or best practices with different numeric classification values. At least one tool profile of the tool is calculated for selected top-level-challenges by way of a function as a statistical function on the basis of numeric classification values assigned to sub-challenges of the top-level-challenges. The method and apparatus according to at least one embodiment of the present invention can be used for evaluating a software tool such as a service information system employed in an industrial system such as a power plant for one or several life cycle phases of the system including its engineering, commissioning, operation, service and modernization phase. The method and...

---

### **20090271766 - Methods, systems and computer program products for improving program performance by anti-refactoring**

A method for developing a computer program product includes: evaluating one or more refactoring actions to determine a performance attribute; associating the performance attribute with a refactoring action used in computer code; and undoing the refactoring action of the computer code based on the performance attribute....

---

### **20090271766 - Methods, systems and computer program products for improving program performance by anti-refactoring**

A method for developing a computer program product includes: evaluating one or more refactoring actions to determine a performance attribute; associating the performance attribute with a refactoring action used in computer code; and undoing the refactoring action of the computer code based on the performance attribute....

---

## **20090271768 - Discriminating program code updates after merging for live review**

A method for dynamically discriminating a program code patch update after merging a patch file into unmodified source code for dynamic review. The method includes providing a graphical user interface (GUI) to enable a user to merge a source code patch file into an existing, pre-modified version of source code to generate a variant form of the existing, pre-modified source code and binary artifacts, via the GUI, presenting a display image highlighting differences between existing, pre-modified source code and the variant form after merging the source code patch file into the existing, pre-modified source code, via the GUI, presenting the user with an option of debugging the existing, pre-modified version of the source code, the GUI presenting linked views highlighting the existing, pre-modified version of the source code with the variant form of the source code, in parallel, debugging the variant form of the source code, the GUI presenting linked views highlighting the existing, pre-modified version with the variant form...

---

## **20090271768 - Discriminating program code updates after merging for live review**

A method for dynamically discriminating a program code patch update after merging a patch file into unmodified source code for dynamic review. The method includes providing a graphical user interface (GUI) to enable a user to merge a source code patch file into an existing, pre-modified version of source code to generate a variant form of the existing, pre-modified source code and binary artifacts, via the GUI, presenting a display image highlighting differences between existing, pre-modified source code and the variant form after merging the source code patch file into the existing, pre-modified source code, via the GUI, presenting the user with an option of debugging the existing, pre-modified version of the source code, the GUI presenting linked views highlighting the existing, pre-modified version of the source code with the variant form of the source code, in parallel, debugging the variant form of the source code, the GUI presenting linked views highlighting the existing, pre-modified version with the variant form...

---

## **20090271769 - Detecting irregular performing code within computer programs**

A computer-implemented method of finding portions of a computer program exhibiting irregular performance can include detecting an entry into a function of a computer program under test indicating a current execution of the function and, responsive to detecting an exit from the function, determining a count of a computing resource utilized by the function during the current execution of the function. The count of the computing resource can be compared with a count range determined according to at least one prior execution of the function. The function can be selectively represented within a call graph as a plurality of nodes according to the comparing....

---

## **20090271769 - Detecting irregular performing code within computer programs**

A computer-implemented method of finding portions of a computer program exhibiting irregular performance can include detecting an entry into a function of a computer program under test indicating a current execution of the function and, responsive to detecting an exit from the function, determining a count of a computing resource utilized by the function during the current execution of the function. The count of the computing resource can be compared with a count range determined according to at least one prior execution of the function. The function can be selectively represented within a call graph as a plurality of nodes according to the comparing....

---

#### **20090271770 - Method, system, and computer program product for generating unit testing scripts**

A method, system, and computer program product are provided for generating unit testing scripts. The method includes receiving programming code having a condition expressed therein, and transforming the code by wrapping the condition into a mock function. The mock function receives a Boolean argument as an input and provides a Boolean value as an output irrespective of a pre-transformed input and a pre-transformed output of the condition. The Boolean value is capable of selectively being the Boolean argument received as the input or a different value specified by a user. The method further includes receiving a user input specifying a behavior relating to how a return value is generated for the mock function. The Boolean value is provided in accordance with the specified behavior. At least one pre-transformation output of the condition prior to the transforming step is direct-function independent....

---

#### **20090271770 - Method, system, and computer program product for generating unit testing scripts**

A method, system, and computer program product are provided for generating unit testing scripts. The method includes receiving programming code having a condition expressed therein, and transforming the code by wrapping the condition into a mock function. The mock function receives a Boolean argument as an input and provides a Boolean value as an output irrespective of a pre-transformed input and a pre-transformed output of the condition. The Boolean value is capable of selectively being the Boolean argument received as the input or a different value specified by a user. The method further includes receiving a user input specifying a behavior relating to how a return value is generated for the mock function. The Boolean value is provided in accordance with the specified behavior. At least one pre-transformation output of the condition prior to the transforming step is direct-function independent....

---

#### **20090271771 - System and methods for distributed execution of computer executable programs utilizing asymmetric translation**

Dynamic program translation is utilized to convert an executable program in a first language executable representation to a second language executable representation that is either optimal or essentially required for execution within a defined execution environment on a client computer system. A dynamic translator, typically executed on a first computer system, is invoked in response to a request from a second computer system to provide an identified executable program. The dynamic translator performs an execution path analysis and optimized translation from a first representation, such as a Java class file, to generate executable code in the second representation, such as JavaScript. Optimizations include cross-language feature modifications, selective replacement of nominally un-translatable code portions, and insertion of call-back service requests to dynamically emulate services required in the execution of the program....

---

### **20090271771 - System and methods for distributed execution of computer executable programs utilizing asymmetric translation**

Dynamic program translation is utilized to convert an executable program in a first language executable representation to a second language executable representation that is either optimal or essentially required for execution within a defined execution environment on a client computer system. A dynamic translator, typically executed on a first computer system, is invoked in response to a request from a second computer system to provide an identified executable program. The dynamic translator performs an execution path analysis and optimized translation from a first representation, such as a Java class file, to generate executable code in the second representation, such as JavaScript. Optimizations include cross-language feature modifications, selective replacement of nominally un-translatable code portions, and insertion of call-back service requests to dynamically emulate services required in the execution of the program....

---

### **20090271772 - Using hardware interrupts to drive dynamic binary code recompilation**

A method, computer system, and computer program product for using one or more hardware interrupts to drive dynamic binary code recompilation. The execution of a plurality of instructions is monitored to detect a problematic instruction. In response to detecting the problematic instruction, a hardware interrupt is thrown to a dynamic interrupt handler. A determination is made whether a threshold for dynamic binary code recompilation is satisfied. If the threshold for dynamic code recompilation is satisfied, the dynamic interrupt handler optimizes at least one of the plurality of instructions....

---

### **20090271772 - Using hardware interrupts to drive dynamic binary code recompilation**

A method, computer system, and computer program product for using one or more hardware interrupts to drive dynamic binary code recompilation. The execution of a plurality of instructions is monitored to detect a problematic instruction. In response to detecting the problematic instruction, a hardware interrupt is thrown to a dynamic interrupt handler. A determination is made whether a threshold for dynamic binary code recompilation is satisfied. If the threshold for dynamic code recompilation is satisfied, the dynamic interrupt handler optimizes at least one of the plurality of instructions....

---

### **20090271773 - Method of and instruction set for executing operations on a device**

A method for executing operations on a device includes executing one or more bytecode language instructions associated with a mobile agent. The mobile agent includes a state of a virtual machine and the bytecode language includes instructions such WARP, MESH, ESEL, ALERT, BROWSE, CAPS, SLEEP, and GPS,...

---

### **20090271773 - Method of and instruction set for executing operations on a device**

A method for executing operations on a device includes executing one or more bytecode language instructions associated with a mobile agent. The mobile agent includes a state of a virtual machine and the bytecode language includes instructions such WARP, MESH, ESEL, ALERT, BROWSE, CAPS, SLEEP, and GPS,...

---

### **20090271774 - System and method for the distribution of a program among cooperating processing elements**

A Veil program analyzes the source code and/or data of an existing sequential target program and determines how best to distribute the target program and data among the processing elements of a multi-processing element computing system. The Veil program analyzes source code loops, data sizes and types to prepare a set of distribution attempts, whereby each distribution is run under a run-time evaluation wrapper and evaluated to determine the optimal distribution across the available processing elements....

---

### **20090271774 - System and method for the distribution of a program among cooperating processing elements**

A Veil program analyzes the source code and/or data of an existing sequential target program and determines how best to distribute the target program and data among the processing elements of a multi-processing element computing system. The Veil program analyzes source code loops, data sizes and types to prepare a set of distribution attempts, whereby each distribution is run under a run-time evaluation wrapper and evaluated to determine the optimal distribution across the available processing elements....

---

### **20090271775 - Optimizing just-in-time compiling for a java application executing on a compute node**

Methods, systems, and products are disclosed for optimizing just-in-time (?JIT?) compiling for a Java application executing on a compute node, the compute node having installed upon it a Java Virtual Machine (?JVM?) capable of supporting the Java application, that include: identifying, by an application manager, a particular portion of the Java application; assigning, by the application manager, a JIT level to the particular portion of the Java application; and jitting, by the JVM installed on the compute node, the particular portion of the Java application in dependence upon the JIT level assigned to that particular portion of the Java application....

---

#### **20090271775 - Optimizing just-in-time compiling for a java application executing on a compute node**

Methods, systems, and products are disclosed for optimizing just-in-time (?JIT?) compiling for a Java application executing on a compute node, the compute node having installed upon it a Java Virtual Machine (?JVM?) capable of supporting the Java application, that include: identifying, by an application manager, a particular portion of the Java application; assigning, by the application manager, a JIT level to the particular portion of the Java application; and jitting, by the JVM installed on the compute node, the particular portion of the Java application in dependence upon the JIT level assigned to that particular portion of the Java application....

---

#### **20090271776 - Dynamic management of operating system resources**

A method for dynamically managing operating system resources of a computing device. Each of the operating system resources have one or more data elements. The method includes: at an observer module of the operating system, monitoring stability and/or usage of the one or more data elements of each of the operating system resources, selecting an unused and/or unstable operating system resource data element; and removing a selected operating system resource data element....

---

#### **20090271776 - Dynamic management of operating system resources**

A method for dynamically managing operating system resources of a computing device. Each of the operating system resources have one or more data elements. The method includes: at an observer module of the operating system, monitoring stability and/or usage of the one or more data elements of each of the operating system resources, selecting an unused and/or unstable operating system resource data element; and removing a selected operating system resource data element....

---

#### **20090271778 - Apparatus and methods for transport optimization for widget content delivery**

Apparatus and methods for transport optimization for widget content delivery includes obtaining updated content corresponding to a mobile widget having a plurality of informational hierarchical depths, and obtaining user navigation pattern historical data corresponding to the mobile widget, wherein the user navigation pattern historical data corresponds to the plurality of informational hierarchical depths. The aspects also include obtaining a first informational hierarchical depth range for inclusion in a first content update message for delivery to a device, wherein the first informational hierarchical depth range is based on the user navigation pattern historical data. Further, the aspects include generating a content update message with a first portion of the updated content corresponding to the first informational hierarchical depth range. Optionally, the aspects further include reducing a size of the first portion of the updated content in the first content update message, thereby defining a first compressed portion of the updated content....

---

#### **20090271778 - Apparatus and methods for transport optimization for widget content delivery**

Apparatus and methods for transport optimization for widget content delivery includes obtaining updated content corresponding to a mobile widget having a plurality of informational hierarchical depths, and obtaining user navigation pattern historical data corresponding to the mobile widget, wherein the user navigation pattern historical data corresponds to the plurality of informational hierarchical depths. The aspects also include obtaining a first informational hierarchical depth range for inclusion in a first content update message for delivery to a device, wherein the first informational hierarchical depth range is based on the user navigation pattern historical data. Further, the aspects include generating a content update message with a first portion of the updated content corresponding to the first informational hierarchical depth range. Optionally, the aspects further include reducing a size of the first portion of the updated content in the first content update message, thereby defining a first compressed portion of the updated content....

---

#### **20090271777 - Method for distributing update modules for computer software over a network**

A method for distributing update modules for computer software has been developed. The method includes requesting and receiving a list of updated software modules stored on a network server. The user then determines if each updated module is presently installed on the user's system. If the updated module is not installed, the user requests a runtime artifact from the server for the module which is then installed on the user's system....

---

#### **20090271777 - Method for distributing update modules for computer software over a network**

A method for distributing update modules for computer software has been developed. The method includes requesting and receiving a list of updated software modules stored on a network server. The user then determines if each updated module is presently installed on the user's system. If the updated module is not installed, the user requests a runtime artifact from the server for the module which is then installed on the user's system....

---

## **20090271779 - Updating a file using differences and file format therefor**

A method for updating a local file and format for the local to be updated is described. An update file, which is a newer version of the local file, is accessible from a server over a network. A list of new hash values corresponding to data blocks of the update file is retrieved from the server. An algorithm locates divisions between the data blocks when a value of a rolling checksum reaches a particular value, the blocks being of variable size. New hash values of the update file are compared with local hash values of the local file. For each of the new hash values not having a matching local hash value, the corresponding data block is downloaded the server written to a new file. For each of the new hash values having a matching local hash value, the corresponding block is copied to the new file. The procedure is executed entirely on the client....

---

## **20090271779 - Updating a file using differences and file format therefor**

A method for updating a local file and format for the local to be updated is described. An update file, which is a newer version of the local file, is accessible from a server over a network. A list of new hash values corresponding to data blocks of the update file is retrieved from the server. An algorithm locates divisions between the data blocks when a value of a rolling checksum reaches a particular value, the blocks being of variable size. New hash values of the update file are compared with local hash values of the local file. For each of the new hash values not having a matching local hash value, the corresponding data block is downloaded the server written to a new file. For each of the new hash values having a matching local hash value, the corresponding block is copied to the new file. The procedure is executed entirely on the client....

---

## **20090271780 - Automatic complete firmware upgrade**

A method for automatic firmware upgrade on a target embedded system connected to a TFTP server and a web console is disclosed. The method includes generating a single image from Linux kernel image and file system image and providing single image name as input to web console for firmware upgrade. The web console write upgrade flag and single image name in the kernel configuration and issue a restart command to the embedded system. At boot up time, the start up module checks for firmware upgrade flag set and issue commands for loading the image on target board and boots up with the upgraded image. The firmware upgrade is performed over the network. Further, the embedded system is upgraded with very less user interaction reducing firmware upgrade time and by reducing dependence of experienced/skilled person....

---

## **20090271780 - Automatic complete firmware upgrade**

A method for automatic firmware upgrade on a target embedded system connected to a TFTP server and a web console is disclosed. The method includes generating a single image from Linux kernel image and file system image and providing single image name as input to web console for firmware upgrade. The web console write upgrade flag and single image name in the kernel configuration and issue a restart command to the embedded system. At boot up time, the start up module checks for firmware upgrade flag set and issue commands for loading the image on target board and boots up with the upgraded image. The firmware upgrade is performed over the network. Further, the embedded system is upgraded with very less user interaction reducing firmware upgrade time and by reducing dependence of experienced/skilled person....

---

**20090271781 - Method, system, terminal and device management server for installing software components**

A method, system, and apparatus for installing software components are provided. The method includes a terminal which receives a software component package delivered by a Device Management (DM) server directly or a download address of the component package and a command instructing to download delivered by the DM server, where the software component package contains more than one software component; if the terminal receives the download address and the command instructing to download, the terminal downloads the software component package from the download address; the terminal installs the obtained software component package locally and generates management nodes and subnodes of the management nodes corresponding to the software components under a Deployed subtree of the Software Component Management Object (SCOMO) of the terminal....

---

**20090271781 - Method, system, terminal and device management server for installing software components**

A method, system, and apparatus for installing software components are provided. The method includes a terminal which receives a software component package delivered by a Device Management (DM) server directly or a download address of the component package and a command instructing to download delivered by the DM server, where the software component package contains more than one software component; if the terminal receives the download address and the command instructing to download, the terminal downloads the software component package from the download address; the terminal installs the obtained software component package locally and generates management nodes and subnodes of the management nodes corresponding to the software components under a Deployed subtree of the Software Component Management Object (SCOMO) of the terminal....

---

**20090271782 - Mechanism for determining applicability of software packages for installation**

Mechanism for determining applicability of a software package for installation is described herein. In one embodiment, a process is provided to retrieve authentication information of a component from an installation descriptor file, where the descriptor file describes installation information of the software package. The software package may include one or more components and each component having zero or more sub-components. For at least one sub-component of at least one existing component that has already been installed, an image of the sub-component is authenticated using an authentication key extracted from the authentication information to determine whether the component can be installed based on the existing component. Other methods and apparatuses are also described....

---

**20090271782 - Mechanism for determining applicability of software packages for installation**

Mechanism for determining applicability of a software package for installation is described herein. In one embodiment, a process is provided to retrieve authentication information of a component from an installation descriptor file, where the descriptor file describes installation information of the software package. The software package may include one or more components and each component having zero or more sub-components. For at least one sub-component of at least one existing component that has already been installed, an image of the sub-component is authenticated using an authentication key extracted from the authentication information to determine whether the component can be installed based on the existing component. Other methods and apparatuses are also described....

---

### **20090271783 - Electronic device and user interface display method thereof**

An electronic device and a user interface display method thereof are provided. The electronic device organizes a plurality of computer programs executable on the electronic device into a virtual book. Each page of the virtual book includes a user interface of one of the computer programs. The electronic device displays one of the pages as a current page so that the user interface of the current page is accessible to a user of the electronic device. The electronic device senses movement and/or pressure caused by the user and interprets the movement and/or the pressure as operating actions of the user on the virtual book. The electronic device downloads new computer programs from a download server and adds corresponding new pages into the virtual book. At least one of the computer programs provides a service of a corporate operator of a wireless service network....

---

### **20090271783 - Electronic device and user interface display method thereof**

An electronic device and a user interface display method thereof are provided. The electronic device organizes a plurality of computer programs executable on the electronic device into a virtual book. Each page of the virtual book includes a user interface of one of the computer programs. The electronic device displays one of the pages as a current page so that the user interface of the current page is accessible to a user of the electronic device. The electronic device senses movement and/or pressure caused by the user and interprets the movement and/or the pressure as operating actions of the user on the virtual book. The electronic device downloads new computer programs from a download server and adds corresponding new pages into the virtual book. At least one of the computer programs provides a service of a corporate operator of a wireless service network....

---