US 20090271214A1

(19) United States

(12) Patent Application Publication

(10) Pub. No.: US 2009/0271214 A1

(43) Pub. Date: Oct. 29, 2009

Kandasamy et al.

(54) **RULES ENGINE FRAMEWORK**

(75) Inventors: **Uma Kandasamy**, Atlanta, GA (US); **Neil Galloway**, Acworth, GA (US); **Krishnam Raju B. Datla**, Atlanta, GA (US)

Correspondence Address:
CONLEY ROSE, P.C.
5601 GRANITE PARKWAY, SUITE 750
PLANO, TX 75024 (US)

(73) Assignee: **Affiliated Computer Services, Inc.**, Dallas, TX (US)

(21) Appl. No.: **12/257,782**

(22) Filed: **Oct. 24, 2008**

Related U.S. Application Data

(60) Provisional application No. 61/048,810, filed on Apr. 29, 2008.

Publication Classification

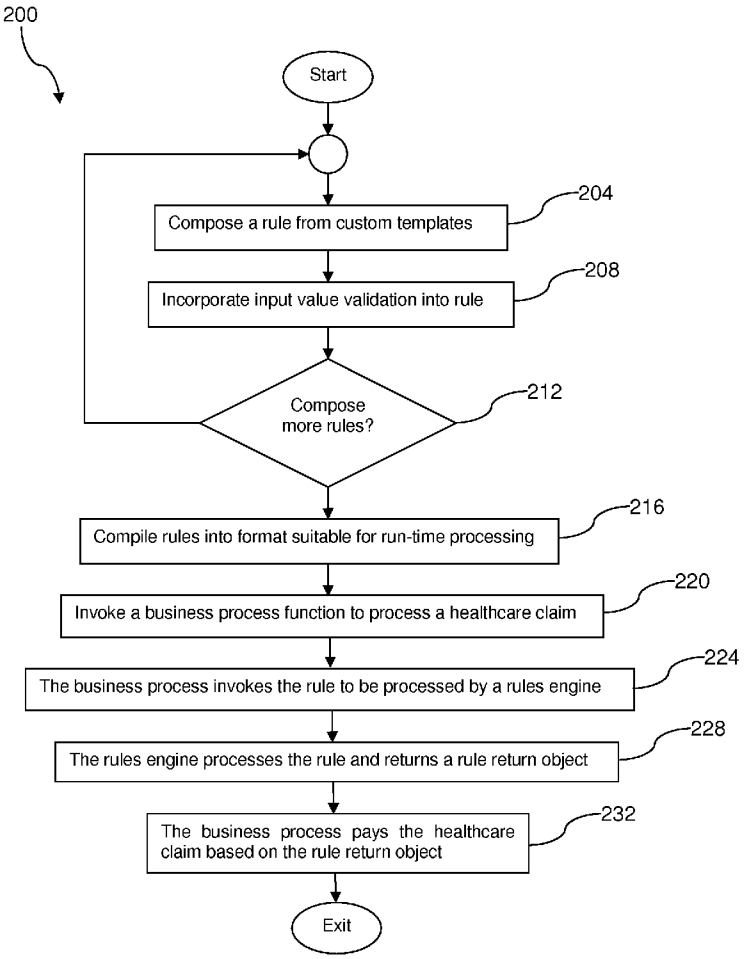
(51) **Int. Cl.**

G06Q 50/00	(2006.01)
G06Q 40/00	(2006.01)
G06Q 10/00	(2006.01)
G06F 17/21	(2006.01)

(52) **U.S. Cl.** **705/2**; 715/235

(57) **ABSTRACT**

An apparatus is provided. The apparatus comprises a computer system and an integrated development environment application. When executed on the computer system, the integrated development environment application promotes defining a plurality of rules based on a plurality of generic templates and automatically incorporating input value validation based on a type of a left hand variable into the rules. The rules, when executed by a healthcare management information system rules engine, promote enrollment of health care providers and healthcare system members and processing of healthcare claims.



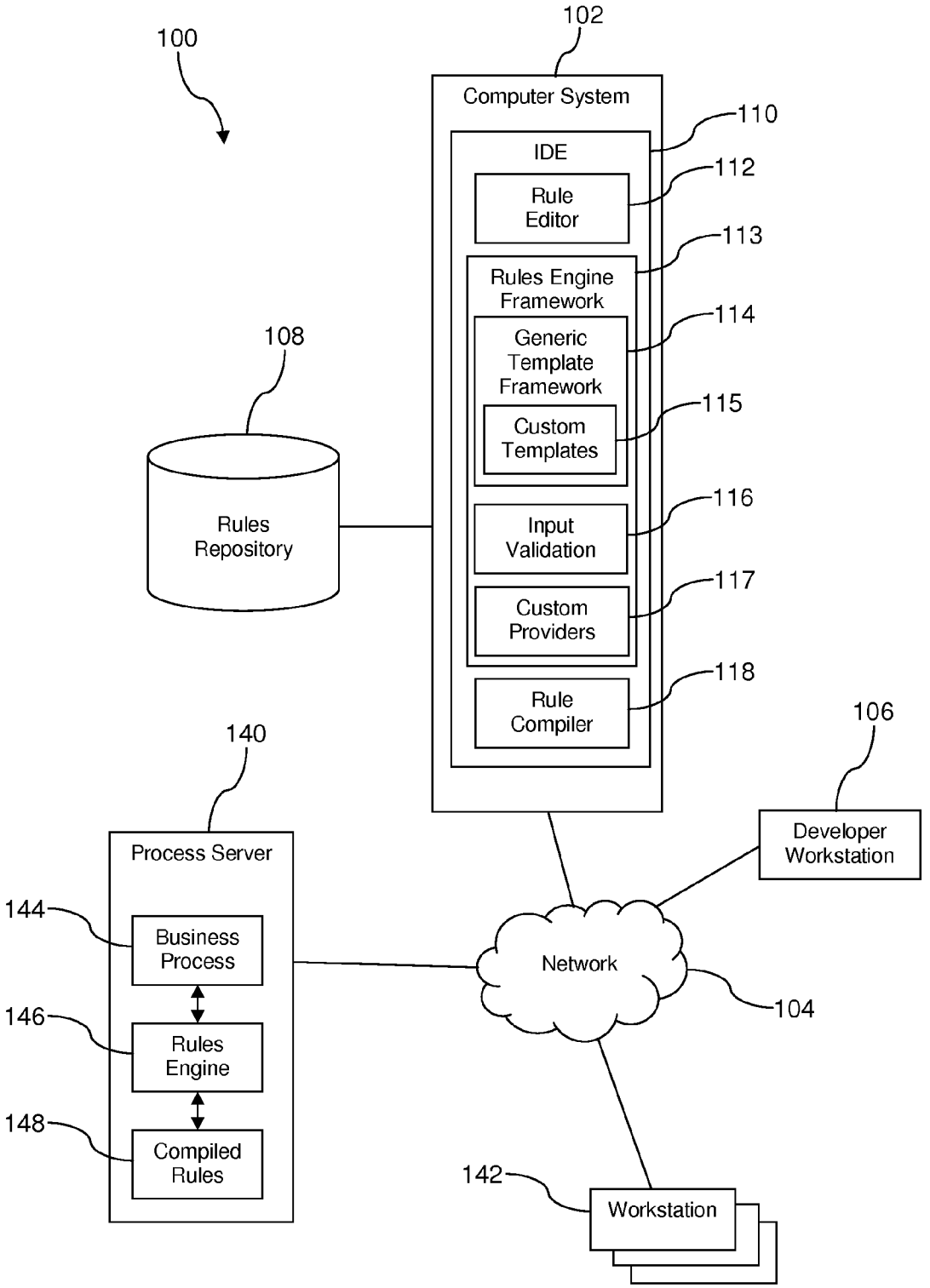

100
102
Computer System
110
IDE
112
Rule Editor
113
Rules Engine Framework
114
Generic Template Framework
115
Custom Templates
116
Input Validation
117
Custom Providers
118
Rule Compiler
108
Rules Repository
140
Process Server
144
Business Process
146
Rules Engine
148
Compiled Rules
106
Developer Workstation
Network
104
142
Workstation

FIG. 1

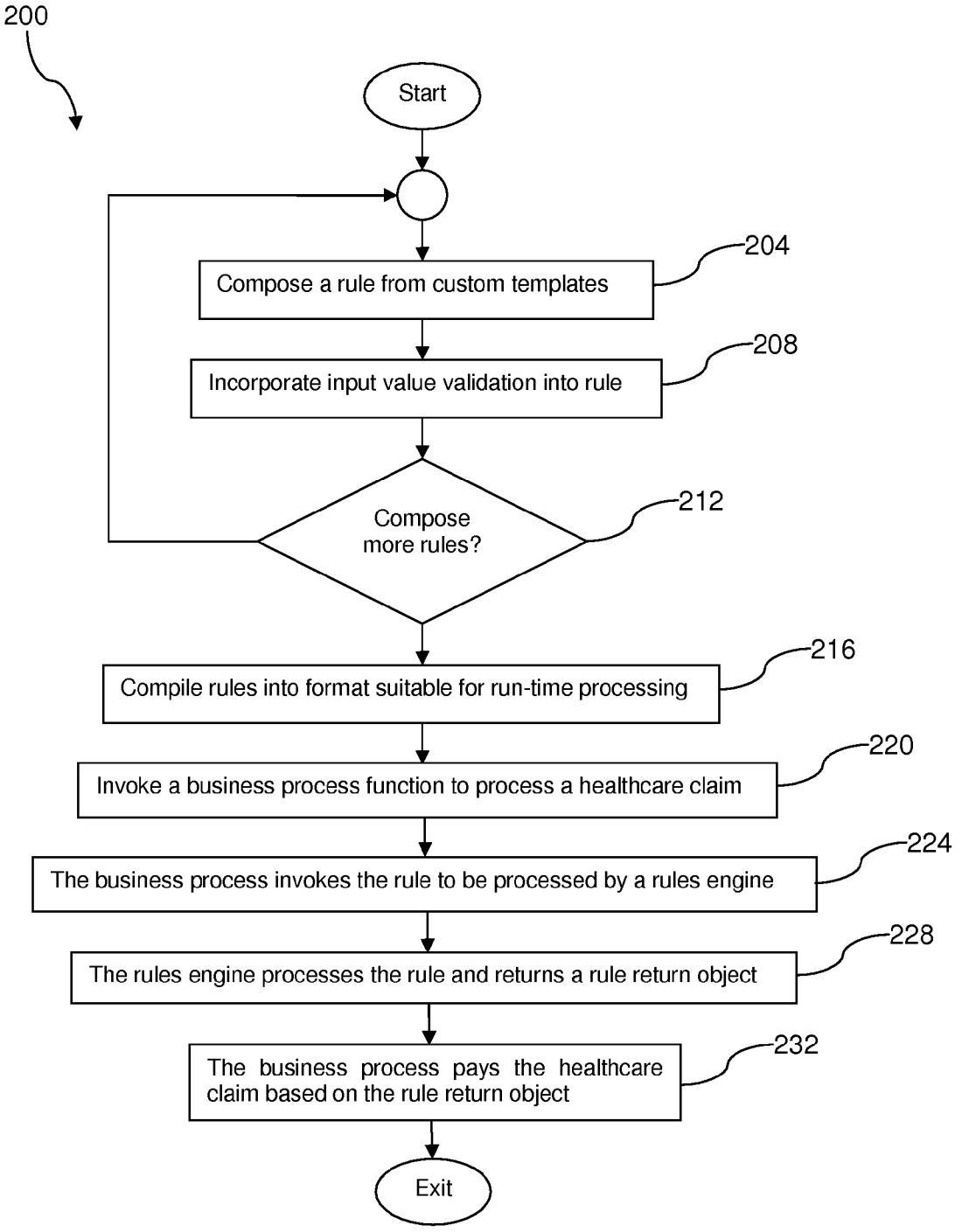

200
Start
204
Compose a rule from custom templates
208
Incorporate input value validation into rule
212
Compose more rules?
216
Compile rules into format suitable for run-time processing
220
Invoke a business process function to process a healthcare claim
224
The business process invokes the rule to be processed by a rules engine
228
The rules engine processes the rule and returns a rule return object
232
The business process pays the healthcare claim based on the rule return object
Exit

FIG. 2

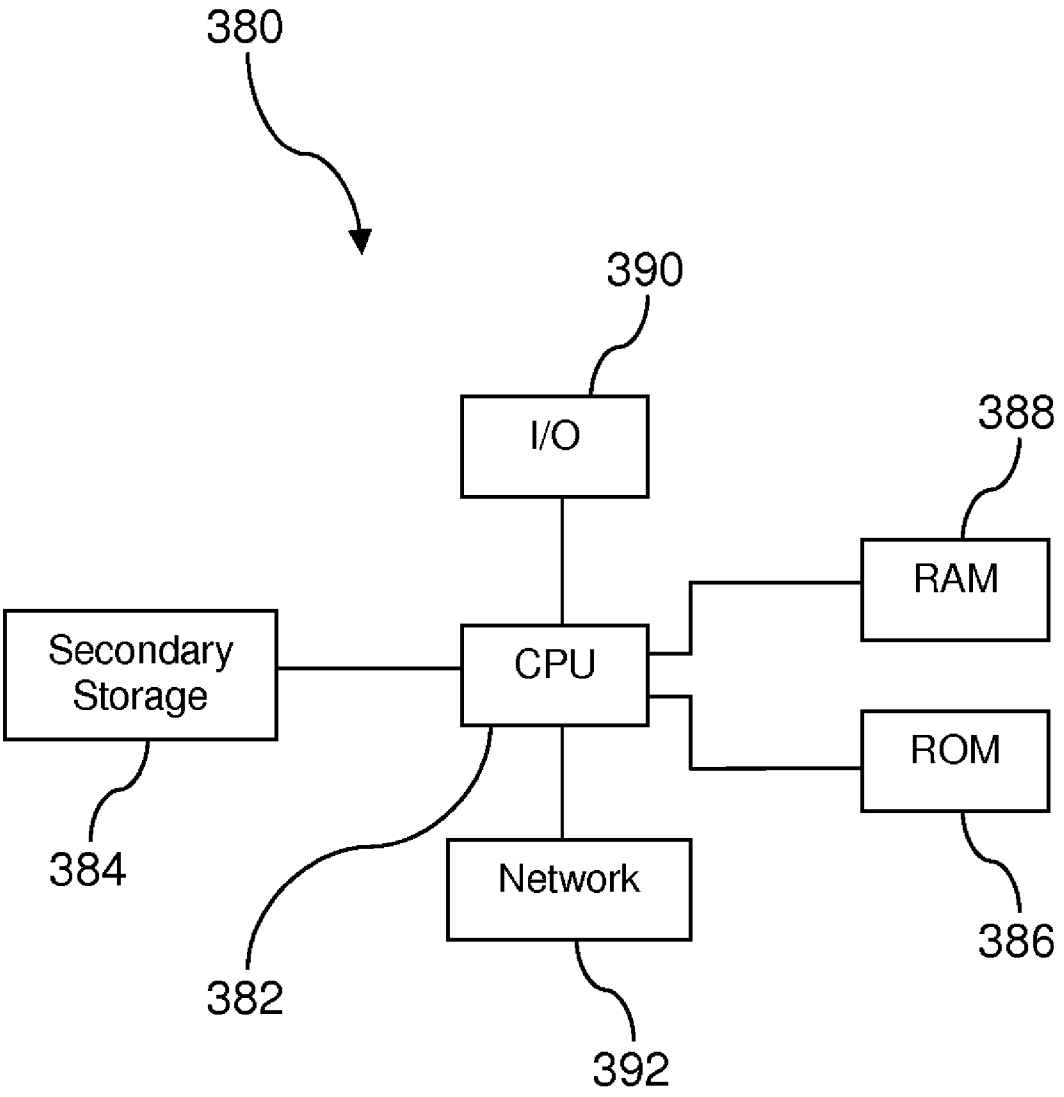

380
390
I/O
388
RAM
Secondary Storage
CPU
ROM
384
Network
386
382
392

FIG. 3

RULES ENGINE FRAMEWORK

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority to U.S. Provisional Application Ser. No. 61/048,810 filed Apr. 29, 2008, and entitled "MMIS Health Enterprise Solution," by Jack Devos, et al., which is incorporated herein by reference for all purposes.

STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT

[0002] Not applicable.

REFERENCE TO A MICROFICHE APPENDIX

[0003] Not applicable.

BACKGROUND

[0004] The United States Medicaid program was enacted in 1965 to provide a medical assistance program for individuals and families with low incomes. The Medicaid program comprises three main entities—the patients, the healthcare providers, and the agency administering the plan (i.e., the payer). The Medicaid program is financed through joint federal and state funding. The Medicaid program is administered by each state according to an approved state plan. The specifics of the Medicaid program differ from state to state. Differences may include covered healthcare procedures, allowable procedure costs, and patient eligibility criteria. The state administrators of the Medicaid program are required to have a Medicaid management information system (MMIS) that provides for mechanized and/or computerized Medicaid claims processing. Recently, the Medicaid information technology architecture (MITA) has been promulgated by the U.S. government to provide a blueprint and a set of standards that individual states are to follow in administering the Medicaid program and for developing the next generation MMIS.

SUMMARY

[0005] In an embodiment, an apparatus is disclosed. The apparatus comprises a computer system and an integrated development environment application. When executed on the computer system, the integrated development environment application promotes defining a plurality of rules based on a plurality of generic templates and automatically incorporating input value validation based on a type of a left hand variable into the rules. The rules, when executed by a healthcare management information system rules engine, promote enrollment of health care providers and healthcare system members and processing of healthcare claims.

[0006] In another embodiment, a method of processing a healthcare claim is disclosed. The method comprises composing a rule from at least one of a generic template, system parameters custom template, a system list custom template, and a valid values custom template. The method also comprises invoking a business process function to process a healthcare claim and the business process invoking the rule to be processed by a rules engine. The method also comprises the rules engine returning the result of processing the rule in a rule return object and the business process approving payment the healthcare claim based on the rule return object.

[0007] In another embodiment, a system is disclosed. The system comprises a computer system and a rules engine framework. When executed on the computer system, the rules engine framework promotes composing a plurality of rules. Each rule is composed at least in part from a left hand variable associated with a right hand input by an operator. The rules engine builder also automatically incorporates into the rules input value validation of the right hand input operand based on the type of the left hand variable. At least some of the rules are composed based on at least one of a generic template, system list custom template, a system parameters custom template, and a valid values custom template. The rules engine builder also builds a healthcare management information system rules engine based at least in part on a plurality of custom providers. The healthcare management information system rules engine promotes enrollment of health care providers and patients and promotes healthcare claims processing.

[0008] These and other features will be more clearly understood from the following detailed description taken in conjunction with the accompanying drawings and claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] For a more complete understanding of the present disclosure, reference is now made to the following brief description, taken in connection with the accompanying drawings and detailed description, wherein like reference numerals represent like parts.

[0010] FIG. **1** is a schematic diagram of a rules engine framework according to an embodiment of the disclosure.

[0011] FIG. **2** is a flowchart of a method according to an embodiment of the disclosure.

[0012] FIG. **3** is a schematic diagram of an exemplary general-purpose computer system suitable for implementing the several embodiments of the disclosure.

DETAILED DESCRIPTION

[0013] It should be understood at the outset that although illustrative implementations of one or more embodiments are illustrated below, the disclosed systems and methods may be implemented using any number of techniques, whether currently known or in existence. The disclosure should in no way be limited to the illustrative implementations, drawings, and techniques illustrated below, but may be modified within the scope of the appended claims along with their full scope of equivalents.

[0014] A rules engine framework is disclosed. In an embodiment, the rules engine framework provides additional functionality and ease of use to a commercial-off-the-shelf (COTS) rules engine application. The rules engine application may comprise various support tools including an integrated development environment (IDE) tool for defining and pre-processing rules as well as a rules engine. The rules engine is a computer executable program that processes rules. Generally speaking, rules engines may be pluggable and/or reusable software components that execute rules that have been externalized from application code. The rules define business rules and/or business logic that may change frequently. The rules may be precompiled into a binary format, (.adb format), which may be stored in random access memory for rapid access and execution by a rules engine.

[0015] The framework extends the functionality provided by the COTS rules engine application in a number of ways.

For example, the framework extends the COTS rules engine by providing a set of generic templates that may be used to define a condition portion of a rule comprising a condition to be evaluated and an action to be performed if the condition evaluates to TRUE. The generic templates identify the association and/or operation to be performed between a left hand variable and a right hand input of a condition. In some contexts, the left hand variable may be referred to as a left hand operand and the right hand input may be referred to as a right hand operand. The left hand variable may correspond to or identify an input field of an interface, for example a claim processing interface. Generic templates are a piece or snippet of code that can be combined together with operands and/or other generic templates to define a rule. Custom templates are a special case of generic templates that are custom developed as part of the rules engine framework to provide functionality not provided by the COTS rules engine application. A system parameter custom template is provided that promotes convenient definition of value constraints, for example a maximum annual patient charge for a specific procedure. A valid values custom template promotes convenient identification of valid values for input fields, for example valid values for place of service code. A systems list custom template is provided that promotes convenient groupings of valid values. The use of the generic templates of the present disclosure may substantially ease the burden of subject matter experts (SMEs) defining new rules.

[0016] The rules engine framework further comprises a plurality of custom providers that extend the COTS rules engine application including a database custom provider, an array object custom provider, and a display name custom provider. The database custom provider promotes retrieval of two columns values from the database. The array object custom provider promotes retrieval of all of the elements of an array in one operation, which may be useful, for example, when checking status of each of a plurality of common elements such as the presence or absence of 32 teeth in a patient. The display name custom provider promotes the use and/or display of data value labels in a consistent format. For example, the display name custom provider causes the label "Last Name" to be displayed with a particular data set, as opposed to code-driven labels, such as "lname," "lastname," or "last_name."

[0017] In an embodiment, the rules engine framework may be provided as an extension of the COTS IDE tool that executes on a computer system. The rules engine framework may also be provided as an associated rules repository. The rules repository stores the defined rules and other artifacts and objects created by using the integrated development environment application.

[0018] In an embodiment, the rules engine executes on a process server computer system in cooperation and coordination with a business process and the compiled rules. The business process may be invoked to perform a business level operation, for example process a submitted claim. In processing the business level operation, the business process may invoke the rules engine to perform decision logic encapsulated in the form of rules definitions, for example to validate a portion of an input screen and/or a portion of a claim. In an embodiment, the business process and rules engine cooperatively provide at least a portion of a healthcare management information system. In an embodiment, the rules engine and the business process need not execute on the same computer system. A user of the healthcare management information system may interact with the business process and rules engine by entering information and selecting actions from a user interface presented at a workstation communicatively coupled to the process server computer system. The business process may promote enrolling healthcare providers into the healthcare system, for example enrolling physicians, therapists, hospitals, minor emergency medical centers, and other healthcare providers. The business process may also promote enrolling healthcare recipients and/or patients. The business process may promote receiving and processing claims from enrolled healthcare providers for services provided to enrolled healthcare recipients.

[0019] A function of the business process may be invoked by the user of the workstation, for example process a claim for healthcare services. The business process may process this claim by invoking a series of rules and/or a rule flow on the rules engine. The use of rules to provide automated processing may have the advantage of allowing for the rapid creation, modification, and deployment of rules with respect to processing based on specifically designed computer programs. The processing of the claim may involve the rules engine executing a sequence of rules that validate the enrollment of the healthcare provider, validate the enrollment of the healthcare recipient, determine a coverage status of the subject procedure, determine a maximum coverage amount, initiate a funds deposit to pay the claim to the healthcare provider, and/or other healthcare processing actions.

[0020] Turning now to FIG. **1**, a system **100** that includes a rules engine framework is discussed. The system **100** comprises a computer system **102**, a network **104**, a developer workstation **106**, and a rules repository **108**. The computer system **102** may contain and/or execute an IDE application **110** used to compose a plurality of rules and to compile the rules into a form suitable for run-time execution. The IDE application **110** may comprise a rule editor **112**, a rules engine framework **113**, and a rule compiler **118**. The rules engine framework **113** comprises a generic template framework **114**, an input value validation component **116**, and a plurality of custom providers **117**. In some contexts, the rules engine framework **113** may also be considered to further comprise the rules repository **108**. The generic template framework **114** comprises a plurality of generic templates (not shown) and a plurality of custom templates **115**. The rules engine framework **113** is a novel extension of the IDE application **110** as delivered as a COTS computer application.

[0021] The system **100** may further include a process server **140** and a plurality of workstations **142**. The process server **140** may comprise at least one business process **144**, a rules engine **146**, and a plurality of compiled rules **148**. The rules engine **146** may be a pluggable and/or reusable software component and executes rules that are defined using the IDE application **110**, in part using the rules engine framework **113**. The process server **140** provides at least a portion of the Enterprise function, for example at least a portion of a healthcare management information system, to users of the workstations **142**.

[0022] The computer system **102**, the developer workstation **106**, the process server **140**, and the workstations **142** may be implemented as general-purpose computer systems, which are discussed in detail hereinafter. In an embodiment, one or more of the computer system **102**, the developer workstation **106**, the process server **140**, and the workstations **142** may be combined and/or distributed in space. For example, in an embodiment, the computer system **102** may be located at a

central location and one or more developer workstations **106** may log into the computer system **102** and/or establish an editing session from one or more remote locations. Alternatively, the IDE application **110** may be executed on a plurality of computer systems **102**, which each execute a developer workstation **106**. Similarly, the process server **140** may be located at a central location and one or more workstations **142** may log into the process server **140**, for example from a plurality of remote locations corresponding to different healthcare management information system offices located in population centers distributed throughout a state. Alternatively, a plurality of process servers **140** may be located at different offices located in population centers distributed throughout a state and each of the process servers **140** may execute on one or more workstations **142**.

[0023] The IDE application **110** may include some portions of a COTS rules engine tool. The COTS rules engine tool may be customized and/or extended, for example extended by including the rules engine framework **113**, to provide functionality and convenience otherwise not provided by the COTS rules engine tool. In an embodiment, a Fair Isaac BLAZE ADVISOR COTS rules engine package may be encapsulated in the system **100**. In an embodiment, the IDE application **110** builds on, extends, or encapsulates a rules management application (RMA) portion of the BLAZE ADVISOR COTS. For example, the IDE application **110** extends the RMA by providing the rules engine framework **113** that comprises the generic template framework **114** that may be used to compose rules, the input validation component **116** that promotes automatically providing input validation for the rules defined using the IDE application **110**, and the custom providers **117** that extend the capabilities of the COTS rules engine **146**. As a further extension of the RMA, the generic template framework **114** comprises a plurality of custom templates **115**.

[0024] The rule editor **112** may be executed, for example from the developer workstation **106** or some other workstation, to compose rules, rule sets, and rule flows. In an embodiment, a rule may define a condition linked to a decision action that is performed if the condition evaluates TRUE. The condition comprises a left hand variable associated with a right hand input by a template. The template may be referred to as an operator, the left hand variable may be referred to as a left hand operand, and the right hand input may be referred to as a right hand operand. The decision actions may be used by the business process **144** to make a decision whether to further process a claim, to approve payment of a claim, to reject a claim, or to mark a claim as suspended due to one or more exceptions associated with the claim. The decision actions may be used by the business process **144** to make a decision in registering a new patient and/or healthcare system member in the healthcare information system, registering a new healthcare provider in the healthcare information system, authorizing a procedure, and/or other actions. A healthcare provider may be a physician, a doctor's office, a hospital, a minor emergency care center, a physical therapist, and other facilities. The action of the business process **144** approving payment of a claim, for example, may be conditioned on the rules engine **146** successfully executing a rule that checks that a right hand input being an integer value greater than a threshold value associated with or defined by the left hand variable. In another example, the left hand variable may represent claimant age and the right hand input may be the age of the subject claimant.

[0025] In an embodiment, a rule set may comprise an ordered sequence of rules. In an embodiment, a rule flow may comprise an ordered sequence of rule sets. Rules may be defined, for example, to promote healthcare providers to enroll in a healthcare information system, to promote patients and/or healthcare system members to enroll in a healthcare information system, to promote processing healthcare claims, and other healthcare information system processing. The rules may promote determining eligible procedures and/or codes, determining allowable procedure costs, and determining patient eligibility. By processing the rules, the rules engine **146** may be said to promote enrollment of healthcare providers and healthcare system members and processing of healthcare claims. In an embodiment, the rules engine **146** may promote enrollment of health care providers and patients in a state operated Medicaid program and promote Medicaid claims processing.

[0026] There may be constraints imposed on the values that can be provided to the right hand side input, based on the left hand variable. For example, if the left hand variable represents age, the right hand input may be constrained to being an integer value. Depending upon the specific rule, the right hand input may be constrained to being an integer value in a specific range of values, for example if the left hand variable represents age of a senior citizen, the right hand input may be constrained to being an integer value in the range from 60 to 130, which may be referred to as checking for an appropriate age for a senior citizen. Alternatively, if the left hand variable represents the age of anyone, the right hand input may be constrained to be an integer value in the range from zero to 130, which may be referred to as checking for an appropriate age for a human being. As another example, if the left hand variable represents sex of a patient, the right hand input may be constrained to being a capital "M," a capital "F." a lower case "m," and a lower case "f," where either "M" or "m" indicates a male and either "F" or "f" indicates a female. These constraints may be imposed during execution of the rules engine **146**. When the constraints are not satisfied, the rules engine **146** may return a rules return object that is associated with exception codes, indicating the rule did not evaluate successfully. The business process **144** takes action based on the rules return object and on the exception codes.

[0027] The templates, including custom templates **115**, may be selected using the rule editor **112** from the generic templates **114** to define rules, for example to define conditions comprising a left hand variable associated with a right hand input by a template and/or operator. In some contexts, the generic templates **114** may be referred to as generic operators. A user interface associated with the rule editor **112**, for example a graphical user interface exported to the developer workstation **106**, may provide drop-down menus listing the generic templates **114**. The generic templates **114** have been defined and/or customized to extend the native constructs of the IDE application **110** to promote ease and convenience of composing rules and for enhancing the complexity of expression that may be achieved using rules. The custom templates **115** have been defined to provide further specialized rules definition expressiveness. Some examples of the custom templates **115** include list operators such as IN LIST and NOT IN LIST. Other examples of the custom templates **115** include system parameters, system lists, and valid values. The IDE application **110** may be modified to add additional custom templates **115**, for example by an administrator modifying and/or configuring the IDE application **110** software.

[0028] Various types of custom templates may exist. For example, a system parameter custom template may be provided to define value constraints. The system parameter custom templates may be used to define any constraints within the claims processing framework, such as a maximum annual patient charge for a specific procedure, a maximum allowable amount for a specific procedure, and so forth. In addition, a valid values custom template may be provided to identify valid values for input fields. For example, the valid values custom template may be used to verify the values for the place of service performance are valid. Finally, a systems list custom template may be provided that promotes convenient groupings of valid values. The systems list custom template may define the valid values for given circumstances. For example, system lists may be created for different types of professional services, such as dental services and medical services.

[0029] More specifically, valid values may be used to identify standard valid code values and descriptions for many MMIS fields. Examples of MMIS fields with valid values are place of service, procedure modifier, and category of service. Place of service has many valid values, some of which might be 03 for school, 04 for homeless shelter, 11 for office, 12 for home, and 32 for nursing facility. System lists are used to define different groupings of valid values. One system list may be created to define all of the possible valid value codes for place of service. A second system list may be created for valid places of service for a type of claim, such as an 837 professional claim. A third system list may be created for valid places of service for yet another type of claim, such as an 837 dental claim. A fourth system list may be created for valid places of service for yet another type of claim, such as 837 institutional claims. All of the codes in the second, third and fourth system lists may also be in the first system list. Some, but not all, of the codes in the system list for place of service in the 837 professional claim are also in the system list for place of service in the 837 dental claim. Conversely, some places of service in an 837 dental claim may not be valid in an 837 professional claim. In a specific example, system list 1026 may include all the place of service codes in the Enterprise, and may only contain two valid codes: 11 (Office), and 32 (Nursing Facility). Another system list 1083 may include the valid place of service codes for dental claims, and may only contain one valid code: 11 (Office). A web portal user interface might use system list 1026 to ensure that any valid place of service code can be associated with a procedure code in the reference subsystem. Similarly, a claims processing business rule might use system list 1083 to post an exception if the place of service in a dental claim is not valid.

[0030] Various types of custom providers that extend the COTS rules engine application may exist. For example, the database custom provider promotes retrieval of two or more columns values from the database. Retrieval of two ore more columns may allow the rules to be executed faster and/or more efficiently. In addition, the array object custom provider promotes retrieval of all of the elements of an array in one operation. Such may be useful when checking the status of multiple data elements from a single array to validate a rule. For example, when processing a claim for dental fillings, it may be desirable to check whether the patient has had the filled teeth removed in a prior procedure. Finally, the display name custom provider promotes the use and/or display of data value labels in a consistent format. For example, the display name custom provider causes the label "Last Name" to be displayed with a particular data set, as opposed to code-driven labels, such as "lname," "lastname," or "last_name." The display name custom provider may comprise a table that cross-references the desired display labels (e.g. "Last Name") with the data labels encountered in the code (e.g. "lname," "lastname," or "last_name"). The display name custom provider may over-right the data labels in the code with the display name labels, or may simply access the table and convert the data labels prior to display the labels to the user.

[0031] As the rule is composed using the rule editor **112**, the input value validation component **116** automatically provides an input value validation code that validates input values associated with the right hand input with respect to a type of the left hand variable of the rule. Using the rule editor **112**, rules, rule sets, and rule flows may be composed by non-computer programmer personnel, for example by SMEs in the business logic of the Enterprise application, such as healthcare information system SMEs. For example, the condition of a first rule may be defined to comprise a left hand variable for medical board certification authority, a right hand input for receiving the medical board certification authority under which a specific physician is licensed to practice medicine, and the operator may be the IN LIST custom template. The action associated with the condition of the first rule may be continue registration process. The condition of a second rule may be defined to comprise a left hand variable for years of licensed practice, a right hand input for receiving the years of practice of the specific physician, and the operator may be GREATER THAN. The action associated with the condition of the second rule may be to approve registration of the subject physician. The first rule and the second rule may be associated in a rule flow wherein the first rule must succeed or evaluate TRUE before processing the second rule and then, if the second rule evaluates TRUE, the end action is to return a rule return object with no exceptions. In the case that no exceptions are associated with the rule return object, the business process **144** may approve registration or enrollment of the subject physician in the healthcare information system.

[0032] As the rules, rule sets, and rule flows are composed using the rule editor **112**, they may be stored in the rules repository **108**. In an embodiment, each rule may be stored as a separate file in the rules repository **108**. In another embodiment, however, a plurality of rules that form a rule group may be stored together in a file. In an embodiment, the rules repository **108** may be implemented as a database, for example a relational database, an object-oriented database, or another type of database. In an embodiment, the rules repository **108** may be implemented as a datastore, for example, a flat file system, a directory based file system, or other kind of datastore.

[0033] While the rules engine **146** may execute the rules directly out of the rules repository **108**, run-time advantages may be obtained by compiling or otherwise processing the rules, rule sets, and rule flows using the rule compiler **118** into the compiled rules **148**. In an embodiment, the rules may be compiled and stored as compiled rules in the rules repository **108**. The compiled rules may then be retrieved from the rules repository **108** and loaded into random access memory or other local memory of the process server **140**, represented in FIG. **1** as the compiled rules **148**, when the process server **140** boots or as one or more of the business process **144** and/or the rules engine **146** are initialized. In an embodiment, the compiled rules **148** may be formatted according to the ADB format. This ADB format is a kind of binary format that the

rules engine **146** loads into local memory of the process server **140**, for example random access memory (RAM), when the rules engine **146** initializes. The rules engine **146** may execute more quickly and efficiently when accessing rules for processing out of the local memory of the process server **140** than when accessing rules out of files stored in a secondary storage device, for example the rules repository **108**. When deployed for service, the process server **140** may be accessed by the workstations **142** to perform a variety of operations. In an embodiment, the workstations **142** may invoke functions of a healthcare information system, for example processing Medicaid claims. Invoking the functions of the healthcare information system may involve invoking a procedure of the business process **144**, and performing the invoked procedure of the business process **144** may involve the business process **144** invoking the execution of one of the compiled rules **148** by the rules engine **146**.

[0034] The system **100** promotes rapid revision and deployment of changes of the rules. In an exemplary embodiment, the system **100** may provide a MMIS. The system **100** may promote simple procedures for modifying rules in response to changes in federal regulations and/or changes in state Medicaid policy. The system **100** may promote rapid deployment of the rule changes, for example by compiling rules into ADB format and transferring to the compiled rules **148**. In some state MMIS systems, rule updates may occur two or three times per year. The system **100** has clear utility in concurrently maintaining a plurality of different healthcare management information systems, for example different Medicaid information systems for different states. Different rules defined by the compiled rules **148** may apply in each of the different states using the common business process **144** and rules engine **146**. Additionally, rules in the different states may change at different times in response to local legislation or administrative mandates. Some examples of composing rules using the IDE **110** are described below.

[0035] In an embodiment, picking a system list in constructing a rule can be performed as follows. Suppose the rule to be composed is "If the claim line item place of service is not a valid value, then post exception code 1093." Assume that "dental claim line item place of service" has been defined as the first operand of the condition of the rule, that "not in" has been defined as the operator of the condition of the rule, and that "post exception code 1093" has been defined as the action of the rule. A system list for place of service may be accessed to define the second operand of the condition for the rule. For example, a menu drop-down list that is associated with the second operand may be displayed, from which "system list" may be selected. A drop-down list that is associated with the selection of "system list" is then displayed, from which "reference" may be selected. Next, a drop-down list that is associated with "reference" is displayed, from which "dental claims places of service" may be selected. This selects the system list for valid dental claim places of service as the second operand.

[0036] In an embodiment, picking a valid value in constructing a rule can be performed as follows. Suppose the rule to be composed is "If the claim line item place of service is equal to the valid value for office and the claim line item category of service is physician services, then perform physician pricing." Assume that "claim line item place of service" has been defined as the first operand of the first condition of the rule, that "equal to" has been defined as the operator of the first condition of the rule, that the second condition has been defined as "the claim line item category of service is physician services," and that "perform physician pricing" has been defined as the action of the rule. A valid value for place of service can be accessed to define the second operand of the first condition. For example, a menu drop-down list that is associated with the second operand is displayed, from which "valid value" may be selected. A drop-down list that is associated with the selection of "valid value" is displayed, from which "reference" may be selected. Next, a drop-down list that is associated with the selection of "reference" is displayed, from which "all places of service" may be selected. Finally, a drop-down list that is associated with the selection of "all places of service" may be displayed, from which "office" may be selected. This selects the valid value for the second operand of the first condition of the rule.

[0037] Turning now to FIG. **2**, a method **200** is described. At block **204**, a rule is composed based on selecting one of the generic templates **114**, for example one of the custom templates. The rule composition may also identify a left hand variable and a right hand input, wherein the left hand variable is related to the right hand input by the selected one of the generic templates **114**. For example, a left hand variable may be the claimant age, the right hand input may be the age of a subject claimant, and the generic template **114** selected may be GREATER THAN. At block **208**, input value validation is automatically incorporated into the rule. For example, using the rule editor **112**, the input value validation component **116** is invoked during rule composition to build input validation into the rule. For example, the right hand input for claimant age may be constrained by the input validation built into the rule to be an integer value. At block **212**, if more rules are to be composed, the process **200** loops back to block **204**. If no more rules are to be composed, the process **200** proceeds to block **216**. At block **216**, the rules may be compiled into a format that is suitable for efficient run-time processing by the rules engine **146**, for example into the Fair Isaac ADB binary format.

[0038] At block **220**, the business **144** process is invoked to process a healthcare claim, for example to approve payment of a claim submitted from the workstation **142**. At block **224**, the business process **144** invokes the rules engine **146** to process a rule associated with the business process **144** processing the claim. At block **228**, the rules engine **146** processes the rule, for example invoking the rule from the compiled rules **148** with the input provided by the business process **144**. The rules engine **146** returns a rule return object to the business process **144** that indicates the result of processing the rule. At block **232** the business process completes, for example approving payment the claim based on the rule return object.

[0039] As described above, rules engines may be pluggable software components that execute rules that have been externalized from application code. The rules define business rules and/or business logic that may change frequently. Typically, rules may be defined by nonprogrammers and may be provided to the rules engine in the form of data or data files. Using a rules engine to provide business rules to an application may reduce time to market and reduce total cost of ownership, with reference to the alternative of encoding the business logic in high level programming language code.

[0040] Some aspects of the system **100** described above may be implemented on any general-purpose computer with sufficient processing power, memory resources, and network throughput capability to handle the necessary workload

placed upon it. For example, the computer system **102**, the developer workstation **106**, the process server **140**, and the workstation **142** each may be implemented as a general-purpose computer system. FIG. **3** illustrates a typical, general-purpose computer system suitable for implementing one or more embodiments disclosed herein. The computer system **380** includes a processor **382** (which may be referred to as a central processor unit or CPU) that is in communication with memory devices including secondary storage **384**, read only memory (ROM) **386**, RAM **388**, input/output (I/O) devices **390**, and network connectivity devices **392**. The processor **382** may be implemented as one or more CPU chips.

[0041] The secondary storage **384** is typically comprised of one or more disk drives or tape drives and is used for non-volatile storage of data and as an over-flow data storage device if RAM **388** is not large enough to hold all working data. Secondary storage **384** may be used to store programs that are loaded into RAM **388** when such programs are selected for execution. The ROM **386** is used to store instructions and perhaps data that are read during program execution. ROM **386** is a non-volatile memory device, which typically has a small memory capacity relative to the larger memory capacity of secondary storage **384**. The RAM **388** is used to store volatile data and perhaps to store instructions. Access to both ROM **386** and RAM **388** is typically faster than to secondary storage **384**.

[0042] I/O devices **390** may include printers, video monitors, liquid crystal displays (LCDs), touch screen displays, keyboards, keypads, switches, dials, mice, track balls, voice recognizers, card readers, paper tape readers, or other well-known input devices.

[0043] The network connectivity devices **392** may take the form of modems, modem banks, Ethernet cards, universal serial bus (USB) interface cards, serial interfaces, token ring cards, fiber distributed data interface (FDDI) cards, wireless local area network (WLAN) cards, radio transceiver cards such as code division multiple access (CDMA), global system for mobile communications (GSM), and/or worldwide interoperability for microwave access (WiMAX) radio transceiver cards, and other well-known network devices. These network connectivity devices **392** may enable the processor **382** to communicate with an Internet or one or more intranets. With such a network connection, it is contemplated that the processor **382** might receive information from the network, or might output information to the network in the course of performing the above-described method steps. Such information, which is often represented as a sequence of instructions to be executed using processor **382**, may be received from and outputted to the network, for example, in the form of a computer data signal embodied in a carrier wave.

[0044] Such information, which may include data or instructions to be executed using processor **382** for example, may be received from and outputted to the network, for example, in the form of a computer data baseband signal or signal embodied in a carrier wave. The baseband signal or signal embodied in the carrier wave generated by the network connectivity devices **392** may propagate in or on the surface of electrical conductors, in coaxial cables, in waveguides, in optical media, for example optical fiber, or in the air or free space. The information contained in the baseband signal or signal embedded in the carrier wave may be ordered according to different sequences, as may be desirable for either processing or generating the information or transmitting or receiving the information. The baseband signal or signal embedded in the carrier wave, or other types of signals currently used or hereafter developed, referred to herein as the transmission medium, may be generated according to several methods well known to one skilled in the art.

[0045] The processor **382** executes instructions, codes, computer programs, scripts which it accesses from hard disk, floppy disk, optical disk (these various disk based systems may all be considered secondary storage **384**), ROM **386**, RAM **388**, or the network connectivity devices **392**. While only one processor **382** is shown, multiple processors may be present. Thus, while instructions may be discussed as executed by a processor **382**, the instructions may be executed simultaneously, serially, or otherwise executed by one or multiple processors **382**.

[0046] While several embodiments have been provided in the present disclosure, it should be understood that the disclosed systems and methods might be embodied in many other specific forms without departing from the spirit or scope of the present disclosure. The present examples are to be considered as illustrative and not restrictive, and the intention is not to be limited to the details given herein. For example, the various elements or components may be combined or integrated in another system or certain features may be omitted or not implemented.

[0047] In addition, techniques, systems, subsystems, and methods described and illustrated in the various embodiments as discrete or separate may be combined or integrated with other systems, modules, techniques, or methods without departing from the scope of the present disclosure. Other items shown or discussed as directly coupled or communicating with each other may be indirectly coupled or communicating through some interface, device, or intermediate component, whether electrically, mechanically, or otherwise. Other examples of changes, substitutions, and alterations are ascertainable by one skilled in the art and could be made without departing from the spirit and scope disclosed herein.

What is claimed is:

1. An apparatus, comprising:

a computer system; and

an integrated development environment application that, when executed on the computer system, promotes defining a plurality of rules based on a plurality of generic templates and automatically incorporating input value validation based on a type of a left hand variable into the rules,

wherein the rules, when executed by a healthcare management information system rules engine, promote enrollment of health care providers and healthcare system members and processing of healthcare claims.

2. The apparatus of claim **1**, wherein the generic templates comprise custom templates comprising at least one of a generic template, system lists custom template, a system parameters custom template, and a valid values custom template.

3. The apparatus of claim **1**, wherein the generic templates comprise an IN LIST generic template and a NOT IN LIST generic template.

4. The apparatus of claim **1**, wherein integrated development environment provides a plurality of custom providers.

5. The apparatus of claim **4**, wherein the custom providers comprise at least one of a database custom provider and an array object custom provider.

6. The apparatus of claim **1**, wherein the rules are defined at least in part as the left hand variable associated with a right hand input based on the generic templates.

7. A method of processing a healthcare claim, comprising:

composing a rule from at least one of a generic template, system parameters custom template, a system list custom template, and a valid values custom template;

invoking a business process function to process a healthcare claim;

wherein the business process invokes the rule to be processed by a rules engine;

wherein the rules engine returns the result of processing the rule in a rule return object; and

wherein the business process approves payment or suspends the healthcare claim based on the rule return object.

8. The method of claim **7**, further comprising compiling the rule into an advanced database binary format, wherein the rules engine processes the rule in the advanced database binary format.

9. The method of claim **7**, wherein the business process and the rules engine are part of a healthcare management information system.

10. The method of claim **9**, wherein the healthcare management information system is a Medicaid management information system.

11. The method of claim **7**, wherein the rule comprises at least one of an IN SET operator, and a NOT IN SET operator.

12. The method of claim **7**, wherein the rule comprises at least one of a NOT IN operator, IS IN operation, greater than operator, less than operator, equal to operator, less than or equal to operator, greater than or equal to operator.

13. The method of claim **7**, wherein the rule comprises at least one of an ADD operator, a SUBTRACT operator, a MULTIPLY operator, and a DIVIDE operator.

14. A system, comprising:

a computer system; and

a rules engine framework that, when executed on the computer system,

promotes composing a plurality of rules, wherein each rule is composed at least in part from a left hand variable associated with a right hand input by an operator, and automatically incorporating input value validation of the right hand input based on the type of the left hand variable, wherein the at least some of the rules are composed based on at least one of a generic template, system list custom template, a system parameters custom template, and a valid values custom template, and

builds a healthcare management information system rules engine based at least in part on a plurality of custom providers,

wherein the healthcare management information system rules engine promotes enrollment of health care providers and patients and promotes healthcare claims processing.

15. The system of claim **14**, wherein the custom providers comprise a database custom provider and an array object custom provider.

16. The system of claim **15**, wherein the database custom provider extends the off-the-shelf functionality of healthcare management information system rules engine by enabling reading two or more columns of data from a database during one read transaction.

17. The system of claim **15**, wherein the database custom provider extends the off-the-shelf functionality of the healthcare management information system rules engine by enabling access to the elements of an array object.

18. The system of claim **14**, wherein the healthcare management information system rules engine promotes enrollment of health care providers and patients in a state operated Medicaid program and promotes Medicaid claims processing.

19. The system of claim **14**, wherein the value checking comprises verifying that age values are in the range from 0 to 130.

20. The system of claim **19**, wherein the rules engine framework further promotes composing a rule based on at least one of an IN LIST operator and a NOT IN LIST operator.

* * * * *