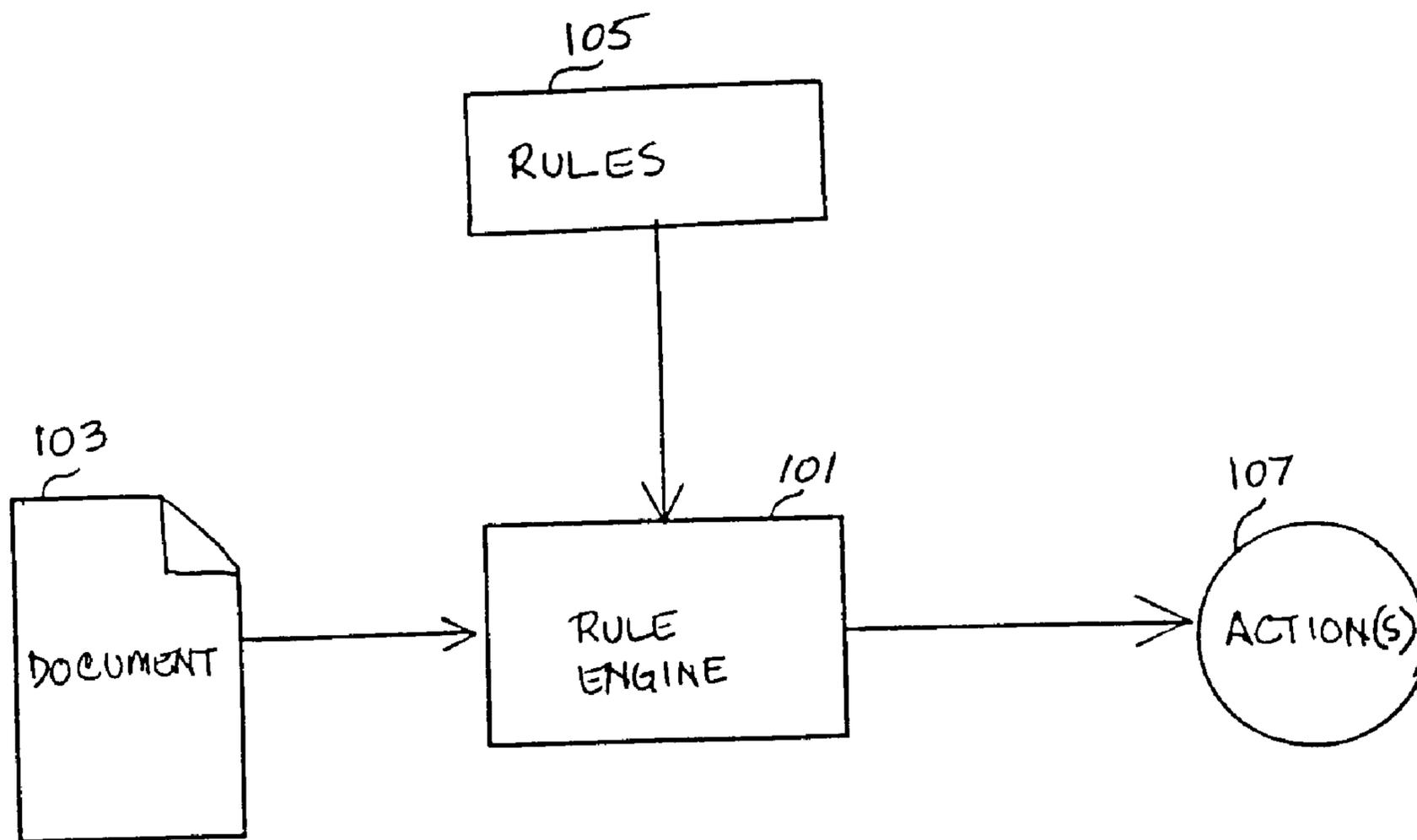




(86) Date de dépôt PCT/PCT Filing Date: 2000/12/22
 (87) Date publication PCT/PCT Publication Date: 2001/06/28
 (45) Date de délivrance/Issue Date: 2010/09/07
 (85) Entrée phase nationale/National Entry: 2002/06/03
 (86) N° demande PCT/PCT Application No.: US 2000/034854
 (87) N° publication PCT/PCT Publication No.: 2001/046839
 (30) Priorité/Priority: 1999/12/22 (US09/469,753)

(51) Cl.Int./Int.Cl. *G06F 17/00* (2006.01),
G06F 21/00 (2006.01)
 (72) Inventeurs/Inventors:
MARTIN, PAUL E., US;
DAVIS, KENNETH S., US;
KYAW, MYO, US;
KALIAPPAN, ARULNAMBI, US;
SHVADSKIY, DMITRY F., US
 (73) Propriétaire/Owner:
SEQUOIA SOFTWARE CORPORATION, US
 (74) Agent: BERESKIN & PARR LLP/S.E.N.C.R.L.,S.R.L.

(54) Titre : PROCÉDE ET SYSTÈME DE SÉCURITÉ, ROUTAGE ET EXECUTION D'ACTIONS BASES SUR LE
CONTENU D'UN DOCUMENT
 (54) Title: METHOD AND SYSTEM FOR CONTENT-BASED DOCUMENT SECURITY, ROUTING, AND ACTION
EXECUTION



(57) Abrégé/Abstract:

A method and system for determining document content based actions. A rule engine (101) is provided with a document (103) to be acted on, preferably in XML format. The rule engine (101) is also provided with a set of rules (105) and associated actions (107) to be taken upon special conditions. The document (103) is evaluated according to the rules (105) and the document contents, each of the rules is parsed into a parse tree. The document is further evaluated for populating the parse tree with at least a portion of the document contents. The portion of the actions to be initiated is selected based on the content-populated parse tree. Actions could include by way of example, security control or routing of the document.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

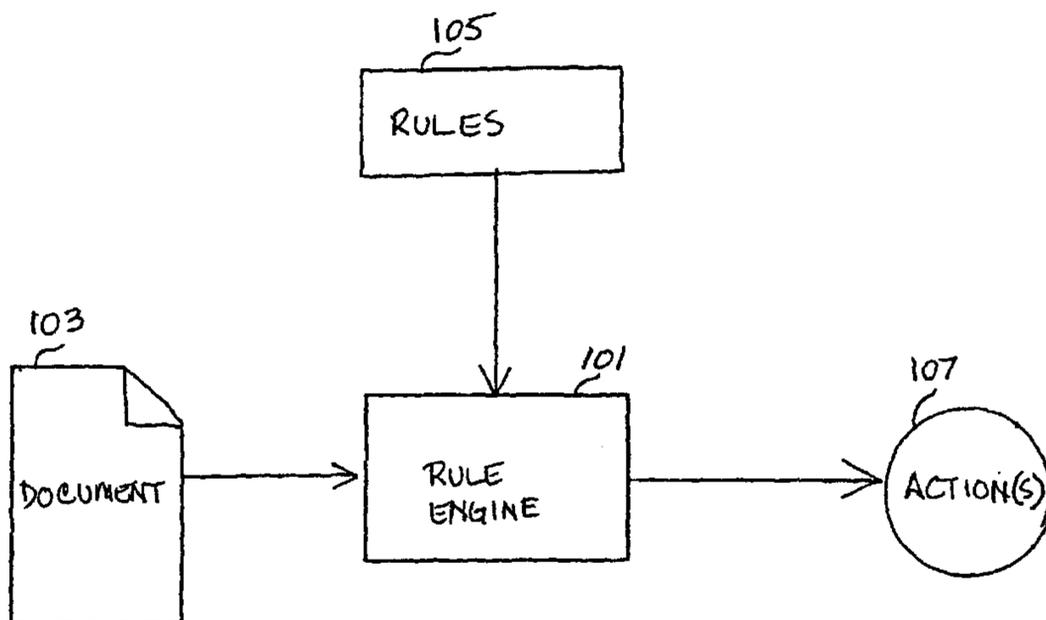
(19) World Intellectual Property Organization
International Bureau(43) International Publication Date
28 June 2001 (28.06.2001)

PCT

(10) International Publication Number
WO 01/46839 A1

- (51) International Patent Classification⁷: G06F 17/00
- (21) International Application Number: PCT/US00/34854
- (22) International Filing Date:
22 December 2000 (22.12.2000)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
09/469,753 22 December 1999 (22.12.1999) US
- (71) Applicant: SEQUOIA SOFTWARE CORPORATION
[US/US]; 8890 McGaw Road, Columbia, MD 21045 (US).
- (72) Inventors: MARTIN, Paul, E.; Columbia, Columbia, MD 21045 (US). DAVIS, Kenneth, S.; Columbia, MD 21045 (US). KYAW, Myo; Columbia, MD 21045 (US). KALIAPPAN, Arulnambi; Columbia, MD 21045 (US). SHVADSKIY, Dmitry, F.; Columbia, MD 21045 (US).
- (74) Agent: NICHOLSON, Cynthia, K.; Piper Marbury Rudnick & Wolfe LLP, 1200 19th Street, NW, Washington, DC 20036 (US).
- (81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.
- (84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).
- Published:**
- With international search report.
 - Before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments.
- For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.*

(54) Title: METHOD AND SYSTEM FOR CONTENT-BASED DOCUMENT SECURITY, ROUTING, AND ACTION EXECUTION



(57) Abstract: A method and system for determining document content based actions. A rule engine (101) is provided with a document (103) to be acted on, preferably in XML format. The rule engine (101) is also provided with a set of rules (105) and associated actions (107) to be taken upon special conditions. The document (103) is evaluated according to the rules (105) and the document contents, each of the rules is parsed into a parse tree. The document is further evaluated for populating the parse tree with at least a portion of the document contents. The portion of the actions to be initiated is selected based on the content-populated parse tree. Actions could include by way of example, security control or routing of the document.

WO 01/46839 A1

5 **METHOD AND SYSTEM FOR CONTENT-BASED DOCUMENT SECURITY,
ROUTING, AND ACTION EXECUTION**

BACKGROUND OF THE INVENTION

1. FIELD OF THE INVENTION

 The present invention relates to a computer-implemented rules engine. More
10 particularly, it concerns a method and system that allows end-users to process documents
according to rules, based on the content of documents, and to execute actions, dependent upon
how the rules evaluate.

2. DESCRIPTION OF THE RELATED ART

 While there have been other rules engines, they all rely on meta data. Meta data based
15 rule engines require that documents be pre-processed to obtain the meta data or that meta data be
manually specified by an end-user. This meta data is then fed to the rule engine for evaluation.

 Thus, a document to be routed according to a conventional system would include, for
example, a header containing the data relevant to routing. Further, a document intended to be
handled in accordance with certain security rules would include a header containing security-
20 related data. A conventional header would include data related to other actions to be taken (or
not taken) on a document.

 Pre-processing of the document is used to determine the data or meta data to be placed in
a header, or otherwise stored outside of the document contents, but associated with the
document. Thus, in routing a document or otherwise action on a document, the pre-processed
25 meta data is principally used.

BRIEF SUMMARY OF THE INVENTION

According to the invention, there is provided a method and system for determining, in a computerized system, document content based actions. A rule engine is provided with a document to be acted on, the document having document content, wherein the document content is defined to include at least one delimited content field, and the document is defined to be in XML format. A rule grammar, including at least one rule operator, at least one constant, and at least one content field, is provided. The rule engine is provided with a set of rules and associated actions to be taken upon special conditions; wherein at least one rule in the set of rules includes at least a specified rule operator, a specified constant, and a specified content field; and a plurality of rule objects, a rule object containing a rule of the set of rules, and referencing one or more next rules to execute based on an evaluation of the document against the rule, and referencing one or more actions to be initiated based on the evaluation;. The document is evaluated according to the rules and the document contents; each of the rules is parsed into a parse tree, the document is further evaluated, including, for the provided document, populating the parse tree with at least a portion of the document contents, to create a content-populated parse tree. The content-populated parse tree is evaluated to a determinant. At least a portion of the actions associated with the rules, the portion of the actions to be initiated is selected based on the determinant.

20 These and other objects, features and advantages of the present invention are readily apparent from the following drawings and detailed description.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention is described in connection with the drawings, in which:

FIG. 1 is a flow diagram illustrating the method and system for content-based document security, routing and action execution;

5 FIG. 2 is a block diagram illustrating a rule object;

FIG. 3 illustrates an example of the parse tree resulting from parsing a document according to a rule;

FIG. 4 illustrates the parse tree of Fig. 3, as populated with document contents;

FIGS. 5a-5c illustrate an inorder traversal of the parse tree of Fig. 4.

10

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

A content based rules engine can evaluate rules based on the content contained within documents without relying on pre-processed data extracted from the document. For example, using the rules engine, a user can write the following rule: 'If the value of the first name field in the document is "john" and the value of the last name field in the document is "doe" then put the document in the Review queue.'

Fig. 1 illustrates the rule engine's operation in general. The rule engine 101 is supplied with a document 103 to be acted on. Also supplied to the rule engine 101 is a set of rules 105, indicating actions to be taken upon certain conditions. One or more actions 107 are then initiated by the rules engine 101.

The rule engine 101 is primed with a set of rules 105, and then for each document 103 processed, it performs a set of actions 107. The document 103 is preferably provided in XML (extended markup language) format.

The rule engine 101 should operate according to a pre-determined rule grammar.

Implementing and operating the rule engine consists of the following.

- 1.) Determine a rule grammar. The specific grammar chosen is not of importance, so long as boolean operators, constants, and delimited content-fields can be distinguished from one another. It is preferred to use the following rule grammar:

(token \$operator\$ token) [\$operator\$ token \$operator\$ token]

where

- 10 token is either a quoted constant of the form "constant-value" or a % delimited content field of the form %content-field%

and

- token is either a quoted constant of the form "constant-value" or a % delimited content field of the form %content-field%

- 15 and

operator is one of the following:

\$eq\$ - equals

\$lt\$ - less than

\$gt\$ - greater than

- 20 \$le\$ - less than or equal

\$ge\$ - greater than or equal

\$and\$ - logical and operation

\$or\$ - logical or operation

\$not\$ - logical not operation

The tokens and operators could be in a form different from the foregoing.

Table 1 depicts a sample document, highly simplified for illustration purposes, and a valid rule.

5 **Table 1:**

Document:

<xml>

<fname>john</fname>

<lname>doe</lname>

10 </xml>

Rule:

%fname% \$eq\$ "john" Sand\$ %lname% \$eq\$ "doe"

15 This example is for illustration purposes. The same principles will apply in a document of any size, and a set of rules of any size. Here, the document is in conventional XML format, and thus the fields within the document are delimited.

A rule condition may be an empty set, or a null operator, etc. The rules check conditions.

20 2.) Define a rule object. The rule object 201 is used to contain a specified rule, to reference the next true rule 207 and false rule 209 to execute if it evaluates to true or false, to reference the next true rule 207 and false rule 209 to evaluate if it evaluates to true or false, to contain a list of true actions 203 to perform if it evaluates to true, and to contain a list of false actions 205 to evaluate if it evaluates to false.

Fig. 2 illustrates a rule object. Here, the rule object 201 relates to the example of Table 1. The rule object 201 contains the rule %fname% \$eq\$ "john". This rule was pre-determined,

preferably by a user. The rule object 201 contains, opints to 211, lists, or otherwise indicates one or more true actions 203 to be initiated if the rule, as applied against the document, evaluates to true. The rule object is associated 213 with one or more false actions 205. An appropriate action is a null, or “do nothing”. In this example, if the rule evaluates to true, the action is to “send
5 email to john@company.com.”

The rule object 201 also references the next rule(2), or rule object, to be executed. In this example, the true rule 207 is %name% \$eq\$ “doe”, and the false object 209 is %fname% \$eq\$ “jane”.

3.) Initialize the Rule Engine. The rule engine 101 (shown in Fig. 1) is provided with
10 a set of rule objects.

4.) Parse the rules. When given a document 103 (shown in Fig. 1) to evaluate, the rule engine 101 will parse the rules into a parse tree. Generating the parse tree may be done in any appropriate manner, such as by commercially available grammar parsers, e.g., “Visual Parser”.

15 The parse tree illustrated in Fig. 3 would result from parsing the rule in Table 1. The tree includes operators 301, content fields 303, and constant values 305. A larger rule would result in a more complex parse tree,

There will be one such parse tree contained within each rule object. Thus, a set of rules will result in a set of parse trees, each corresponding to a rule object.

20 As an alternative to parsing the set of rules each time a document is provided, the rule engine could process the rules and re-use the parse tree for additional documents.

5.) Evaluate Rules. For the provided document, starting with the first rule object, evaluate the parse tree to populate document content values. This advantageously is done by

performing a depth-first inorder traversal of the parse tree, and for each encountered node which contains a content field, substitute the specified value from the document 307 into the tree.

The provided document has contents including values that belong to an identifiable type. Preferably, the document is in XML format. Thus, the specified value for that type may be readily located within the document contents, in any conventional manner.

After performing this operation for the rule and document, the content-populated parse tree will be as in Fig. 4.

Once all content fields in the parse tree have been substituted with document values 307, the boolean logic of the rule is evaluated, preferably by performing a depth-first inorder traversal of the parse tree. For each encountered node (301, 305, 307) which contains an operator 301, the operator node advantageously is replaced with the result of applying the boolean operation specified. This is discussed in detail below in reference to Figs. 5a-5c. The ultimate result of evaluating the rule (the content-populated parse tree) is a determinant, preferably a boolean value. One of skill in the art will appreciate other manners in which a parse tree may be evaluated.

6.) Execute Actions. At this point, according to the preferred embodiment, the parse tree will only be a single node of interest, the determinant. The determinant indicates how the rule evaluated. The value of the determinant determines the actions to be executed. In the preferred embodiment, this indicates whether the rule evaluates to true or false. If the rule is true, then the rule engine initiates execution of the configured set of true actions. If the rule is false, then the rule engine initiates the configured set of false actions.

7.) Go to Next Rule. After the actions for the present rule have been executed, the rule engine goes to the next rule associated with the determinant value, if any, and repeats step 5.

(Actions are further discussed below.) This process continues until there are no more rules to be processed.

Figs. 5a-5c show the result of performing the rule evaluation of a content-populated parse tree, by way of reference to the example of Table 1, and the content-populated parse tree illustrated in Fig. 4. In a depth first inorder traversal for this example, the document value 307 of John and operator 301 of \$eq\$ are applied to the constant value 305 of John, and the operation result 309 is true. Advantageously, the operation result 309 replaces the operator 301 node of the parse tree (Fig. 5a). Next, the document value 307 of Doe and operator 301 of \$eq\$ are applied to the constant value 305 of Doe, and the operation result 309 is true (Fig. 5b). Next, the operation results 309 of true are operated on by operand 301 \$and\$, and the operation result 309 is true (Fig. 5c).

Actions can be any computing operation. In the presently preferred embodiment, the rule engine initiates security or document routine.

The rule engine may be used to perform content based document security by controlling which operations may be performed by which users on documents. For content-based security, the set of actions preferably includes the following: Grant Read Access, Grant Write Access, Grant Update Access, Grant Delete Access, Grant Search Access, Deny Read Access, Deny Write Access, Deny Update Access, Deny Delete Access, Deny Search Access. Other security actions are also possible. These actions can be implemented in any conventional manner, and can be initiated by the rules engine in any of the usual manners to initiate such actions. When a user tries to perform an operation (e.g., Read, Write, Update, Delete, or search) on a document, the rules engine executes the configured set of security rules to determine if the user is allowed to perform the operation.

The actual implementation of each of the above actions will be well understood by one of skill in the relevant art. In the preferred embodiment, each action sets a boolean variable to either true or false, indicating Grant or Deny.

The engine is also used to perform content based document routing. In the presently preferred embodiment, the following actions are provided: Send Email and Move Document to Queue. Other document routine actions are possible. The rule engine can be invoked, for example, when a document is created or updated, to evaluate document routine rules against that document to determine if the document should be placed in a particular queue or emailed to a user. Other document routing actions are possible, and the above list is not exhaustive.

The implementation of each of the above actions themselves is conventional, as will be understood by one of skill in the art. In the preferred embodiment, commercially available products are used to implement the actions: Send Email is implemented using "Ultimate TCP/IP" from Dundas Software; Move Document to Queue is implemented using "Message Queue" from Microsoft.

Sets of other actions and associated sets of rules may be provided. It is the intention that the principles of the present invention will accommodate other actions and rules. Indeed, the rule engine is sufficiently flexible to accommodate documents in a wide variety of formats, other sets of rules, and other sets of actions. Such is anticipated, and is taught by the foregoing, and is within the spirit and scope of the present invention.

While the preferred mode and best mode for carrying out the invention have been described, those familiar with the art to which this invention relates will appreciate that various alternative designs and embodiments for practicing the invention are possible, and will fall within the scope of the following claims.

1. A method for determining, in a computerized system, document content based actions, comprising the steps of:
 - (A) providing, to a rule engine, a document to be acted on, the document having document content and the document is defined to be in extensible markup language (XML) format;
 - (B) providing, to the rule engine, a first rule and a first action to be taken upon a first condition;
 - (C) providing, to the rule engine, a second rule and a second action to be taken upon a second condition;
 - (D) evaluating the document contents to determine the existence of one of the first condition and the second condition; and
 - (E) initiating the first action when the first condition exists and initiating the second action when the second condition exists.
2. The method as claimed in claim 1, wherein the document content is defined to comprise at least one delimited content field.
3. The method as claimed in claim 1, comprising the step of providing a rule grammar for use by the rule engine.
4. The method as claimed in claim 3, wherein the rule grammar comprises at least one rule operator, at least one constant, and at least one content field.

5. The method as claimed in claim 4, wherein the content field is delimited.
6. The method as claimed in claim 4, wherein at least one rule in the set of rules comprises at least a specified rule operator, a specified constant, and a specified content field.
7. The method as claimed in claim 3, wherein at least one rule in the set of rules is in the form of the rule grammar.
8. The method as claimed in claim 1, wherein at least one rule is empty.
9. The method as claimed in claim 1, comprising a plurality of rule objects, a rule object containing a rule of the set of rules, and referencing one or more next rules to execute based on an evaluation of the document against the rule, and referencing one or more actions to be initiated based on the evaluation.
10. The method as claimed in claim 1, further comprising the step of parsing each of the rules into a parse tree.
11. The method as claimed in claim 10, wherein the step of evaluating the document comprises, for the provided document, populating the parse tree with at least a portion of the document contents, to create a content-populated parse tree.

12. The method as claimed in claim 11, wherein the step of evaluating the document comprises evaluating the content-populated parse tree to a determinant.
13. The method as claimed in claim 12, wherein the portion of the actions to be initiated is selected based on the determinant.
14. The method as claimed in claim 1, wherein the first action and the second action are associated with document security or document routing.
15. The method as claimed in claim 1, comprising processing any next rule in the set of rules.
16. The method as claimed in claim 1, wherein the set of actions control security for the provided document.
17. The method as claimed in claim 1, wherein the set of actions control routing of the provided document.
18. A method for determining, in a computerized system, document content based actions, comprising the steps of:
 - (A) providing, to a rule engine, a document to be acted on, the document having document content, wherein the document content is defined to comprise at least

- one delimited content field, and the document is defined to be in extensible markup language (XML) format;
- (B) providing a rule grammar, comprising at least one rule operator, at least one constant, and at least one content field;
- (C) providing, to the rule engine, a set of rules and associated actions to be taken upon special conditions and a plurality of rule objects; wherein at least one rule in the set of rules comprises at least a specified rule operator, a specified constant, and a specified content field; and wherein a rule object of the plurality of rule objects contains a rule of the set of rules, and references one or more next rules to execute based on an evaluation of the document against the rule, and references one or more actions to be initiated based on the evaluation;
- (D) evaluating the document according to the rules and the document contents; parsing each of the rules into a parse tree, evaluating the document comprises, for the provided document, populating the parse tree with at least a portion of the document contents, to create a content-populated parse tree, and evaluating the content-populated parse tree to a determinant; and
- (E) initiating at least a portion of the actions associated with the rules, the portion of the actions to be initiated is selected based on the determinant.

19. A computerized system for determining document content based actions, comprising:

a set of rules comprising a first rule to evaluate XML document content associated with a first condition and a first action to be taken and a second rule to evaluate XML

document content associated with a second condition and a second action to be taken;

a document evaluator processing document content to determine the existence of one of a the first condition and the second condition; and
a rule engine initiating the first action when the first action exists and initiating the second action when the second condition exists.

20. The system as claimed in claim 19, wherein the document content is defined to comprise at least one delimited content field.
21. The system as claimed in claim 19, comprising a rule grammar for the rule.
22. The system as claimed in claim 21, wherein the rule grammar comprises at least one rule operator, at least one constant, and at least one content field.
23. The system as claimed in claim 22, wherein the content field is delimited.
24. The system as claimed in claim 22, wherein at least one rule in the set of rules comprises at least a specified rule operator, a specified constant, and a specified content field.
25. The system as claimed in claim 21, wherein at least one rule in the set of rules is in the form of the rule grammar.

26. The system as claimed in claim 19, wherein at least one rule is empty.
27. The system as claimed in claim 19, further comprising a plurality of rule objects, a rule object containing a rule of the set of rules, and referencing one or more next rules to execute based on an evaluation of the document against the rule, and referencing one or more actions to be initiated based on the evaluation.
28. The system as claimed in claim 19, comprising a parse tree corresponding to each of the rules.
29. The system as claimed in claim 28, wherein, during a stage of the evaluation, the parse tree is populated with at least a portion of the document contents, to create a content populated parse tree.
30. The system as claimed in claim 29, wherein, after a stage of the evaluation, the content-populated parse tree evaluates to a determinant.
31. The system as claimed in claim 30, wherein the portion of the actions to be initiated is selected based on the determinant.
32. The system as claimed in claim 19, wherein the first action and the second action are associated with document security or document routing.

33. The system as claimed in claim 19, wherein the set of actions control security for the provided document.
34. The system as claimed in claim 19, wherein the set of actions control routing of the provided document.
35. A system for determining, in a computerized system, document content based actions, comprising:
- (A) a rule engine, responsive to a document to be acted on, the document defined to have document content, wherein the document content is defined to comprise at least one delimited content field, and the document is defined to be in extensible markup language (XML) format;
 - (B) a rule grammar, comprising at least one rule operator, at least one constant, and at least one content field;
 - (C) a set of rules and associated actions to be taken upon special conditions, wherein at least one rule in the set of rules comprises at least a specified rule operator, a specified constant, and a specified content field; and a plurality of rule objects, a rule object containing a rule of the set of rules, and referencing one or more next rules to execute based on an evaluation of the document against the rule, and referencing one or more actions to be initiated based on the evaluation;
 - (D) a parse tree populated with at least a portion of the document contents, and a determinant, corresponding to the evaluated content-populated parse tree

wherein a portion of the actions to be initiated is selected based on the determinant.

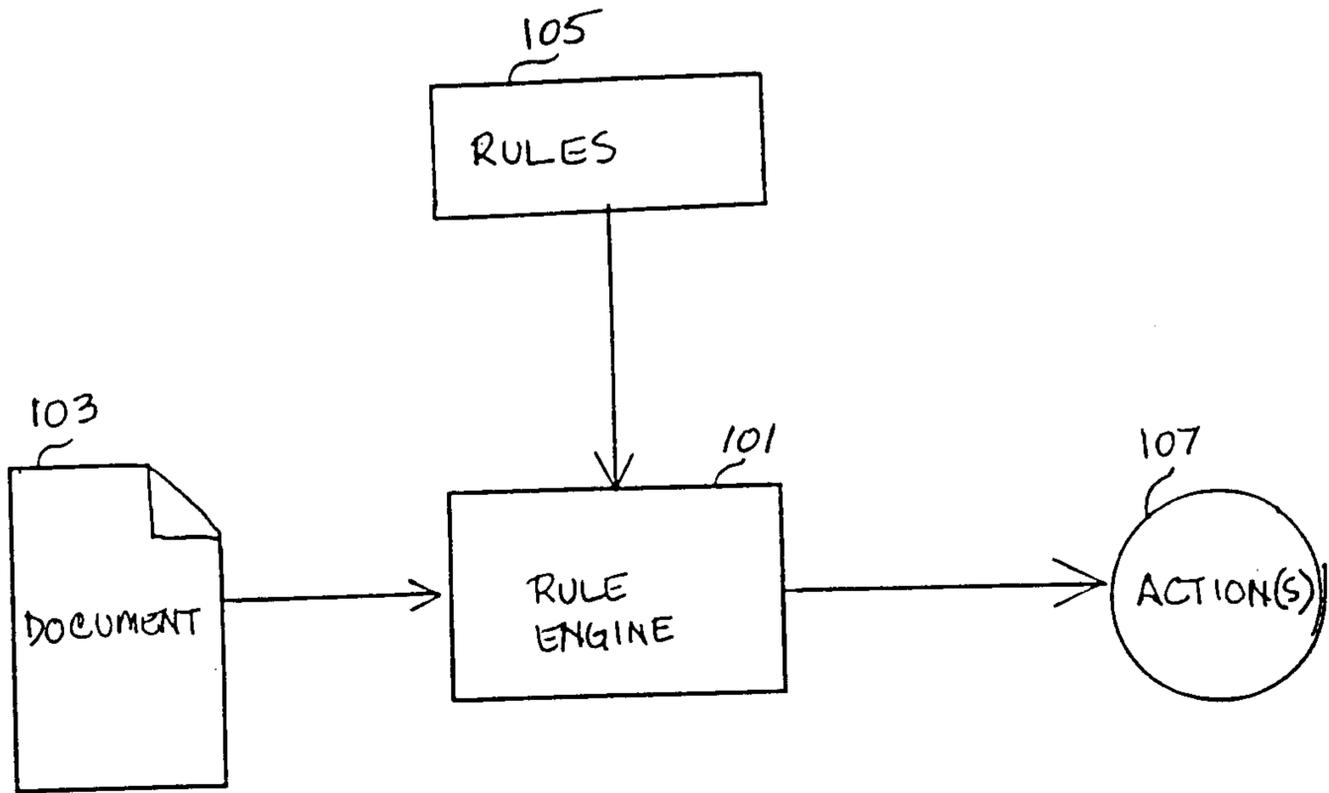


Fig. 1

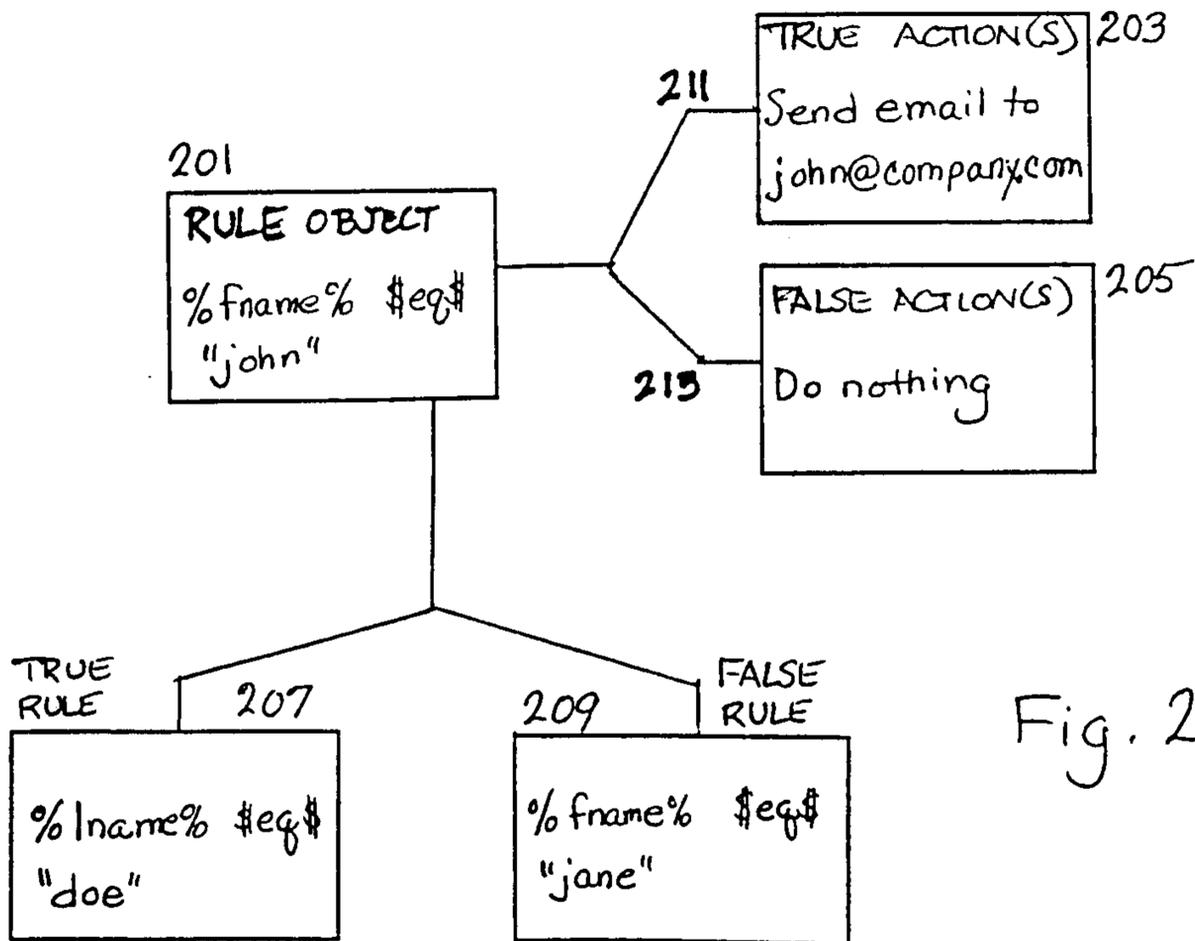


Fig. 2

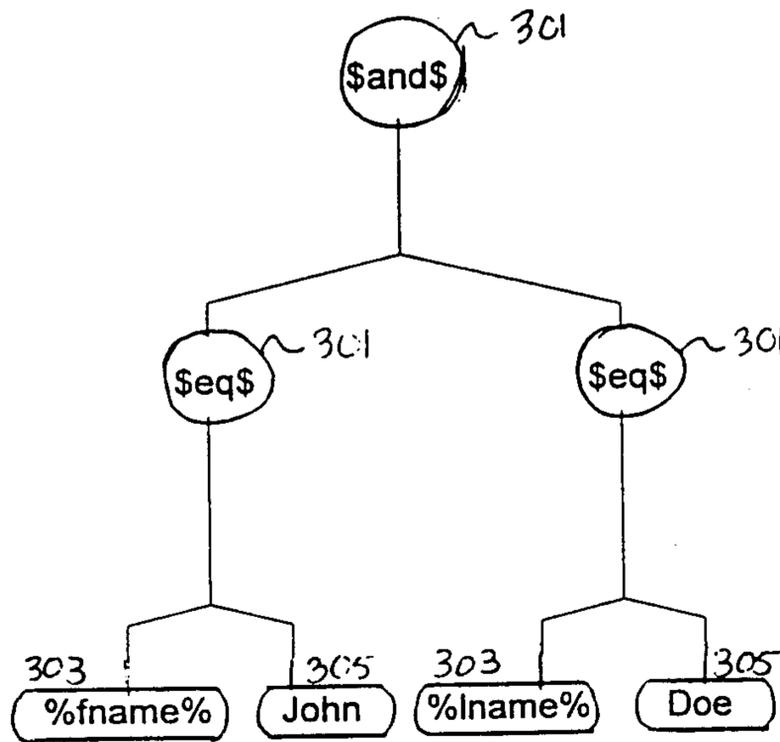


Fig. 3

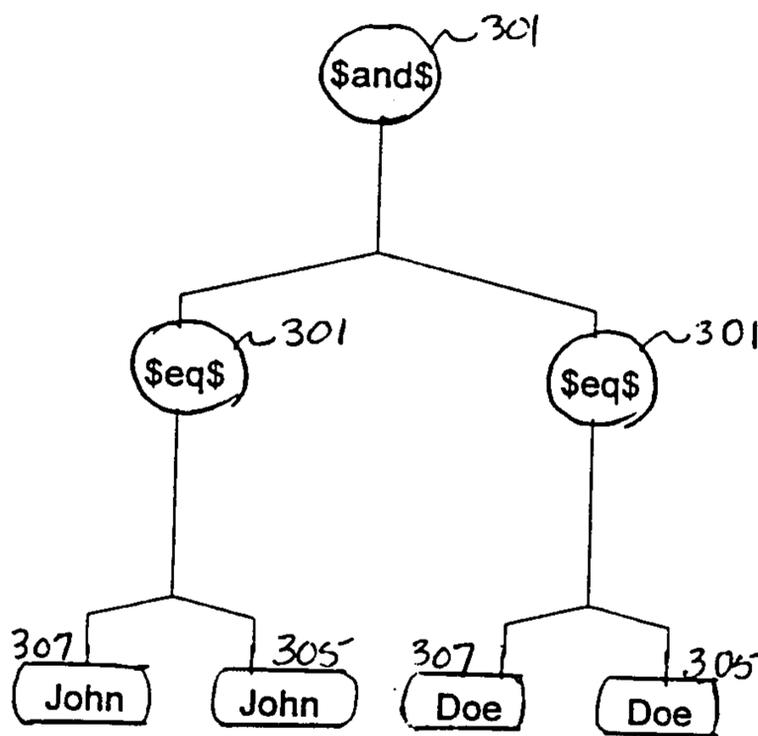


Fig. 4

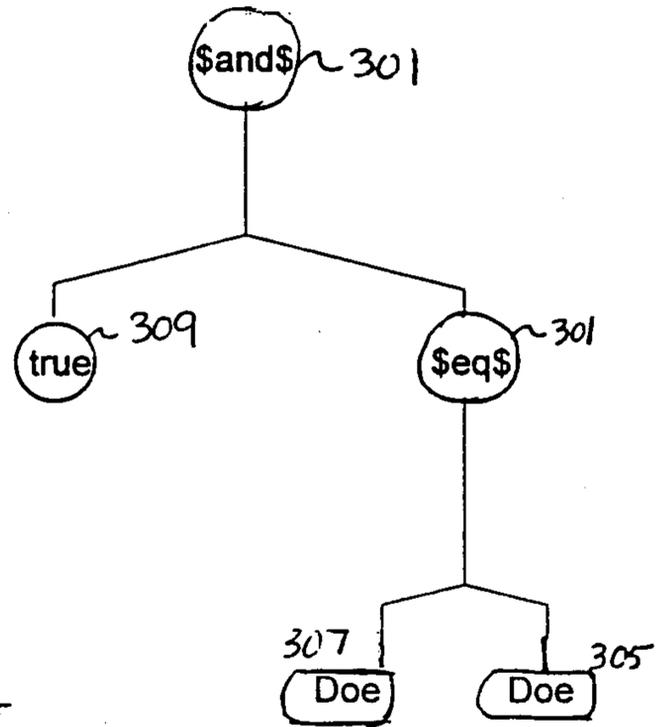


Fig. 5a

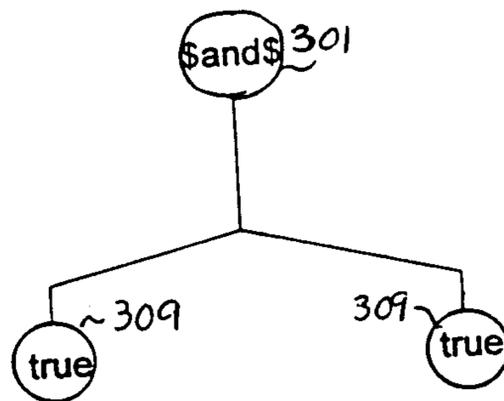


Fig. 5b

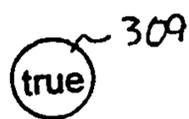


Fig. 5c

