



(22) Date de dépôt/Filing Date: 2004/08/18
(41) Mise à la disp. pub./Open to Public Insp.: 2006/02/18

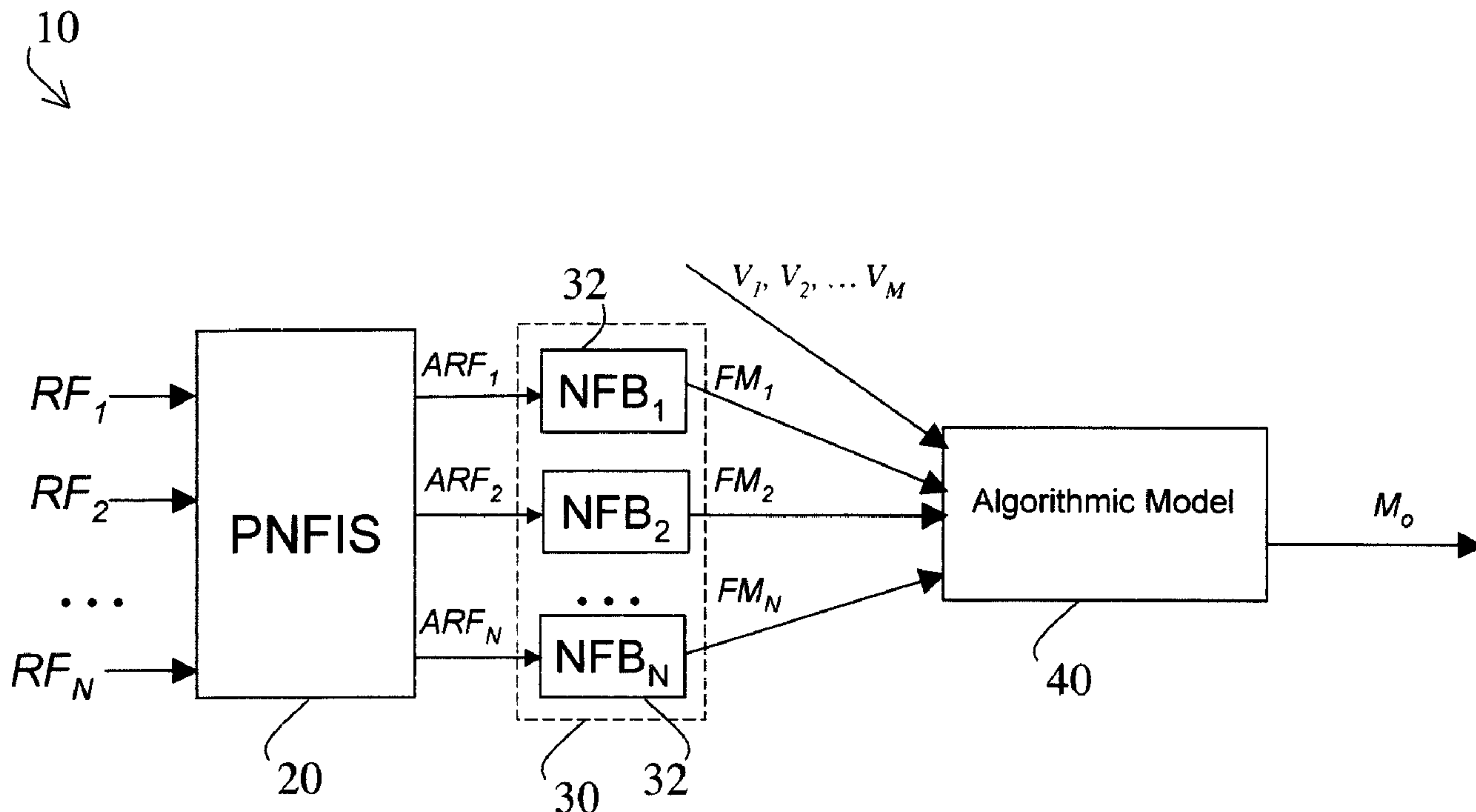
(51) Cl.Int./Int.Cl. *G06F 17/00* (2006.01),
G06N 7/00 (2006.01), *G06F 17/10* (2006.01)

(71) Demandeurs/Applicants:
HUANG, XISHI, CA;
HO, DANNY SIU KAU, CA;
REN, JING, CA;
CAPRETZ, LUIZ F., CA

(72) Inventeurs/Inventors:
HUANG, XISHI, CA;
HO, DANNY SIU KAU, CA;
REN, JING, CA;
CAPRETZ, LUIZ F., CA

(74) Agent: BERESKIN & PARR

(54) Titre : SYSTEME ET METHODE D'ESTIMATION LOGICIELLE
(54) Title: SYSTEM AND METHOD FOR SOFTWARE ESTIMATION



(57) Abrégé/Abstract:

A system and method for software estimation. In one embodiment, the software estimation system comprises a pre-processing neuro-fuzzy inference system used to resolve the effect of dependencies among contributing factors to produce adjusted rating values for the contributing factors, a neuro-fuzzy bank used to calibrate the contributing factors by mapping the adjusted rating values for the contributing factors to generate corresponding numerical parameter values, and a module that applies an algorithmic model (e.g. COCOMO) to produce one or more software output metrics.

ABSTRACT OF THE DISCLOSURE

A system and method for software estimation. In one embodiment, the software estimation system comprises a pre-processing neuro-fuzzy inference system used to resolve the effect of dependencies among contributing factors to produce adjusted rating values for the contributing factors, a neuro-fuzzy bank used to
5 calibrate the contributing factors by mapping the adjusted rating values for the contributing factors to generate corresponding numerical parameter values, and a module that applies an algorithmic model (e.g. COCOMO) to produce one or more software output metrics.

Title: System and Method for Software Estimation**Field of the Invention**

[0001] The invention relates generally to software estimation techniques. More specifically, the invention is directed to a novel and inventive system and method for computing output metrics indicative of the cost, quality, size, or other characteristic of a software development project.

Background of the Invention

[0002] Software estimation, which can include cost estimation, quality estimation, size estimation, or investment risk estimation, for example, is a major issue in software project management faced by many organizations. In this regard, there is a need for software estimation models that will facilitate effective monitoring, control, and assessment of software development projects.

[0003] However, achieving accurate software estimation is inherently a daunting task. The software estimation problem is highly complex, particularly since the relationships between software output metrics and contributing factors generally exhibit strong, complex, non-linear characteristics. Accurate software estimation also typically requires the consideration of many factors, some of which can be difficult to quantify. Prior art approaches used to solve this problem have not been widely successful in effectively and consistently predicting software output metrics.

[0004] One example of a well-known software cost estimation model is the Constructive Cost Model (COCOMO), which integrates expert knowledge. This model is considered to be generally simple, in that it does not require the use of complex mathematics in the estimation process. However, it is also one example of several known models that rely heavily on the availability of sufficient historical project data to be effectively employed, which is not always readily available.

[0005] More recently, techniques based on artificial neural networks have been applied to solve the software estimation problem. However, such techniques have not been widely accepted by software engineering practitioners. While artificial neural networks (ANNs) have the ability to model complex, non-linear relationships, and are capable of approximating measurable functions through learning, ANNs generally operate as "black boxes". Accordingly, known ANN-based models do not provide an explicit explanation of how results are obtained. This lack of transparency may be one of the primary reasons that such techniques have not gained wide acceptance among software engineering practitioners.

[0006] Fuzzy logic techniques have been applied to software estimation problems to a limited extent. Fuzzy logic can be a powerful technique used to solve real world applications with imprecise and uncertain information, and in dealing with semantic knowledge. It is also generally easily understood and interpreted. However, fuzzy-logic based models traditionally do not have learning ability, and the quality of the results obtained when applied to software estimation problems have not, in general, compared favorably to results obtained from applications of more conventional models, such as COCOMO.

20 **Summary of the Invention**

[0007] Embodiments of the invention relate generally to a novel and inventive software estimation model and framework that address at least some of the disadvantages of known techniques. In particular, at least some embodiments of the invention are directed to a system and method for software estimation that makes improved use of both numerical project data and available expert knowledge, by uniquely combining certain aspects of relatively newer software estimation techniques (e.g., neural networks and fuzzy logic) with certain aspects of more conventional software estimation models (e.g. COCOMO), to produce more accurate estimation results.

[0008] Furthermore, the software estimation model provides a good degree of interpretability. For example, in one embodiment of the invention, fuzzy rules are used, in order to better simulate a software engineering practitioner's line of thought when performing software estimation.

5 **[0009]** In one broad aspect of the invention, there is provided a software estimation system for use in software engineering, comprising: at least one neuro-fuzzy component, wherein each neuro-fuzzy component implements a plurality of fuzzy rules, and wherein the at least one neuro-fuzzy component takes as input a plurality of contributing factor ratings and processes the
10 contributing factor ratings in accordance with the fuzzy rules to compute numerical parameters for an algorithmic model; and an algorithmic model module coupled to the at least one neuro-fuzzy component, wherein the module takes as input the numerical parameters and processes the numerical parameters in accordance with the algorithmic model to compute one or more software output
15 metrics; wherein each output metric provides an estimate of a characteristic associated with a software development project.

[0010] In another broad aspect of the invention, the at least one neuro-fuzzy component of the software estimation system comprises: a neuro-fuzzy inference system for resolving the effect of dependencies among a plurality of
20 contributing factors associated with the plurality of contributing factor ratings, wherein the neuro-fuzzy inference system implements a first subset of the plurality of fuzzy rules, and wherein the neuro-fuzzy inference system takes as input the plurality of contributing factor ratings and processes the contributing factor ratings in accordance with the first subset to compute a plurality of
25 adjusted ratings; and a neuro-fuzzy bank coupled to the neuro-fuzzy inference system, wherein the neuro-fuzzy bank implements a second subset of the plurality of fuzzy rules, and wherein the neuro-fuzzy bank takes as input the plurality of adjusted ratings and processes the adjusted ratings in accordance with the second subset to compute the numerical parameters.

Brief Description of the Drawings

[0011] For a better understanding of various embodiments described herein by way of example, reference will be made to the accompanying drawings in which:

5 FIG. 1 is a schematic diagram illustrating components in a software estimation system in an embodiment of the invention;

FIG. 2 is a schematic diagram that illustrates the logical structure of the Pre-processing Neuro-Fuzzy Inference System (PNFIS) of FIG. 1;

10 FIG. 3 is a schematic diagram that illustrates the logical structure of the Neuro-Fuzzy Bank (NFB) of FIG. 1;

FIG. 4 is a flowchart illustrating the steps in a software estimation method in an embodiment of the invention;

FIG. 5 is a flowchart illustrating the steps in a method of computing adjusted ratings of contributing factors performed by the PNFIS of FIG. 1; and

15 FIG. 6 is a flowchart illustrating the steps in a method of computing numerical parameter values for use in an algorithmic model performed by elements of the NFB of FIG. 1.

Detailed Description of the Invention

20 [0012] Embodiments of the invention relate generally to a novel and inventive software estimation model and framework, applicable to various applications such as cost estimation, quality estimation, risk analysis, size estimation, effort estimation, and other estimation problems. Validated results have shown that this framework can greatly improve estimation accuracy. In
25 addition, embodiments of the invention generally provide learning ability, integration capability of expert knowledge and project data, good interpretability, and robustness to imprecise and uncertain inputs.

[0013] In accordance with one broad aspect of the invention, there is provided a system and method for software estimation that combines a neuro-fuzzy technique with at least one algorithmic model. The general architecture of the framework is inherently independent of the choice of algorithmic model and the nature of the estimation problem being considered, and can be applied to a wide variety of estimation problems.

[0014] In one embodiment of the invention, the software estimation system comprises a pre-processing neuro-fuzzy inference system used to resolve the effect of dependencies among contributing factors to produce adjusted rating values for the contributing factors, a neuro-fuzzy bank used to calibrate the contributing factors by mapping the adjusted rating values for the contributing factors to generate corresponding numerical parameter values, and a module that takes the numerical parameter values as input and applies an algorithmic model to produce one or more software output metrics.

[0015] A software output metric produced by the software estimation system will be indicative of the cost, quality, size, or other characteristic of a software development project, depending on the particular algorithmic model employed. For instance, in one example implementation, COCOMO, a model for cost estimation, may be used as the algorithmic model. Software output metrics computed in accordance with embodiments of the invention may be useful for more effectively managing software development projects and significantly reducing associated investment risk, as embodiments of the invention serve to more effectively analyze the feasibility of such projects in comparison to known methods.

[0016] Referring now to FIG. 1, a schematic diagram illustrating components in a software estimation system in an embodiment of the invention is shown generally as 10. In this embodiment of the invention, software estimation system 10 is a computer-implemented system that comprises three primary data

processing components: a pre-processing neuro-fuzzy inference system 20 (“PNFIS”), a neuro-fuzzy bank 30 (“NFB”), and an algorithmic model module 40.

[0017] As will be described in greater detail herein, in FIG. 1, N is the number of contributing factors, M is the number of other variables (e.g. size) in the algorithmic model, RF is a contributing factor rating, ARF is an adjusted contributing factor rating, NFB is a neuro-fuzzy bank, FM is a numerical parameter value generated by the neuro-fuzzy bank for input to the algorithmic model, V is an input to the algorithmic model, and M_o is at least one output metric.

10 [0018] A contributing factor may be broadly defined as any factor that contributes to software output metrics. Contributing factors may include product, computer, project, and personnel attributes, for example. The specific contributing factors employed in generating a given software output metric will generally depend on the specific algorithmic model used. To evaluate the contribution to the software development project of a particular contributing factor, a rating value can be associated with that contributing factor. In this regard, the rating value can be a numerical value defined on a continuous scale, or a qualitative rating level expressed in linguistic terms. Not all rating levels need be valid for each contributing factor.

20 [0019] In many model-based software estimation approaches, it is assumed that the effects of contributing factors on an estimated software output metric are independent, as between contributing factors. However, this assumption does not hold true for all models in all situations. Accordingly, PNFIS 20 not only encodes expert knowledge into fuzzy if-then rules, but also resolves the effect of dependencies between contributing factors in producing adjusted rating values for the contributing factors. More specifically, PNFIS 20 takes a set of contributing factor ratings RF_i as input, and produces a set of corresponding adjusted ratings for the contributing factors ARF_i as output, which take into account the effect of interdependencies between contributing factors.

- 7 -

[0020] The logical structure of PNFIS 20 in one embodiment of the invention is depicted schematically in FIG. 2. In one implementation of PNFIS 20, fuzzy rules take the form of:

Fuzzy Rule (i,k): IF (RF_1 is A_{1jik}) AND (RF_2 is A_{2jik}) AND ... AND (RF_N is A_{Njik})

5 THEN $ARF_i = PFP_{ik} \cdot RF_i, i = 1, 2, \dots, N, k = 1, 2, \dots, M_i$

where M_i is the number of fuzzy rules with contributing factor i as the consequent, PFP_{ik} is an adjustable parameter associated with the fuzzy rule (i,k) , and A_{sjik} is a fuzzy set associated with the j_{ik} -th rating level of factor s for fuzzy rule (i,k) .

[0021] The form of the fuzzy rules used may differ in variant
10 implementations of PNFIS 20. For example, the fuzzy rules may change depending on the estimation problem being considered, and/or on the specific algorithmic model applied in algorithmic model module 40.

[0022] Further details with respect to the processing performed by PNFIS 20 will be discussed with reference to FIG. 5.

15 **[0023]** NFB 30 is used to calibrate each contributing factor i by mapping the adjusted rating values ARF_i for the respective contributing factor produced by PNFIS 20 into corresponding numerical parameter values FM_i , to be used as input for the algorithmic model embodied in algorithmic model module 40. In NFB 30, the i -th element is a neuro-fuzzy subsystem 32 (NFB_i), which is
20 associated with contributing factor i . In one embodiment of the invention, each contributing factor is associated with one of several qualitative rating levels, which may be expressed in linguistic terms. For example, the COCOMO II model uses six such rating levels: Very Low (VL), Low (L), Nominal (N), High (H), Very High (VH) and Extra High (XH).

25 **[0024]** In a software estimation system, each contributing factor and the corresponding rating criteria for each rating level are typically defined. For instance, where "PCAP" denotes the contributing factor of programmers'

capability, if rating level "H" has been associated with "PCAP", then the capability of the programmers is considered "high". However, for the algorithmic model used to estimate software output metrics in this embodiment of the invention, a numerical value corresponding to the rating value of each contributing factor, which can be used in a mathematical formula associated with the algorithmic model, needs to be determined. Put another way, for every contributing factor, each rating level needs to relate to a quantitative numerical value, namely a numerical parameter value, for use in the algorithmic model. Accordingly, mappings from adjusted rating values for the contributing factors to numerical parameter values are made by NFB 30.

[0025] The logical structure of each NFB_i 32 of NFB 30 in one embodiment of the invention is depicted schematically in FIG. 3. In each NFB_i 32, fuzzy rules take the form of:

Fuzzy Rule (i,k): IF (ARF_i is A_{ik})

THEN $FM_i = FMP_{ik}$, $i = 1, 2, \dots, N$, $k = 1, 2, \dots, N_i$

where N_i is the number of rating levels for contributing factor i , A_{ik} is a fuzzy set associated with the k -th rating level of contributing factor i , and FMP_{ik} is an adjustable parameter associated with the k -th rating level of contributing factor i .

[0026] The number of elements NFB_i 32 in NFB 30 is equal to the number of contributing factors, and the number of fuzzy rules equals the number of rating levels associated with the corresponding contributing factor. For example, if contributing factor i has six rating levels, each NFB_i 32 is composed of the following six fuzzy if-then rules:

Fuzzy Rule (i,1): IF ARF_i is A_{i1} (Very Low), THEN $FM_i = FMP_{i1}$

Fuzzy Rule (i,2): IF ARF_i is A_{i2} (Low), THEN $FM_i = FMP_{i2}$

Fuzzy Rule (i,3): IF ARF_i is A_{i3} (Nominal), THEN $FM_i = FMP_{i3}$

Fuzzy Rule (i,4): IF ARF_i is A_{i4} (High), THEN $FM_i = FMP_{i4}$

Fuzzy Rule (i,5): IF ARF_i is A_{i5} (Very High), THEN $FM_i = FMP_{i5}$

Fuzzy Rule (i,6): IF ARF_i is A_{i6} (Extra High), THEN $FM_i = FMP_{i6}$

- [0027] For any particular software estimation problem, once the number of contributing factors are determined and the number of rating levels for each contributing factor is determined, the structure and fuzzy rules of NFB 30 are determined. Only the fuzzy rule parameters (e.g. PFP_{ik} , $k=1,2,\dots,M_i$ of FIG. 2, FMP_{ik} , $k=1,2,\dots,N_i$ of FIG. 3) need to be fine-tuned by the learning process from numerical project data. Different learning processes and algorithms may be employed for this purpose in variant implementations.
- 10 [0028] In a variant embodiment of the invention, the fuzzy rules for each NFB_i 32 are subject to a monotonic constraint on the corresponding contributing factor i to ensure that the calibrated results generated by NFB 30 are reasonable, and not counter-intuitive. Monotonic constraints reflect the expert knowledge about the effects of contributing factors on an estimated software output metric.
- 15 For most contributing factors, when the rating value of a contributing factor goes high, the estimated software output metric should change monotonically; in other words, it should increase or decrease along only one direction. For example, in one implementation of this embodiment, the following monotonic constraints may be applied:

$$20 \quad FMP_{i1} \leq FMP_{i2} \leq \dots \leq FMP_{iN_i}, i \in I_{INC}(F)$$

$$FMP_{i1} \geq FMP_{i2} \geq \dots \geq FMP_{iN_i}, i \in I_{DEC}(F)$$

- where $I_{INC}(F)$ is the index set of increasing contributing factors whose higher rating value corresponds to the higher values of the estimated software output metric, and $I_{DEC}(F)$ is the index set of decreasing contributing factors whose higher rating value corresponds to the lower value of the estimated software output metric. Monotonic constraints may be formulated for other variables and parameters in variant implementations.
- 25

[0029] The functionality of *NFB_i 32* (and similarly, PNFIS 20) can be looked at from two perspectives. From the learning perspective, we can treat it as a neural network, so that we can use available learning algorithms for neural networks to calibrate the corresponding parameters. Therefore, *NFB_i 32* has learning capability. From the reasoning perspective, *NFB_i 32* can be considered as a fuzzy logic system. Its output is derived using fuzzy if-then rules, and the reasoning process is transparent and is similar to the decision-making process of human beings. Therefore, *NFB_i 32* is not a black box. The entire reasoning process is clear to users of the software estimation system, and can be traced and validated by users and experts, thereby making the framework more easily accepted for application in project management.

[0030] Further details with respect to the processing performed by each *NFB_i 32* will be discussed with reference to FIG. 6 below.

[0031] Algorithmic model module 40 performs further processing in software estimation system 10, by computing software output metric(s) M_o from numerical parameter values FM_i generated by *NFB 30* and other variables V_i , in accordance with a selected algorithmic model. Software estimation system 10 is flexible, in that different algorithmic models can be selected for use in software estimation system 10 in variant implementations of the invention. Depending on the application, the algorithmic model may take a different form.

[0032] An algorithmic model can be built by analyzing software output metrics and attributes of completed projects, and used to predict software output metrics based on the attributes of the software development product and development process under consideration. Many algorithmic models have been proposed to estimate different software output metrics, such as software development cost, software maintenance cost, software quality, software development productivity, software size, scheduling, staffing, and defect prediction, for example. Algorithmic model module 40 may be adapted to use any of these known algorithmic models to compute associated software output

metric(s). Specific known algorithmic models may include models employed in a Quantitative Software Management (QSM) Software Life cycle Management model (SLIM) tool, models employed in a Software Productivity Research (SPR) KnowledgePLAN[®] tool, models employed in Computer Associate's CA-Estimacs package, SEER estimating models developed by GA SEER Technologies, and models used in CostXpert tools developed by Cost Xpert Group, Inc., for example.

[0033] Another example of an algorithmic model that may be used is the COCOMO II post architecture model, used to predict software development effort, which takes as input the software size and ratings of 22 cost drivers as contributing factors, including five scale factors and 17 effort multipliers. This model takes the form of:

$$Effort = A \times (Size)^{B+0.01 \times \sum_{i=1}^5 SF_i} \times \prod_{i=1}^{17} EM_i$$

where A and B are constants, $Size$ refers to the size of software product, scale factors SF_i and effort multipliers EM_i are software product, platform, personnel, and project attributes. In one implementation of an embodiment of the invention, the value of $Effort$ computed upon applying this algorithmic model would be provided as a software output metric (e.g. as an output metric M_o of software estimation system 10 of FIG. 1).

[0034] Referring now to FIG. 4, a flowchart illustrating the steps in a software estimation method in an embodiment of the invention is shown generally as 100. In this embodiment, method 100 is computer-implemented, wherein the steps of method 100 are implemented in software instructions of a computing application for execution by computer.

[0035] At step 110, adjusted ratings of contributing factors are computed by a pre-processing neuro-fuzzy inference system (e.g. PNFIS 20 of FIG. 1). At step 112, the adjusted ratings computed at step 110 are used by a neuro-fuzzy

bank (e.g. NFB 30 of FIG. 1) to compute numerical parameter values for the algorithmic model to be employed (e.g. as implemented by algorithmic model module 40 of FIG. 1). At step 114, an algorithmic model (e.g. COCOMO) is applied in computing at least one output metric from the numerical parameter values computed at step 112. Steps 110 and 112 of method 100 will now be described in further detail with reference to FIGS. 5 and 6.

[0036] Referring now to FIG. 5, a flowchart illustrating the steps in a method of computing adjusted ratings of contributing factors performed by the PNFIS of FIG. 1 is shown. This method corresponds to step 110 of FIG. 4.

10 **[0037]** The structure of the PNFIS used in this embodiment of the invention was depicted schematically in FIG. 2. The function of the PNFIS can be described by layer as follows.

[0038] First, at step 120, contributing factor ratings RF_i , $i = 1, 2, \dots, N$, where N is the total number of contributing factors, are received as input to be processed.

[0039] With respect to the first layer, at step 122 (layer 1), the membership value for rating level j of contributing factor i is calculated. The activation function of a node in this layer is defined as the corresponding membership function:

$$O_{ij}^1 = \mu_{ij}(RF_i), i = 1, 2, \dots, N, j = 1, 2, \dots, N_i$$

20 where N_i is the number of rating levels for contributing factor i , RF_i is the rating of contributing factor i , and $\mu_{ij}(RF_i)$ is the membership function of a fuzzy set A_{ij} associated with the j -th rating level of factor i .

[0040] In a variant embodiment of the invention, the membership function may be a triangular function. Other membership functions may be employed in variant embodiments.

25 **[0041]** With respect to the second layer, at step 124 (layer 2), the firing strength w_{ik} for fuzzy rule (i, k) is calculated. The inputs are the membership

- 13 -

values in the premise of the fuzzy rule. The output is the product of all input membership values, which is called the firing strength of the corresponding fuzzy rule. The firing strength can be calculated as follows:

$$w_{ik} = \prod_{s=1}^N O_{sj_{ik}}^1, i=1,2,\dots,N, k=1,2,\dots,M_i$$

5 where M_i is the number of fuzzy rules with contributing factor i as the consequent.

[0042] Moreover, at step 126, the sum of firing strength S_{iw} associated with contributing factor i as the consequent in all fuzzy rules within the PNFIS is then calculated:

$$S_{iw} = \sum_{k=1}^{M_i} w_{ik}, i=1,2,\dots,N$$

10 [0043] With respect to the third layer, at step 128, the firing strength for fuzzy rule (i,k) is normalized. The output of the k -th node is called the normalized firing strength, which is defined as follows:

$$\bar{w}_{ik} = \frac{w_{ik}}{S_{iw}}, i=1,2,\dots,N, k=1,2,\dots,M_i$$

[0044] With respect to the fourth layer, at step 130, the reasoning result for
15 fuzzy rule (i,k) is calculated as follows:

$$O_{ik}^4 = \bar{w}_{ik} \cdot PFP_{ik} \cdot RF_i, i=1,2,\dots,N, k=1,2,\dots,M_i$$

where PFP_{ik} is an adjustable parameter associated with fuzzy rule (i,k) . This adjustable parameter may be obtained initially from an algorithmic model or based on expert knowledge. It can subsequently be changed by learning.

20 [0045] With respect to the fifth layer, at step 132, all reasoning results calculated at step 130 are summed as follows to obtain adjusted rating values for contributing factor i :

- 14 -

$$ARF_i = \sum_{k=1}^{M_i} O_{ik}^4, \quad i = 1, 2, \dots, N$$

[0046] In summary, the overall output of the PNFIS, namely the i -th adjusted contributing factor rating can be defined as follows:

$$ARF_i = \sum_{k=1}^{M_i} \left(\frac{\prod_{s=1}^N \mu_{sjk}(RF_s)}{\sum_{j=1}^{M_i} \left(\prod_{s=1}^N \mu_{sjj}(RF_s) \right)} \cdot PFP_{ik} \cdot RF_i \right), \quad i = 1, 2, \dots, N$$

5 **[0047]** Referring to FIG. 6, a flowchart illustrating the steps in a method of computing numerical parameter values for use in an algorithmic model performed by elements of the NFB of FIG. 1 is shown. This method corresponds to step 112 of FIG. 4.

10 **[0048]** The structure of each subsystem of the NFB used in this embodiment of the invention is depicted schematically in FIG. 3. The function of the NFB_i can be described by layer as follows.

[0049] First, at step 140, adjusted contributing factor ratings ARF_i , $i=1,2,\dots,N$, where N is the total number of contributing factors, are received as input to be processed.

15 **[0050]** With respect to the first layer, at step 142 (layer 1), the membership value for each fuzzy rule (i,k) is calculated. The activation function of a node in this layer is defined as the corresponding membership function:

$$O_{ik}^1 = \mu_{ik}(ARF_i), \quad k=1,2,\dots,N_i$$

20 where N_i is the number of rating levels for contributing factor i , ARF_i is the adjusted rating value of contributing factor i , and $\mu_{ik}(ARF_i)$ is the membership function of a fuzzy set A_{ik} associated with the k -th rating level of contributing factor i .

[0051] In this embodiment, the membership function is the same as that described with reference to FIG. 5. In a variant embodiment of the invention, the membership function may be a triangular function. Other membership functions may be employed in variant embodiments.

5 **[0052]** With respect to the second layer, at step 144 (layer 2), the firing strength w_{ik} for each fuzzy rule (i,k) is calculated. The inputs are the membership values in the premise of the fuzzy rule. The output is the product of all input membership values, which is called the firing strength of the corresponding fuzzy rule. In this case, because there is only one condition in the premise of each
10 fuzzy rule, the firing strength is the same as the membership value obtained in layer 1, namely:

$$w_{ik} = O_{ik}^1$$

[0053] Moreover, at step 146, the sum of firing strength S_{iw} associated with contributing factor i as the consequent in the N_i fuzzy rules for contributing factor
15 i is then calculated:

$$S_{iw} = \sum_{j=1}^{N_i} w_{ij}$$

[0054] With respect to the third layer, at step 148, the firing strength for each fuzzy rule is normalized. The output of the k -th node is called the normalized firing strength, which is defined as follows:

20
$$\bar{w}_{ik} = \frac{w_{ik}}{S_{iw}}, \quad k = 1, 2, \dots, N_i$$

[0055] With respect to the fourth layer, at step 150, the reasoning result of a fuzzy rule is calculated as follows:

$$O_{ik}^4 = \bar{w}_{ik} FMP_{ik}$$

- 16 -

where FMP_{ik} is an adjustable parameter associated with the k -th rating level of contributing factor i . This adjustable parameter may be obtained initially from an algorithmic model or based on expert knowledge. It can subsequently be changed by learning.

- 5 [0056] With respect to the fifth layer, at step 152, all reasoning results calculated at step 150 are summed as follows, to obtain numerical parameter values to be further processed by the algorithmic model:

$$FM_i = \sum_{k=1}^{N_i} O_{ik}^4$$

- 10 [0057] In summary, the overall output of the i -th element NFB_i in the NFB is:

$$FM_i = \sum_{k=1}^{N_i} \bar{w}_{ik} FMP_{ik} = \sum_{k=1}^{N_i} \frac{\mu_{ik}(ARF_i)}{\sum_{j=1}^{N_i} \mu_{ij}(ARF_i)} \cdot FMP_{ik}, \quad i=1,2,\dots,N$$

- 15 [0058] Industrial project data was used to validate an implementation of an embodiment of the invention. For example, one case study utilized data from 69 projects. The results of the study as summarized in Table 1 below illustrate that the software estimation system in this embodiment of the invention facilitated a significant improvement in cost estimation accuracy as compared to a standard COCOMO model.

Table 1. Cost Estimation for 69 Project Data Points

ARE	COCOMO Model	An implementation of an embodiment of the Software Estimation System					
		Case I		Case II		Case III	
	PERC	PERC	IMPRV	PERC	IMPRV	PERC	IMPRV
20%	71%	86%	15%	88%	17%	88%	17%
30%	81%	92%	11%	92%	11%	92%	11%

- 20 ARE – absolute relative error
 PERC – percent of projects
 IMPRV – improvement provided by the implementation of an embodiment of the software estimation system over COCOMO

- 17 -

Case I. Learning with all project data
Case II. Learning with part of project data
Case III. Use larger weights for local project data

5 **[0059]** It will be understood by persons skilled in the art that the software estimation systems and methods described herein may be applied to a variety of software estimation applications, as well as other types of estimation more generally (e.g. predicting stock performance, predicting the medical condition of patients). Applications of the framework may also be extended to applications in
10 a wide variety of industries, including aerospace applications, communication systems applications, consumer appliance applications, electric power systems applications, manufacturing automation and robotics applications, power electronics and motion control applications, industrial process engineering applications, and transportation applications, for example.

15 **[0060]** Furthermore, in variant implementations, an embodiment of the invention may be employed as an element in a larger framework, working in conjunction with other systems, models, or tools to produce output metrics.

[0061] It will be understood by persons skilled in the art that the specific configuration of the PNFIS and NFB described in the foregoing description is
20 provided by way of example only, and that other configurations may be employed without departing from the scope of the invention. For example, in variant embodiments of the invention, each of the PNFIS and NFB may have a fewer number or a greater number of layers than that shown in the Figures. Additional layers may be added and certain layers may be deleted, for example. As a
25 further example, in variant embodiments of the invention, the functions of the PNFIS and NFB may be merged in a particular module or component.

[0062] The steps of the methods described herein with respect to one or more embodiments of the invention may be provided as executable software instructions stored on computer-readable media, which may include
30 transmission-type media.

- 18 -

[0063] The invention has been described with regard to a number of embodiments. However, it will be understood by persons skilled in the art that other variants and modifications may be made without departing from the scope of the invention as defined in the claims appended hereto.

5

Claims:

1. A software estimation system for use in software engineering, comprising:
 - a) at least one neuro-fuzzy component, wherein each neuro-fuzzy component implements a plurality of fuzzy rules, and wherein said at least one
5 neuro-fuzzy component takes as input a plurality of contributing factor ratings and processes said contributing factor ratings in accordance with said fuzzy rules to compute numerical parameter values for an algorithmic model; and
 - b) an algorithmic model module coupled to said at least one neuro-fuzzy component, wherein said module takes as input said numerical parameters
10 and processes said numerical parameter values in accordance with said algorithmic model to compute at least one output metric; wherein each output metric provides an estimate of a characteristic associated with a software development project.

2. The software estimation system of claim 1, wherein said at least one
15 neuro-fuzzy component comprises:
 - i) a neuro-fuzzy inference system for resolving the effect of dependencies among a plurality of contributing factors associated with said plurality of contributing factor ratings, wherein said neuro-fuzzy inference system implements a first subset of said plurality of fuzzy rules, and wherein said neuro-
20 fuzzy inference system takes as input said plurality of contributing factor ratings and processes said contributing factor ratings in accordance with said first subset to compute a plurality of adjusted ratings; and
 - ii) a neuro-fuzzy bank coupled to said neuro-fuzzy inference system, wherein said neuro-fuzzy bank implements a second subset of said plurality of
25 fuzzy rules, and wherein said neuro-fuzzy bank takes as input said plurality of adjusted ratings and processes said adjusted ratings in accordance with said second subset to compute said numerical parameter values.

- 20 -

3. The software estimation system of claim 2, wherein said first subset of said plurality of fuzzy rules is defined by:

Fuzzy Rule (i,k): IF (RF_1 is A_{1jik}) AND (RF_2 is A_{2jik}) AND ... AND (RF_N is A_{Njik})
THEN $ARF_i = PFP_{ik} \cdot RF_i$, $i = 1, 2, \dots, N$, $k = 1, 2, \dots, M_i$

5 where RF_i is the rating of contributing factor i , ARF_i is the adjusted rating of contributing factor i , M_i is the number of fuzzy rules with contributing factor i as the consequent, PFP_{ik} is an adjustable parameter associated with the fuzzy rule (i,k) , and A_{sjik} is a fuzzy set associated with the jik -th rating level of factor s for fuzzy rule (i,k) .

10 4. The software estimation system of claim 3, wherein said plurality of adjusted ratings computed by said neuro-fuzzy inference system satisfy:

$$ARF_i = \sum_{k=1}^{M_i} \left(\frac{\prod_{s=1}^N \mu_{sjik}(RF_s)}{\sum_{j=1}^{M_i} \left(\prod_{s=1}^N \mu_{sjij}(RF_s) \right)} \cdot PFP_{ik} \cdot RF_i \right), \quad i = 1, 2, \dots, N$$

15 where ARF_i is the adjusted rating of contributing factor i , RF_i is the rating of contributing factor i , PFP_{ik} is an adjustable parameter associated with the fuzzy rule (i,k) , and $\mu_{sjik}(RF_s)$ is the membership function of a fuzzy set A_{sjik} associated with the jik -th rating level of contributing factor s .

5. The software estimation system of claim 4, wherein said membership function is a triangular membership function.

20 6. The software estimation system of claim 2, wherein said second subset of said plurality of fuzzy rules is defined by:

Fuzzy Rule (i,k): IF (ARF_i is A_{ik})
THEN $FM_i = FMP_{ik}$, $i = 1, 2, \dots, N$, $k = 1, 2, \dots, N_i$

- 21 -

where ARF_i is the adjusted rating of contributing factor i , FM_i is the numerical parameter value for contributing factor i , N_i is the number of rating levels for contributing factor i , A_{ik} is a fuzzy set associated with the k -th rating level of contributing factor i , and FMP_{ik} is an adjustable parameter associated with the k -th rating level of contributing factor i .

7. The software estimation system of claim 6, wherein said plurality of numerical parameter values computed by said neuro-fuzzy bank satisfy:

$$FM_i = \sum_{k=1}^{N_i} \frac{\mu_{ik}(ARF_i)}{\sum_{j=1}^{N_i} \mu_{ij}(ARF_i)} \cdot FMP_{ik}, \quad i = 1, 2, \dots, N$$

where FM_i is the numerical parameter value for contributing factor i , ARF_i is the adjusted rating of contributing factor i , FMP_{ik} is a corresponding parameter value associated with the k -th rating level, and $\mu_{ik}(ARF_i)$ is the membership function of a fuzzy set A_{ik} associated with the k -th rating level of contributing factor i .

8. The software estimation system of claim 7, wherein said membership function is a triangular membership function.

9. The software estimation system of claim 6, wherein said numerical parameter values are subject to the following monotonic constraints:

$$FMP_{i1} \leq FMP_{i2} \leq \dots \leq FMP_{iN_i}, \quad i \in I_{INC}(F)$$

$$FMP_{i1} \geq FMP_{i2} \geq \dots \geq FMP_{iN_i}, \quad i \in I_{DEC}(F)$$

where $I_{INC}(F)$ is the index set of increasing contributing factors whose higher rating value corresponds to the higher values of said at least one output metric, and $I_{DEC}(F)$ is the index set of decreasing contributing factors whose higher rating value corresponds to the lower value of said at least one output metric.

- 22 -

10. The software estimation system of claim 1, wherein said algorithmic model is the constructive cost model COCOMO.
11. The software estimation system of claim 10, wherein said at least one output metric is an estimate of development effort associated with said software
5 development project.
12. The software estimation system of claim 1, wherein each of said at least one output metric is an estimate of one of the elements selected from the following group: software development cost, software maintenance cost, software quality, software development productivity, software size, scheduling, staffing,
10 and defect prediction.
13. A software estimation method comprising the steps of:
- a) computing numerical parameter values for an algorithmic model in at least one neuro-fuzzy component, wherein each neuro-fuzzy component implements a plurality of fuzzy rules, and wherein said at least one
15 neuro-fuzzy component takes as input a plurality of contributing factor ratings and processes said contributing factor ratings in accordance with said fuzzy rules; and
 - b) computing at least one output metric in an algorithmic model module, wherein said module takes as input said numerical parameters and
20 processes said numerical parameter values in accordance with said algorithmic model, and wherein each output metric provides an estimate of a characteristic associated with a software development project.
14. The software estimation method of claim 13, wherein said at least one neuro-fuzzy component comprises:
- 25 i) a neuro-fuzzy inference system for resolving the effect of dependencies among a plurality of contributing factors associated with said plurality of contributing factor ratings, wherein said neuro-fuzzy inference system
-

implements a first subset of said plurality of fuzzy rules, and wherein said neuro-fuzzy inference system takes as input said plurality of contributing factor ratings and processes said contributing factor ratings in accordance with said first subset to compute a plurality of adjusted ratings; and

- 5 ii) a neuro-fuzzy bank coupled to said neuro-fuzzy inference system, wherein said neuro-fuzzy bank implements a second subset of said plurality of fuzzy rules, and wherein said neuro-fuzzy bank takes as input said plurality of adjusted ratings and processes said adjusted ratings in accordance with said second subset to compute said numerical parameter values.

- 10 15. The software estimation method of claim 14, wherein said first subset of said plurality of fuzzy rules is defined by:

Fuzzy Rule (i,k): IF (RF_1 is A_{1jik}) AND (RF_2 is A_{2jik}) AND ... AND (RF_N is A_{Njik})
 THEN $ARF_i = PFP_{ik} \cdot RF_i$, $i = 1, 2, \dots, N$, $k = 1, 2, \dots, M_i$

- 15 where RF_i is the rating of contributing factor i , ARF_i is the adjusted rating of contributing factor i , M_i is the number of fuzzy rules with contributing factor i as the consequent, PFP_{ik} is an adjustable parameter associated with the fuzzy rule (i,k), and A_{sjik} is a fuzzy set associated with the jik -th rating level of factor s for fuzzy rule (i,k).

- 20 16. The software estimation method of claim 15, wherein said plurality of adjusted ratings computed by said neuro-fuzzy inference system satisfy:

$$ARF_i = \sum_{k=1}^{M_i} \left(\frac{\prod_{s=1}^N \mu_{sjik}(RF_s)}{\sum_{j=1}^{M_i} \left(\prod_{s=1}^N \mu_{sjj}(RF_s) \right)} \cdot PFP_{ik} \cdot RF_i \right), \quad i = 1, 2, \dots, N$$

where ARF_i is the adjusted rating of contributing factor i , RF_i is the rating of contributing factor i , PFP_{ik} is an adjustable parameter associated with the fuzzy

- 24 -

rule (i,k) , and $\mu_{sjk}(RF_s)$ is the membership function of a fuzzy set A_{sjk} associated with the j_{ik} -th rating level of contributing factor s .

17. The software estimation method of claim 16, wherein said membership function is a triangular membership function.

5 18. The software estimation method of claim 14, wherein said second subset of said plurality of fuzzy rules is defined by:

Fuzzy Rule (i,k) : IF (ARF_i is A_{ik})

THEN $FM_i = FMP_{ik}$, $i = 1, 2, \dots, N$, $k = 1, 2, \dots, N_i$

10 where ARF_i is the adjusted rating of contributing factor i , FM_i is the numerical parameter value for contributing factor i , N_i is the number of rating levels for contributing factor i , A_{ik} is a fuzzy set associated with the k -th rating level of contributing factor i , and FMP_{ik} is an adjustable parameter associated with the k -th rating level of contributing factor i .

15 19. The software estimation method of claim 18, wherein said plurality of numerical parameter values computed by said neuro-fuzzy bank satisfy:

$$FM_i = \frac{\sum_{k=1}^{N_i} \mu_{ik}(ARF_i)}{\sum_{j=1}^{N_i} \mu_{ij}(ARF_i)} \cdot FMP_{ik}, \quad i = 1, 2, \dots, N$$

20 where FM_i is the numerical parameter value for contributing factor i , ARF_i is the adjusted rating of contributing factor i , FMP_{ik} is a corresponding parameter value associated with the k -th rating level, and $\mu_{ik}(ARF_i)$ is the membership function of a fuzzy set A_{ik} associated with the k -th rating level of contributing factor i .

20. The software estimation method of claim 19, wherein said membership function is a triangular membership function.

- 25 -

21. The software estimation method of claim 18, wherein said numerical parameter values are subject to the following monotonic constraints:

$$FMP_{i1} \leq FMP_{i2} \leq \dots \leq FMP_{iN_i}, i \in I_{INC}(F)$$

$$FMP_{i1} \geq FMP_{i2} \geq \dots \geq FMP_{iN_i}, i \in I_{DEC}(F)$$

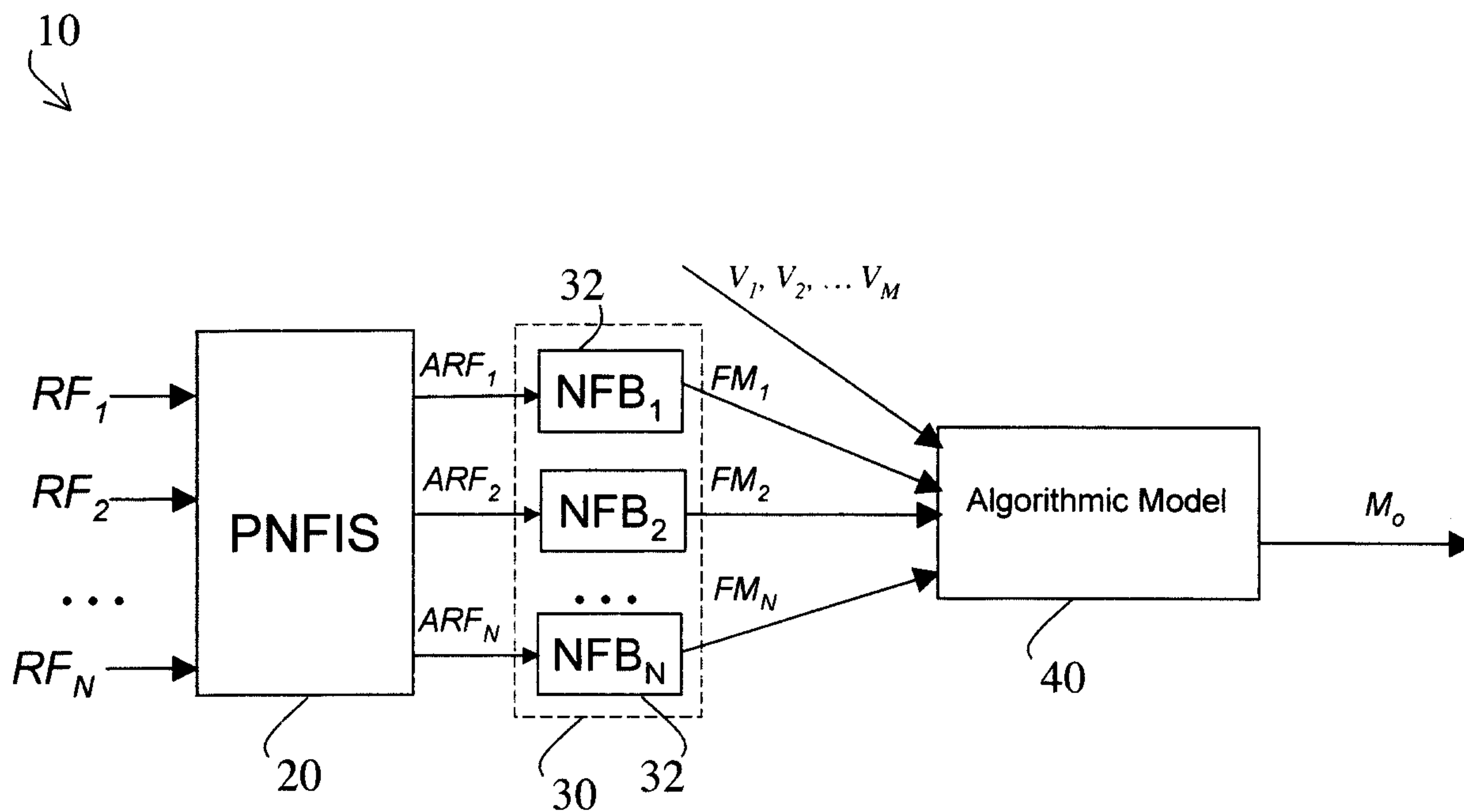
5 where $I_{INC}(F)$ is the index set of increasing contributing factors whose higher rating value corresponds to the higher values of said at least one output metric, and $I_{DEC}(F)$ is the index set of decreasing contributing factors whose higher rating value corresponds to the lower value of said at least one output metric.

22. The software estimation method of claim 13, wherein said algorithmic
10 model is the constructive cost model COCOMO.

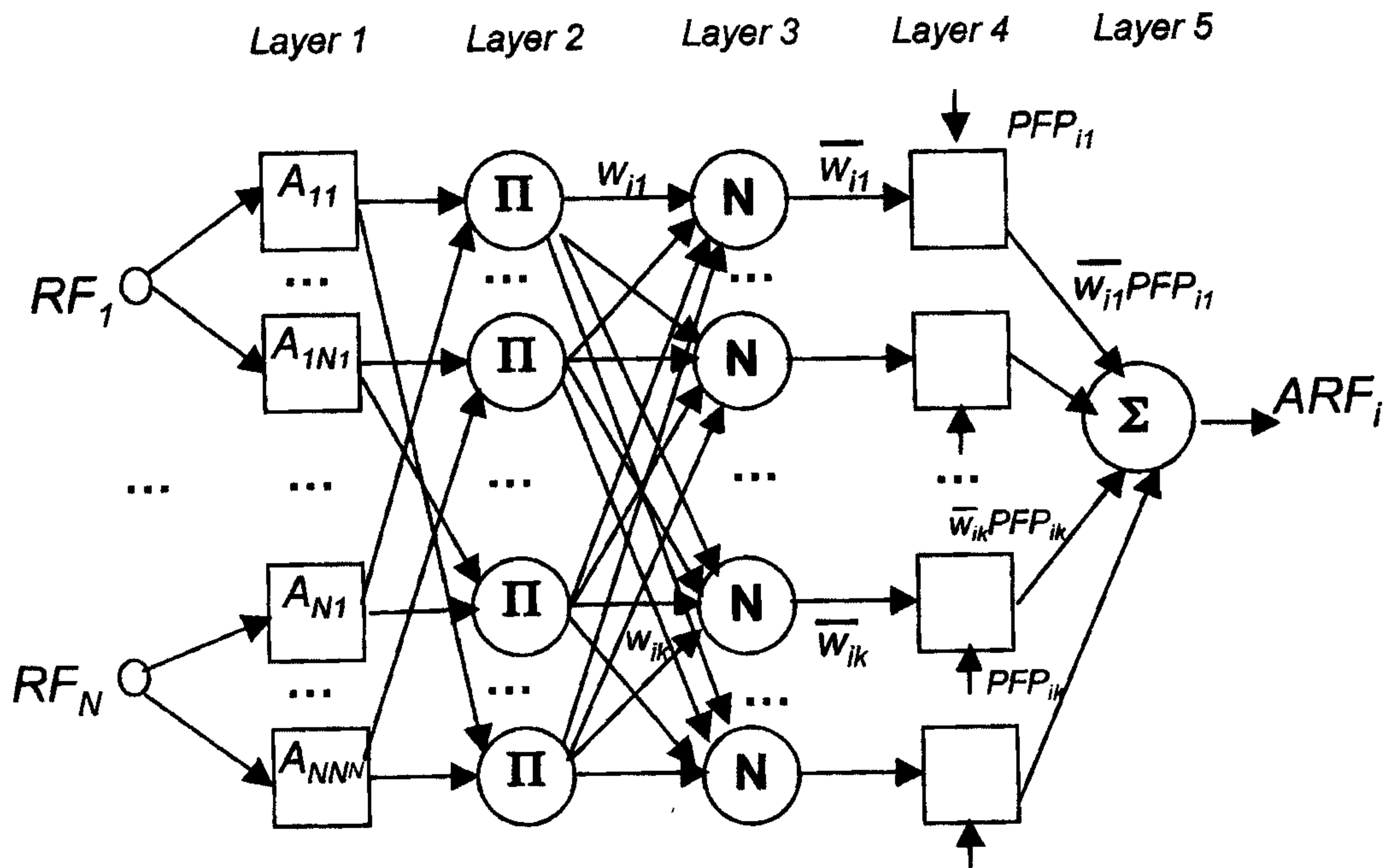
23. The software estimation method of claim 22, wherein said at least one output metric is an estimate of development effort associated with said software development project.

24. The software estimation method of claim 13, wherein each of said at least
15 one output metric is an estimate of one of the elements selected from the following group: software development cost, software maintenance cost, software quality, software development productivity, software size, scheduling, staffing, and defect prediction.

20

**FIG. 1**

20

**FIG. 2**

32

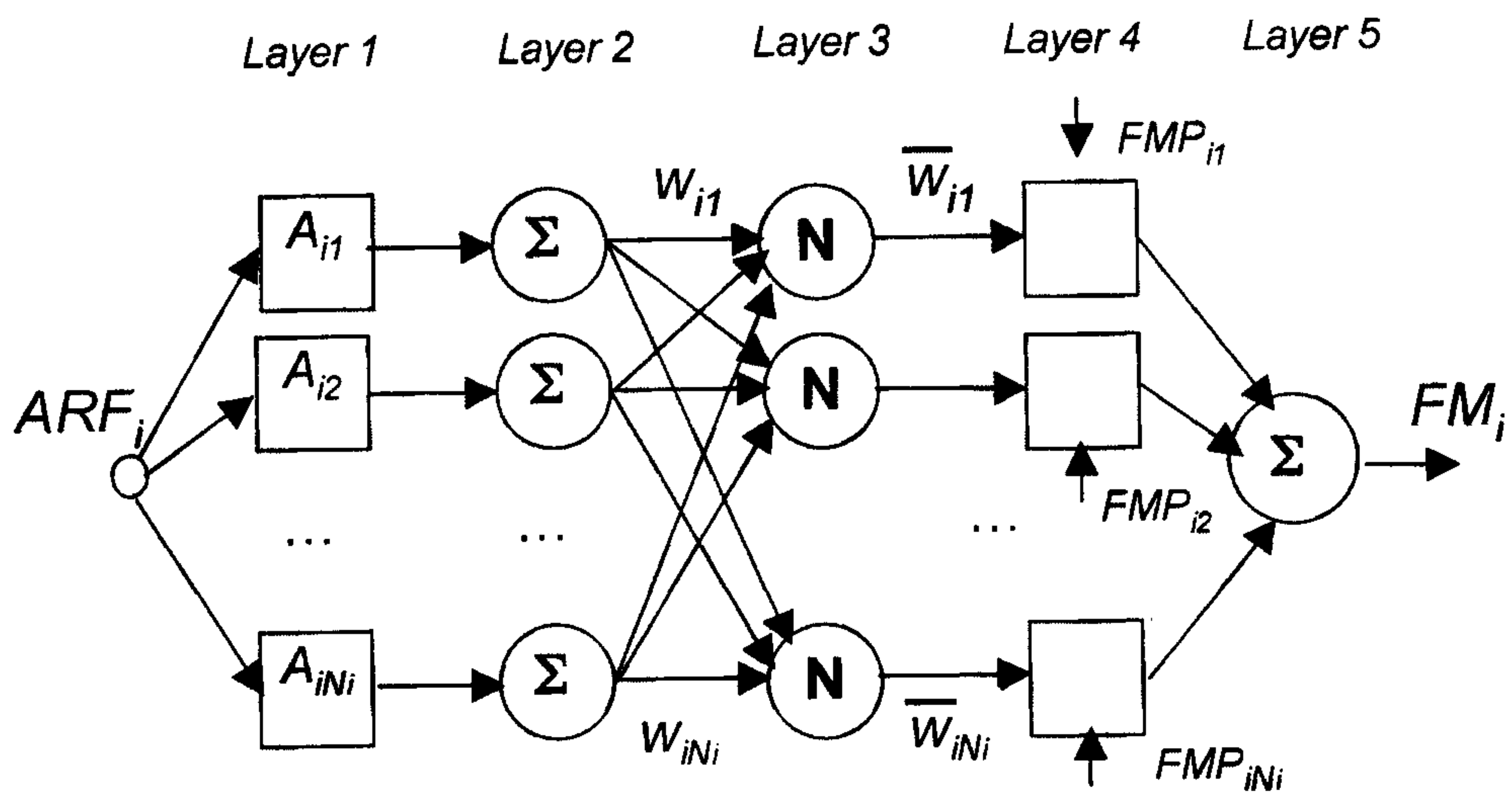


FIG. 3

100
↙

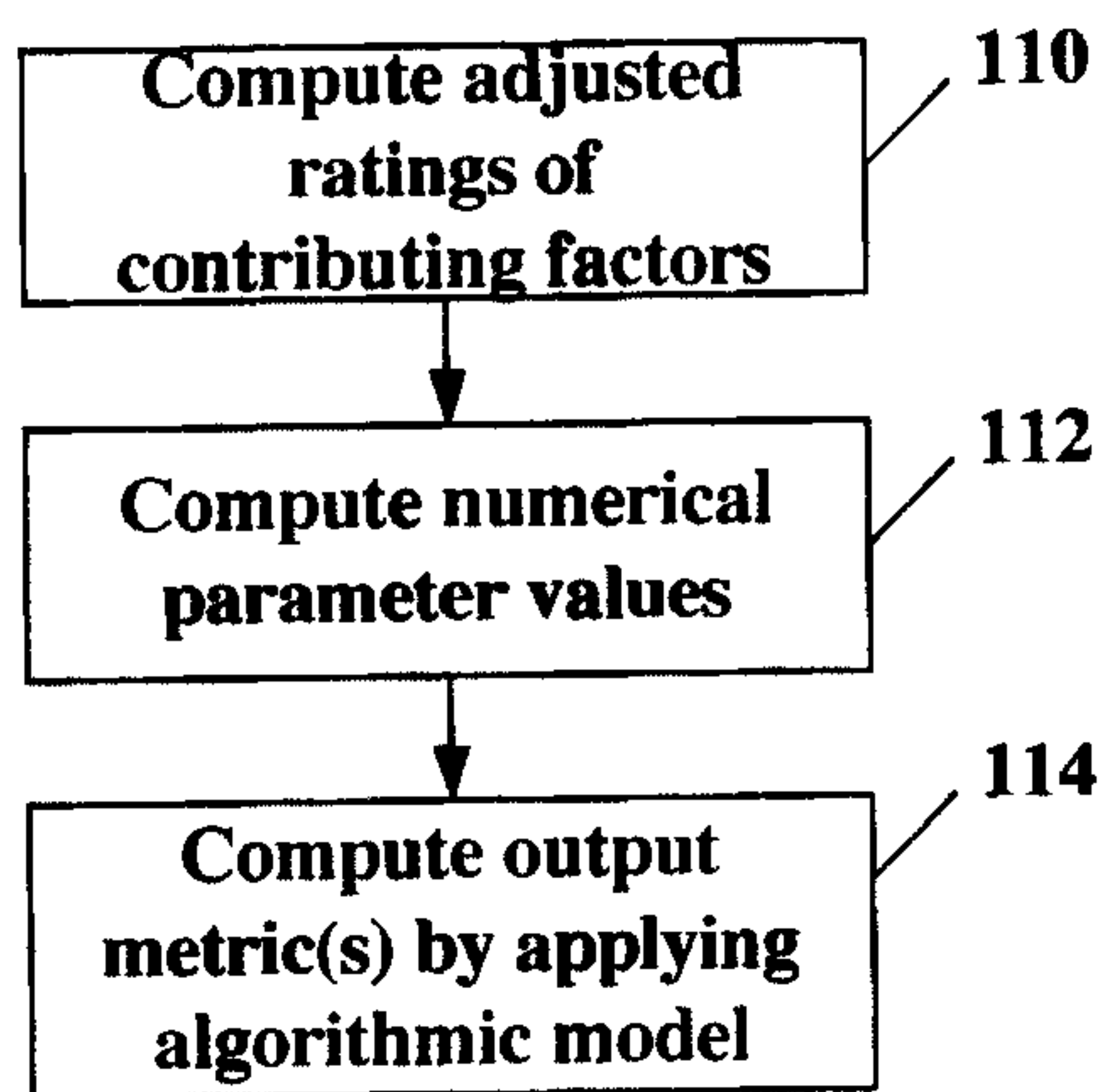
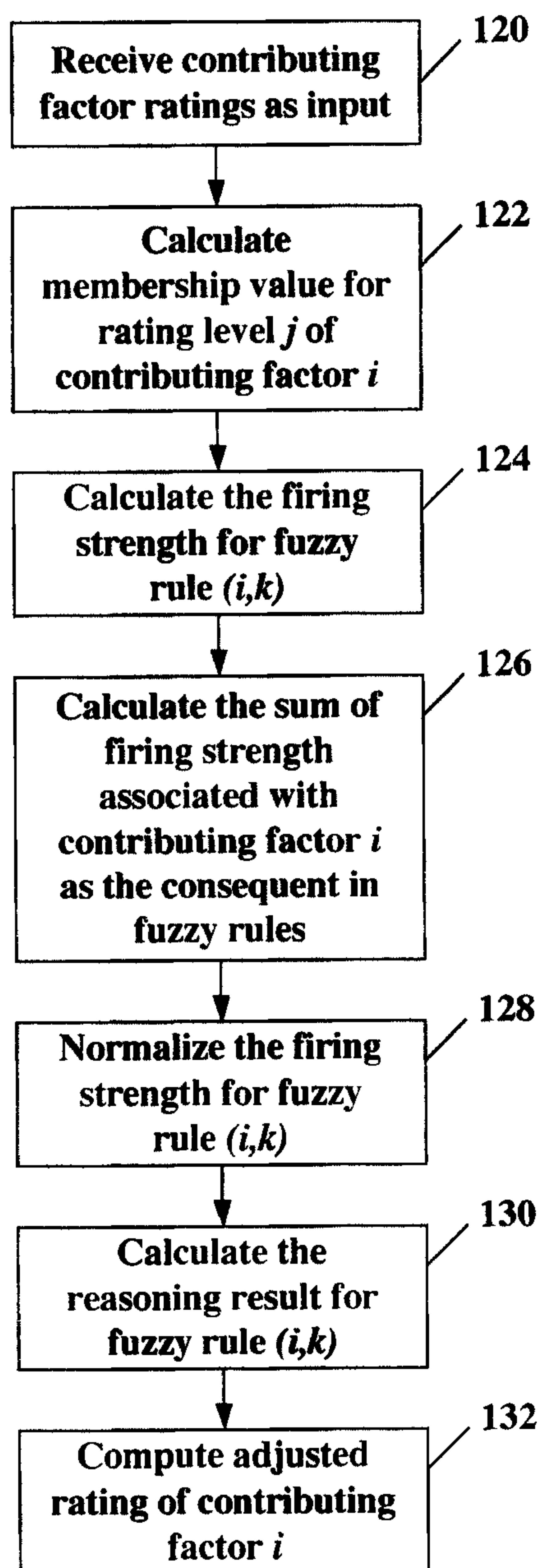


FIG. 4

110
↙**FIG. 5**

112
↙

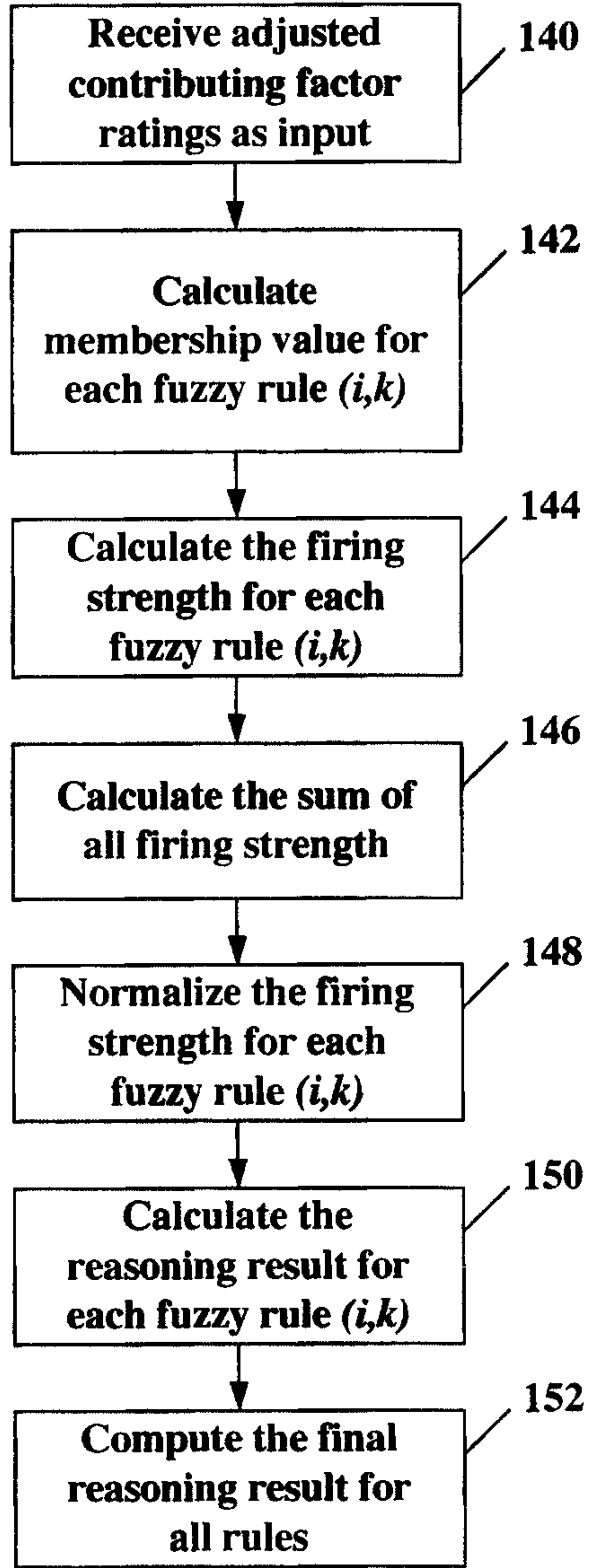


FIG. 6

10

