5 ТЕОРИЯ, ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ И ЯЗЫКИ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ

Голосовский М.С.

МОДЕЛЬ ОЦЕНИВАНИЯ ПОГРЕШНОСТЕЙ ПРОГНОЗИРОВАНИЯ СРОКОВ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Аннотация: *Предметом исследования является сфера разработки программного обеспечения. Важной прикладной задачей является достоверное прогнозирование сроков разработки программного обеспечения. Однако отсутствие массива типовых работ по созданию программного обеспечения, время выполнения которых заранее определено регламентом или замерено с использованием хронометража, существенно затрудняет решение названной задачи. С учетом такой неопределенности создана математическая модель оценивания погрешностей прогнозирования сроков разработки программного обеспечения на основе нечёткой адаптивной системы, основанной на массиве продукционных правил, антецеденты и консеквенты которых представлены лингвистическими переменными. Методология исследования объединяет методы программной инженерии и методы нечеткого логического вывода с применением нечетких регуляторов на основе продукционных правил. Основным результатом исследования является модель оценивания погрешностей прогнозирования сроков разработки программного обеспечения на основе нечёткой адаптивной системы и результаты исследования ее потенциальной эффективности. Показано, что достоинствами разработанной модели являются: возможность получения оценки ошибки на основе экспертных оценок при отсутствии статистических данных; возможность корректировки модели в ходе выполнения проекта; устойчивость модели к разовым шумовым изменениям в результирующих значениях;возможность переноса модели в новый проект.*

Ключевые слова: *программная инженерия, разработка программ, время разработки программы, нечеткий регулятор, программное обеспечение, оценка времени программирования, продукционный логический вывод, затраты на программирование, сложность программного обеспечения, проект разработки программы*

Abstract: *The subject of research is in the area of software development. An important task for this*

DOI: 10.7256/2305-6061.2015.3.17100

field is to accurately predict the timing of software development. However, the lack of standard set of tasks for which the execution time is predetermined or measured considerably complicates time management. Given these uncertainties, the author created a mathematical model of estimation of errors in predicting the timing of software development based on adaptive fuzzy system with an array of production rules, the antecedents and the consequent for which are represented by linguistic variables. The research methodology combines software engineering techniques and methods of fuzzy inference using fuzzy controllers based on production rules. The main result of the research is in building a model of estimation of errors in predicting the timing of software development based on fuzzy adaptive control system and the studies of its potential effectiveness. It is shown that the advantages of the developed model are: the possibility of obtaining error estimates based on expert judgment in the absence of statistical data; the possibility of adjusting the model during the project; the stability of the model to the one-time changes in the resulting noise values; the ability to transfer the model in a new project.

Keywords: *productional logical conclusion, evaluation of programming time, software, fuzzy controller, development time program, software development, software engineering, programming costs, complexity of the software, software development project*

Сфера разработки программного обеспечения отличается от других в первую очередь тем, что достаточно сложно выделить типовые работы, время выполнения которых заранее определено регламентом или замерено с использованием хронометража. Согласно различным методикам управления проектами, получение окончательной оценки сроков выполнения проекта формируется на основании обобщения мнения экспертов [1, 2]. В связи с этим возникает потребность в разработке модели формирования уточнённой оценки сроков выполнения задач, полученных в результате декомпозиции атомарных задач с использованием не только экспертных оценок, но и статистических результатов получаемых как в ходе текущего проекта, так и предыдущих проектов, выполненных с привлечением участников текущего проекта. Для построения модели приняты следующие гипотезы [3, 4]:

– чем больше декомпозиция (чем меньше финальная длительность задачи), тем более точной является оценка;
– чем более сложной (неопределённой) является задача, тем менее точной является оценка;
– чем больше у разработчика опыт в соответствующей проекту предметной области и технологиями, используемыми в проекте, тем более точной является оценка.

Проверку гипотез осуществим с использованием статистических данных собранных на 5 проектах длительностью от трех месяцев до года. Общий объём выборки составляет 9011 оцененных и выполненных задач. Распределение зависимости величины отношения оценки задачи/срок выполнения (относительной величины) от реальной величины задачи представлено на рисунке 1.

DOI: 10.7256/2305-6061.2015.3.17100

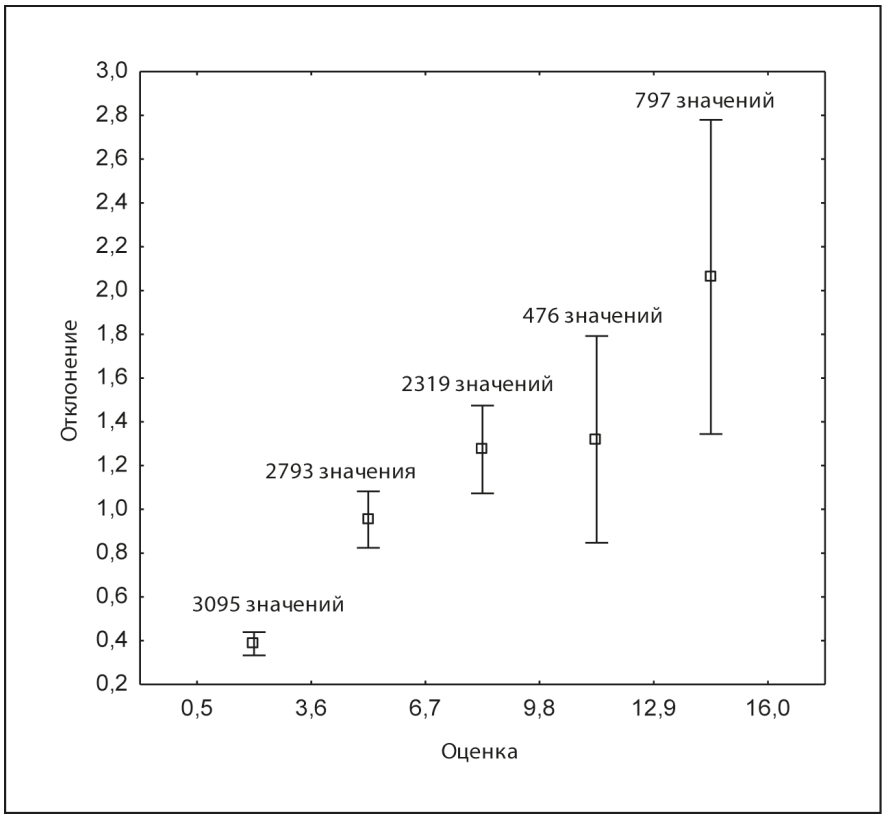


Рисунок 1 – Зависимость величины отклонения (ошибки) выполнения задачи от оценочного значения (центр – среднее арифметическое, размах – 95%-доверительный интервал).

Для решения задачи прогнозирования времени выполнения работ зададим модель в виде

$$M = <T, E, F, S>,$$

где T – (*Tasks*) множество работ, которые необходимо выполнить, для завершения проекта $T = \{t1, t2,...,tn\}$, n – число задач в проекте;

E – (Estimate) множество оценок трудозатрат для работ из множества T, $E = \{e1, e2,...,en\}$;

F – (Factor) множество факторов, влияющих на получение оценки $F = \{f1, f2, ..., fm\}$, m – число факторов, влияющих на точность оценки;

S – (Spent time) время, затраченное на выполнение задачи из множества T, $S = \{s1, s2,...,sn\}$.

В идеальной ситуации должно выполняться соотношение:

$$\Delta E_i = E_i - S_i \to \forall i \in [0, n]$$

Исходные данные заполняются исходя из модели корректировки погрешности оценок при итеративно-инкрементном подходе к разработке ПО, представленной на рисунке 2.

DOI: 10.7256/2305-6061.2015.3.17100

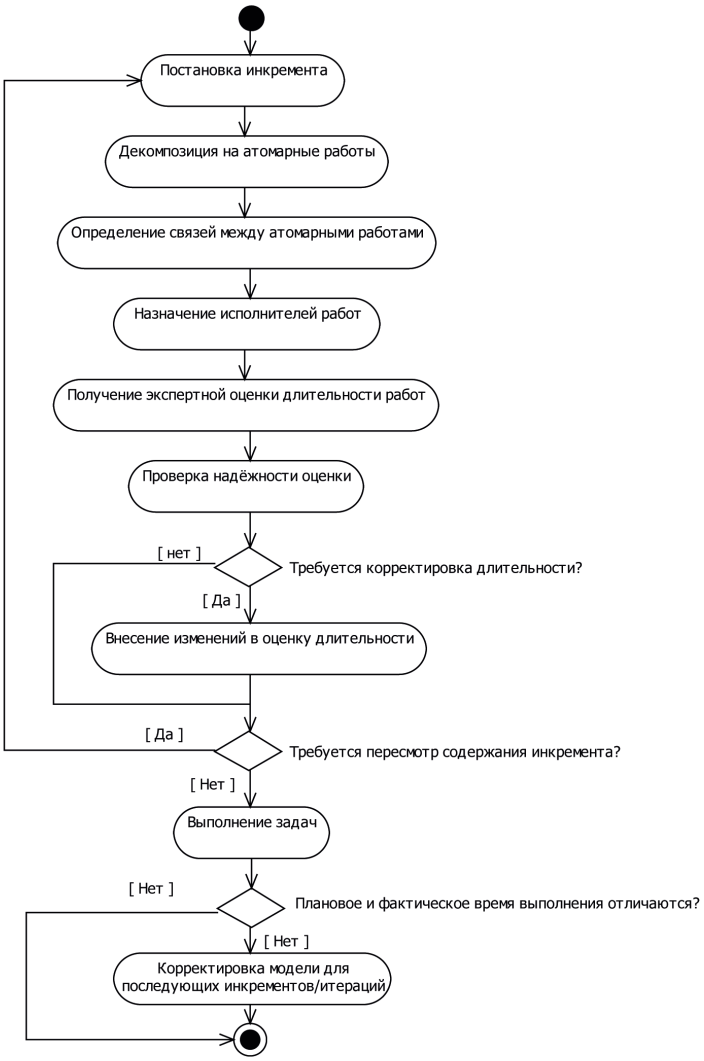


Рисунок 2 – Алгоритм корректировки погрешности оценок при итеративно-инкрементном подходе к разработке ПО

Согласно этой модели, на начальном этапе выбирается блок работ для реализации и производится постановка инкремента, как большой высокоуровневой задачи [5, 6]. После чего, с использованием методов декомпозиции строится иерархическая структура работ, на основе которой формируется множество атомарных работ T [7, 8].

Следующим шагом формируются связи между работами, для определения последовательности их выполнения после чего, работам назначаются исполнители, совместно с которыми (или без них) формируется множество оценок E [9, 10].

Для полученных оценок, с учетом факторов влияния F, проводится проверка надёжности полученной оценки, и задаётся прогнозируемая величина ошибки $\Delta E_{П}$. Суммарная оценка времени выполнения задач служит основой для принятия решения о начале выполнения инкремента, или о пересмотре содержания в сторону его уменьшения или увеличения [11, 12]. Полученные в процессе выполнения работ фактические значения длительности, служат основой для корректировки элементов модели, в частности настройки весов системы прогнозирования. Для её корректной работы сформируем набор факторов F:

- сложность задачи (чем сложнее, тем менее точна оценка);
- первоначальная оценка длительности;
- переключаемость исполнителя во время выполнения задачи (как часто человек отвлекается);
- время на проекте;
- общий опыт разработчика;
- время, выделенное на планирование;
- планирование и оценка выполнялись коллегиально или нет

DOI: 10.7256/2305-6061.2015.3.17100

В разрабатываемой системе проверку надёжности оценки будем проводить на основе системы нечёткого вывода, в которой входными значениями являются факторы, от которых зависит оценка выходной переменной, будет прогнозируемая величина ошибки оценки [13, 14].

Для работы в системе логического вывода каждый фактор fi описывается нечеткой лингвистической переменной с термами Aij (i – номер фактора, j – номер терма в рамках выбранного фактора, при этом количество термов может быть от двух до нескольких десятков и изменяться в рамках каждого фактора: $j \in [1, k_i]$, где k_i – число термов, описывающих i-й фактор). На практике в системах с ручным определением термов редко используется количество термов больше 10 [15]. Каждый терм описывается функцией принадлежности $\mu_{ij}(x)$ при помощи которой происходит определение степени принадлежности чёткого значения входной переменной терму [16-18].

Генерируется набор продукционных правил вида:

$$\text{ЕСЛИ } x_1 \text{ ЕСТЬ } A_{1j} \wedge x_2 \text{ ЕСТЬ } A_{2j} \wedge \ldots \wedge x_n \text{ ЕСТЬ } A_{nj} \text{ ТО } y \text{ ЕСТЬ } B_j$$

где x – значения входных переменных, y – значение выходной переменной.

Нечёткий логический вывод проводится по методу Мамдани [19]. Для процедуры дефаззификации будем использовать центроидный метод

$$E = \frac{\int_{\min}^{mox} x\mu(x)dx}{\int_{\min}^{mox} \mu(x)dx},$$

где $\mu(x)$ – результирующая функция принадлежности, полученная посредством суммирования частных усечённых функций принадлежности $\mu_{jr}(x)$, полученных при срабатывании j-го правила.

С целью подстройки системы под результаты фактических измерений для каждого правила накапливается ошибка R:

$$R_{ij} = \sum_{i=1}^{N} w_i \ln \frac{S_i}{E_i},$$

где j – номер сработавшего правила, i – номер оценки E, полученной при срабатывании j-го правила, и соответствующего этой оценки реально затраченного времени S, w_i – вес вклада результирующей функции принадлежности j-го правила при формировании i-й оценки, вычисляется как:

DOI: 10.7256/2305-6061.2015.3.17100

$$w_i = \begin{cases} 1 - \dfrac{E_i - e_i}{E_i - E_{\min}} \max(\mu_{ir}(x)); \; при \;\; E_i > e_i \\ 1 - \dfrac{e_i - E_i}{E_i - E_{\max}} \max(\mu_{ir}(x)); \; при \;\; E_i < e_i \\ 1; \; при \;\; E_i = e_i \end{cases}$$

где $\max(\mu_{ir}(x))$ максимальное значение усечённой функции принадлежности, e_i центр тяжести i-й усечённой результирующей функции принадлежности, рассчитываемый как

$$e_i = \frac{\int_{\min}^{mox} x\mu_{ir}(x)dx}{\int_{\min}^{mox} \mu_{ir}(x)dx} .$$

Эта формула реализована для общего случая, в котором функция принадлежности может быть произвольной формы [5, 18]. При использовании симметричных функций принадлежности, проекция центра масс будет совпадать с центром фигуры и, в частности, для гауссовых функций принадлежности совпадает с ядром функции – точкой, в которой исходное значение функции принадлежности равно 1. В случае если знак суммы ошибок R_j отрицательный, то решение о смене следствия для j-го правила с B_k на B_k-1, принимается при выполнении неравенства:

$$R_j \geq \left| C_k^j - C_{k-1} \right| ,$$

где C – ядро k-й функции принадлежности результирующей лингвистической переменной, в случае, если знак суммы ошибок положительный, то рассматривается возможность о смене заключения с B_k на B_{k+1} при условии выполнения неравенства:

$$R_j \geq \left| C_{k+1} - C_k^j \right| .$$

Проведём проверку устойчивости системы к ступенчатому изменению ошибки. Для этого зададим две системы. Каждая содержит одну входную и одну выходную переменную. Обе системы будут иметь одинаковую входную переменную с областью определения [0, 25], содержащая 8 функций принадлежности $(m1 \ldots m8)$ гауссова типа. Выходная переменная первой системы будет содержать 8 функций принадлежности, выходная переменная второй системы будет содержать 100 функций принадлежности, ядра которых равномерно распределены на области определения [0, 1]. Графически входная и выходные переменные изображены на рисунке 3.

В каждой системе задано 8 правил вывода. На вход обеих систем на подаётся только одно значение, не изменяемое в процессе работы и равное 12. На первых 10 замерах смоделированное реальное значение равно выходу из системы. Для первой системы это

DOI: 10.7256/2305-6061.2015.3.17100

значение равно 0,39, для второй 0,25. Начиная с 11 замера смоделированное реальное значение было изменено на 0,6, из за чего, обе системы начали подстройку правил, по приведённому выше алгоритму. Графики обработки ступенчатого изменения реального значения наблюдаемой переменной приведены на рисунке 4.

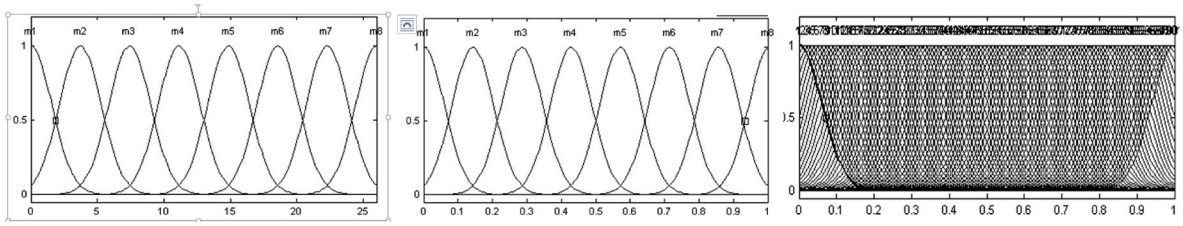
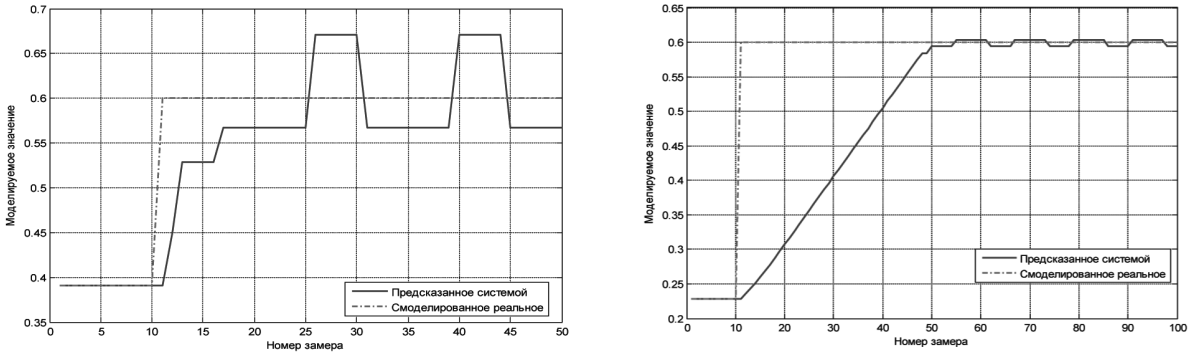


Рисунок 3 – Функции принадлежности переменных: входной (а) и выходной с 8 (б) и 100 (в) функциями принадлежности



а) б)

Рисунок 4 – Графики обработки ступенчатого изменения реального значения наблюдаемой переменной при 8 (а) и 100 (б) функциях принадлежности выходной переменной

На основании полученных данных можно сделать вывод, что система с большим количеством функций переменных более точно подстраивает выходное значение под моделируемое реальное значение, при этом точность определяется разностью расстояний между ядрами функций принадлежности, но система с не большим числом переменных более быстро реагирует на изменения - 6 шагов вместо 40. В связи с тем, что экспертные оценки длительности задач на разработку имеют высокую степень неопределённости, для практического применения можно использовать системы с количеством функций принадлежности в выходной переменной в пределах 10.

Рассмотрим решение задачи с использованием правил нечёткого вывода. Зададим систему нечёткого логического вывода первоначально для двух входных факторов: сложность задачи и первоначальная оценка длительности (для описания будем использовать более короткий термин оценка выполнения), описав эти задачи в виде нечётких лингвистических переменных со следующими функциями принадлежности гауссова типа:

Лингвистическая переменная «Оценка выполнения» с возможными значениями: «маленькая», «средняя», «большая», «очень большая», «огромная» задана в соответствии

DOI: 10.7256/2305-6061.2015.3.17100

с рисунком 5.

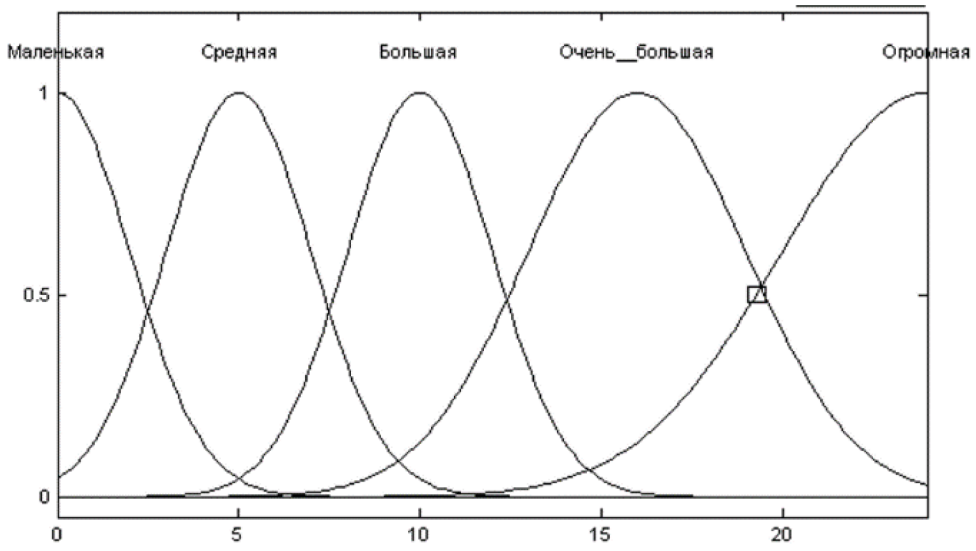


Рисунок 5 – Графическое представление лингвистической переменной «Оценка выполнения»

Лингвистическая переменная «Сложность задачи» с возможными значениями: «тривиальная», «легкая», «средняя», «сложная», «невозможная» задана в соответствии с рисунком 6.

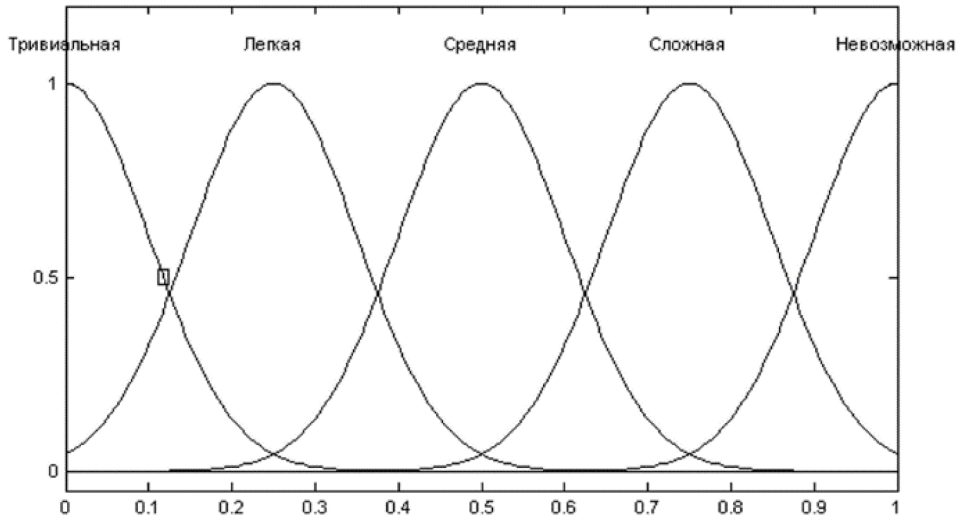


Рисунок 6 – Графическое представление лингвистической переменной «Сложность задачи»

Лингвистическая переменная «Величина ошибки» с возможными значениями: «маленькая», «небольшая», «средняя», «больше средней», «большая», «очень большая», «огромная» задана в соответствии с рисунком 7.

DOI: 10.7256/2305-6061.2015.3.17100

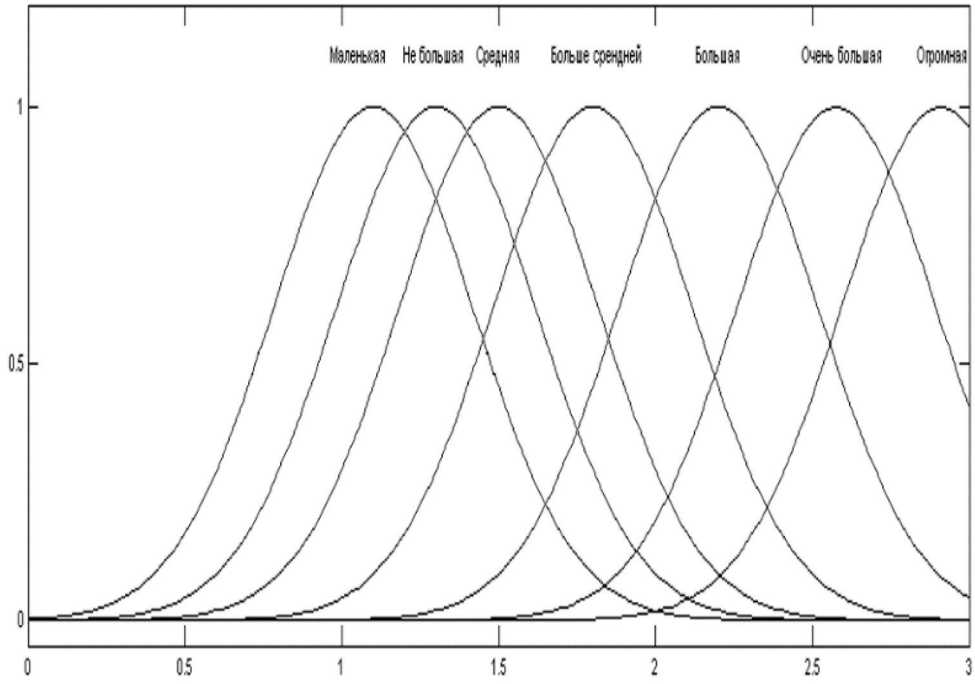
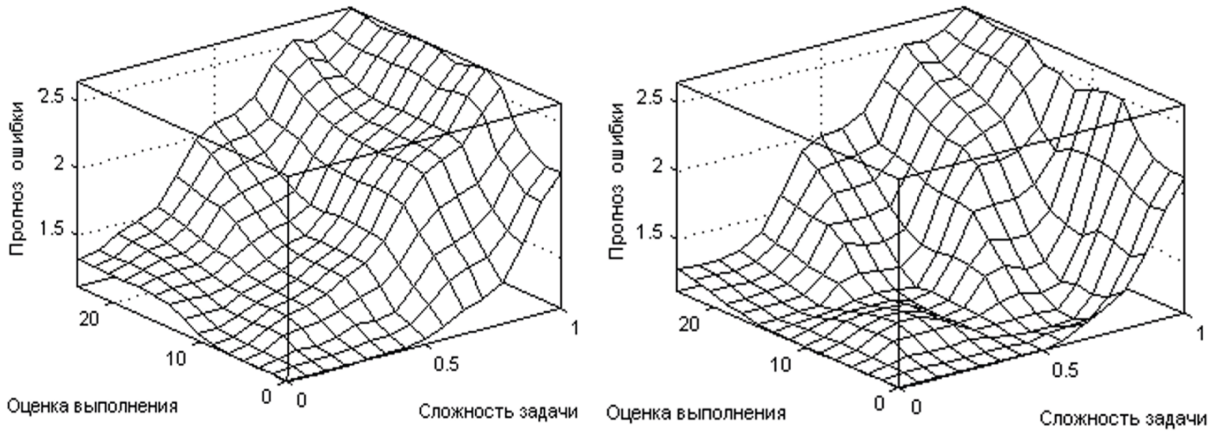


Рисунок 7 – Графическое представление лингвистической переменной «Сложность задачи»

Исследование качества предложенной модели проведено на выборке из 50 задач для одного разработчика. Суммарная оценка составила 299 часов, предсказание системы составило 426 человеко-часа разработки, при этом реально было затрачено 334 часа работы разработчика. Большая часть задач была выполнена в рамках данной разработчиком оценки. По результатам выполнения итерации была выполнена корректировка правил экспертной системы, в соответствии, с предложенным алгоритмом. Изменение поверхности срабатывания правил показано на рисунке 8.



а) б)

Рисунок 8 – Изменение поверхности срабатывания правил системы после корректировки правил (а – исходная поверхность, б – поверхность после корректировки)

DOI: 10.7256/2305-6061.2015.3.17100

Для второй итерации отобрали следующие 50 задач с оценкой трудозатрат 307 часов, предсказанное системой значение составило 371 час, реально затраченное 363 часа. Приведённый пример показывает применимость предложенного метода на практике. Для получения более объективных результатов необходим учёт большего числа факторов в качестве входных переменных.

В качестве достоинств предложенного метода можно выделить:

– получение оценки ошибки на основе экспертных оценок при отсутствии статистических данных;
– корректировка модели в ходе выполнения проекта на основе полученных результатов;
– модель устойчива к разовым шумовым изменениям в результирующих значениях;
– возможность переноса модели в новый проект.
– Недостатками предложенного метода являются:
– необходимо формировать и поддерживать базу правил для каждого участника проекта;
– при небольшом количестве термов входных и выходных переменных снижается точность аппроксимации выходных значений.
– при большом количестве термов выходных переменных снижается скорость подстройки значений. указанный недостаток можно исправить изменив алгоритм смены функции принадлежности в заключении правила, позволив переходить не на следующую функцию принадлежности, а сразу через некоторое количество, которое зависит от величины ошибки.

Библиография :

1. Каратыгин С.А., Дукин А.Н. Оценка сроков разработки программного обеспечения // Естественные и технические науки. 2009. № 4 (42). С. 350-354.
2. Голосовский М.С. Модель жизненного цикла разработки программного обеспечения в рамках научно-исследовательских работ // Автоматизация. Современные технологии. 2014. № 1. С. 43-46.
3. Фёдоров М.В., Калинин К.М., Богомолов А.В., Стецюк А.Н. Математическая модель автоматизированного контроля выполнения мероприятий в органах военного управления // Информационно-измерительные и управляющие системы. 2011. Т. 9. № 5. С. 46-54.
4. Голосовский М.С. Информационно-логическая модель процесса разработки программного обеспечения // Программные системы и вычислительные методы. 2015. № 1. С. 59-68.
5. Кукушкин Ю.А., Богомолов А.В., Ушаков И.Б. Математическое обеспечение оценивания состояния материальных систем // Информационные технологии. 2004. № 7 (приложение). 32 с.
6. Гольфанд И.Я., Хлебутин П.С. Оценка трудозатрат разработки программной компоненты // Труды ИСА РАН, Т.15, 2005. С. 125-135.

DOI: 10.7256/2305-6061.2015.3.17100

7. Козлов В.Е., Богомолов А.В., Рудаков С.В., Оленченко В.Т. Математическое обеспечение обработки рейтинговой информации в задачах экспертного оценивания // Мир измерений. 2012. № 9. С. 42-49.
8. Bourque P., Fairley R.E. Guide to the Software Engineering Body of Knowledge (SWEBOK) Version 3.0. IEEE Computer Society, 2013. 335 p.
9. Щеглов И.Н., Печатнов Ю.А., Богомолов А.В. Интенсификация разработки автоматизированных систем обучения на основе нейросетевых технологий // Информационные технологии. 2003. № 4. С. 31.
10. Афанасьев Ю.И. Устойчивости автоматизированных систем. Концепция устойчивого взаимодействия // Образовательные ресурсы и технологии. – 2014.-№ 5(8). [Электронный ресурс]. URL:http://www.muiv.ru/vestnik/pdf/pp/ot_2014_5_86-94.pdf
11. Рудаков И.С., Рудаков С.В., Богомолов А.В. Методика идентификации вида закона распределения параметров при проведения контроля состояния сложных систем // Информационно-измерительные и управляющие системы. 2007. Т. 5. № 1. С. 66-72.
12. Есев А.А., Мережко А.Н., Ткачук А.В. Технология квалиметрии технического уровня сложных систем // Вестник компьютерных и информационных технологий. 2014. № 7 (121). С. 28-34.
13. Iancu I. Extended Mamdani Fuzzy Logic Controller // The 4th IASTED Int. Conf. on Computational Intelligence, ACTA Press, Honolulu, USA, 2009. PP. 143–149.
14. Mann G.K.I., Bao-Gang Hu, Gosine R.G. Analysis of direct action fuzzy PID controller structures // IEEE Transactions on Systems, Man and Cybernetics, Part B. June 1999. Vol. 29. Issue 3. P. 371-388.
15. Takagi T., Sugeno M. Fuzzy identification of systems and its applications to modeling and control // IEEE Trans. Systems Man Cybernet. – 1985. – Vol. 15. – № 116. – P.116–132.
16. Zadeh, L.A. A theory of approximate reasoning // Machine Intelligence. New York: John Wiley & Sons, 1979. 194 p.
17. Заде Л.А. Понятие лингвистической переменной и его применение к принятию приближенных решений. М.: Мир, 1976. 165 с.
18. Усков А.А., Круглов В.В. Интеллектуальные системы управления на основе методов нечеткой логики // Смоленская городская типография, 2003. 177 с.
19. Mamdani E.H. Application of fuzzy logic to approximate reasoning using linguistic synthesis // IEEE Transactions on Computers, №26(12), 1977. PP 1182–1191

References:

1. Karatygin S.A., Dukin A.N. Otsenka srokov razrabotki programmnogo obespecheniya // Estestvennye i tekhnicheskie nauki. 2009. № 4 (42). S. 350-354.
2. Golosovskii M.S. Model' zhiznennogo tsikla razrabotki programmnogo obespecheniya v ramkakh nauchno-issledovatel'skikh rabot // Avtomatizatsiya. Sovremennye tekhnologii. 2014. № 1. S. 43-46.
3. Fedorov M.V., Kalinin K.M., Bogomolov A.V., Stetsyuk A.N. Matematicheskaya model' avtomatizirovannogo kontrolya vypolneniya meropriyatii v organakh voennogo upravleniya // Informatsionno-izmeritel'nye i upravlyayushchie sistemy. 2011. T. 9. № 5. S. 46-54.
4. Golosovskii M.S. Informatsionno-logicheskaya model' protsessa razrabotki programmnogo obespecheniya // Programmnye sistemy i vychislitel'nye metody. 2015. № 1. S. 59-68.

DOI: 10.7256/2305-6061.2015.3.17100

5. Kukushkin Yu.A., Bogomolov A.V., Ushakov I.B. Matematicheskoe obespechenie otsenivaniya sostoyaniya material'nykh sistem // Informatsionnye tekhnologii. 2004. № 7 (prilozhenie). 32 s.
6. Gol'fand I.Ya., Khlebutin P.S. Otsenka trudozatrat razrabotki programmnoi komponenty // Trudy ISA RAN, T.15, 2005. S. 125-135.
7. Kozlov V.E., Bogomolov A.V., Rudakov S.V., Olenchenko V.T. Matematicheskoe obespechenie obrabotki reitingovoi informatsii v zadachakh ekspertnogo otsenivaniya // Mir izmerenii. 2012. № 9. S. 42-49.
8. Bourque P., Fairley R.E. Guide to the Software Engineering Body of Knowledge (SWEBOK) Version 3.0. IEEE Computer Society, 2013. 335 p.
9. Shcheglov I.N., Pechatnov Yu.A., Bogomolov A.V. Intensifikatsiya razrabotki avtomatizirovannykh sistem obucheniya na osnove neirosetevykh tekhnologii // Informatsionnye tekhnologii. 2003. № 4. S. 31.
10. Afanas'ev Yu.I. Ustoichivosti avtomatizirovannykh sistem. Kontseptsiya ustoichivogo vzaimodeistviya // Obrazovatel'nye resursy i tekhnologii. – 2014.-№ 5(8). [Elektronnyi resurs]. URL:http://www.muiv.ru/vestnik/pdf/pp/ot_2014_5_86-94.pdf
11. Rudakov I.S., Rudakov S.V., Bogomolov A.V. Metodika identifikatsii vida zakona raspredeleniya parametrov pri provedeniya kontrolya sostoyaniya slozhnykh sistem // Informatsionno-izmeritel'nye i upravlyayushchie sistemy. 2007. T. 5. № 1. S. 66-72.
12. Esev A.A., Merezhko A.N., Tkachuk A.V. Tekhnologiya kvalimetrii tekhnicheskogo urovnya slozhnykh sistem // Vestnik komp'yuternykh i informatsionnykh tekhnologii. 2014. № 7 (121). S. 28-34.
13. Iancu I. Extended Mamdani Fuzzy Logic Controller // The 4th IASTED Int. Conf. on Computational Intelligence, ACTA Press, Honolulu, USA, 2009. RR. 143–149.
14. Mann G.K.I., Bao-Gang Hu, Gosine R.G. Analysis of direct action fuzzy PID controller structures // IEEE Transactions on Systems, Man and Cybernetics, Part B. June 1999. Vol. 29. Issue 3. P. 371-388.
15. Takagi T., Sugeno M. Fuzzy identification of systems and its applications to modeling and control // IEEE Trans. Systems Man Cybernet. – 1985. – Vol. 15. – № 116. – P.116–132.
16. Zadeh, L.A. A theory of approximate reasoning // Machine Intelligence. New York: John Wiley & Sons, 1979. 194 p.
17. Zade L.A. Ponyatie lingvisticheskoi peremennoi i ego primenenie k prinyatiyu priblizhennykh reshenii. M.: Mir, 1976. 165 s.
18. Uskov A.A., Kruglov V.V. Intellektual'nye sistemy upravleniya na osnove metodov nechetkoi logiki // Smolenskaya gorodskaya tipografiya, 2003. 177 s.
19. Mamdani E.H. Application of fuzzy logic to approximate reasoning using linguistic synthesis // IEEE Transactions on Computers, №26(12), 1977. RR 1182–1191

DOI: 10.7256/2305-6061.2015.3.17100