

Розглянута проблема ефективності використання пост об'єктно-орієнтованих технологій (ПООТ) в успадкованій програмній системі. Запропонований підхід для визначення ефективності ПООТ на основі нечіткої логіки та алгоритму Мамдані. Отримані кількісні оцінки ефективності використання ООП та трьох ПООТ підходів на прикладі успадкованої підсистеми управління персональними даними

Ключові слова: наскрізна функціональність, пост об'єктно-орієнтована технологія, оцінка ефективності, алгоритм Мамдані

Рассмотрена проблема эффективности применения пост объектно-ориентированных технологий (ПООТ) в унаследованной программной системе. Предложен подход для определения эффективности ПООТ на основе нечеткой логики и алгоритма Мамдани. Получены количественные оценки эффективности использования ООП и трех ПООТ подходов на примере унаследованной подсистемы управления персональными данными

Ключевые слова: сквозная функциональность, пост объектно-ориентированная технология, оценка эффективности, алгоритм Мамдани

РАЗРАБОТКА И ПРИМЕНЕНИЕ МЕТОДИКИ ОЦЕНКИ ЭФФЕКТИВНОСТИ ПОСТ ОБЪЕКТНО-ОРИЕНТИРОВАННЫХ ТЕХНОЛОГИЙ

К. А. Нагорный

Ассистент

Кафедра автоматизированных систем управления
Национальный технический университет
«Харьковский политехнический институт»
ул. Фрунзе, 21, г. Харьков, Украина, 61002
E-mail: k.nagorny@gmail.com

1. Введение

Сквозная функциональность – СФ (crosscutting concern) [1, 2], является одной из проблем разработки и сопровождения унаследованных программных систем (УПС) [3] на основе объектно-ориентированного подхода (ООП). В литературе определен подход, позволяющий выделить программный код СФ и локализовать его – разделение функциональности (separation of concerns - SoC) [1, 4]. На сегодняшний день существует подмножество технологий, объединяющих парадигму ООП и идею SoC, наиболее развитыми является пост объектно-ориентированные технологии (ПООТ) [5 – 7]: аспектно-ориентированная технология (aspect-oriented software development - AOSD), технология, ориентированная на свойства (feature-oriented software development - FOSD), контекстно-ориентированная технология (context-oriented software development - COSD). Приведенные ПООТ имеют ряд преимуществ и недостатков, что усложняет мотивированный выбор той или иной технологии.

Качественное сравнение основных свойств AOSD, FOSD, COSD приводится в [8 – 10], однако на его основе, сложно определить, какая из технологий наиболее эффективна при сопровождении УПС, поэтому возникает необходимость в дополнительных количественных оценках эффективности применения этих ПООТ.

Исследования количественных характеристик рассматриваемых технологий можно разделить на две группы: (1) применение метрик исходного кода, для измерения сложности программных компонентов в которых присутствует СФ; (2) применение метрик

сквозной функциональности для измерения степени присутствия СФ в ПС.

К первой группе можно отнести [11], где рассматривается 11 программ, реализованных на основе ООП и модифицированных с использованием AOSD, сравнение происходит по метрике количества строк кода (Lines of Code - LOC) и ее производным, а также по количеству классов. В [12] сравниваются ООП- и AOSD-программные реализации, здесь используется несколько объектно-ориентированных метрик, таких как сцепление между объектами (Coupling Between Object classes - CBO), недостаток связности методов (Lack of Cohesion in Methods - LOCM) и LOC. В [13] для сравнения программных систем на основе ООП и FOSD использованы метрики: LOC, количество свойств (feature).

Ко второй группе можно отнести [14], в этой работе проанализированы некоторые количественные метрики отдельных свойств СФ: запутывание (tangling) и рассеивание (scattering). Приведена методика систематизации таких разрозненных метрик, что позволило определить их единицы измерения и выявить метрики-клоны. Предложена классификация СФ на основе шаблонов СФ, и эвристические стратегии применения такой классификации для распознавания СФ кода программной системы. В работах [15, 16] рассмотрено некоторое подмножество метрик СФ, имеющих различные единицы измерения, что позволяет лишь попарно сравнивать результаты измерений. В то же время, подходы первой группы не учитывают специфику конструкций ПООТ, а методы оценки второй группы предлагают лишь отдельные, локальные метрики СФ, и при этом не предлагаются способы снижения СФ [14].

Таким образом, проблема оценки эффективности применения ПООТ остается достаточно актуальной, и целью данной статьи является разработка и экспериментальное исследование комплексной методики оценки эффективности применения таких технологий для реинжиниринга унаследованных программных систем (УПС) с учетом их различных характеристик.

2. Подход к оценке эффективности пост объектно-ориентированных технологий на основе нечеткой логики

В работе [17] предложен подход к оценке сложности ПООТ. Элементы каждой технологии представляются в единой нотации языка определения архитектур ADL [18], учитывая взаимосвязи между конструктивными элементами, были построены архитектурные модели - расширенные архитектурные примитивы – РАП. На основе РАП получены количественные оценки сложности каждой ПООТ, выраженные в архитектурных единицах – А.Е. Для оценки затрат на ПООТ-модификацию УПС предлагается использовать принципы построения и оценки РАП. Таким образом, модификации целевой системы будут выражены в единой нотации языка ADL для любой рассматриваемой ПООТ, оценки будут исчисляться в А.Е.

Для количественной оценки степени снижения СФ в результате применения ПООТ, предлагается использовать метрику CFD_{ratio} (Crosscutting functionality decrease ratio), результат вычисления выражается в %, (1):

$$CFD_{ratio} = DOS \cdot CF_{ratio}, \tag{1}$$

где метрика DOS (Degree of scattering) – степень рассеивания, [15],

метрика CF_{ratio} (Crosscutting Functionality Ratio) – показатель сквозной функциональности, предложена в [8].

На основе оценки затрат на применение ПООТ в целевой ПС и степени снижения СФ можно вычислить эффективность использования ПООТ, однако т.к. указанные оценки имеют разные единицы исчисления, то возникает сложность в их использовании в одном математическом выражении. Для дальнейших преобразований предлагается использовать один из алгоритмов нечеткой логики – алгоритм Мамдани, который состоит из 6 шагов, см. например в [19].

В соответствии с алгоритмом Мамдани, для оценки эффективности ПООТ, необходимо сформировать базу нечетких правил продукций. Для упрощения выкладок, в данной работе, пропущено вербальное описание правил и приводится их фрагмент в сокращенной форме.

В табл. 1 приведены общепринятые сокращения [19]. Рассматриваемая система правил нечетких продукций состоит из 20 правил. Весовые коэффициенты определенности для всех правил $F_i \in [0,1]$, $F_i = 1$, где $i = 1, 20$ и они имеют следующий вид:

1. ...
2. ПРАВИЛО_10: Если « β_1 есть РМ» И « β_2 есть РМ», ТО « β_3 есть PS»;
3. ПРАВИЛО_11: Если « β_1 есть РВ» И « β_2 есть РМ», ТО « β_3 есть PS»;

4. ПРАВИЛО_12: Если « β_1 есть РН» И « β_2 есть РМ», ТО « β_3 есть Z»;
5. ПРАВИЛО_13: Если « β_1 есть PS» И « β_2 есть РВ», ТО « β_3 есть РН»;
6. ПРАВИЛО_14: Если « β_1 есть РМ» И « β_2 есть РВ», ТО « β_3 есть РВ»;
7. ...

Таблица 1

Общепринятые сокращения для значений основных термов лингвистических переменных в системах нечеткого вывода

Символьное обозначение	Англоязычная нотация	Русскоязычная нотация
NB	Negative Big	Отрицательное большое
NM	Negative Middle	Отрицательное среднее
NS	Negative Small	Отрицательное малое
ZN	Zero Negative	Отрицательное близкое к нулю
Z	Zero	Нуль, близкое к нулю
ZP	Zero Positive	Положительное близкое к нулю
PS	Positive Small	Положительное малое
PM	Positive Middle	Положительное среднее
PB	Positive Big	Положительное большое
PH	Positive Huge	Положительное очень большое

Следующим шагом, согласно алгоритму Мамдани, является фаззификация переменных. Для этого затраты на использования ПООТ, степень снижения СФ и эффективность использования ПООТ должны быть представлены в форме лингвистических переменных.

Входная лингвистическая переменная $C_{роот}$ - затраты на реализацию ПС в пост объектно-ориентированной технологии – ПООТ. $C_{роот}$ ограничена на универсуме X и принадлежит интервалу [4, 3; 150], где 4,3 – сложность РАП для ООП (минимально возможная сложность архитектурно схемы), 150 – сложность архитектурной схемы ПС (максимальная сложность в рамках данного исследования). $C_{роот}$ измеряется в А.Е. Терм-множество имеет вид: $C_{роот} \in \{низкие, средние, высокие, очень высокие\}$, или в сокращенной форме: $C_{роот} \in \{PS, PM, PB, PH\}$. Переменной $C_{роот}$, соответствует идентификатор β_1 в нечетких правилах продукций. Графическая интерпретация функции принадлежности $\mu_c(c)$ приведена на рис. 1.

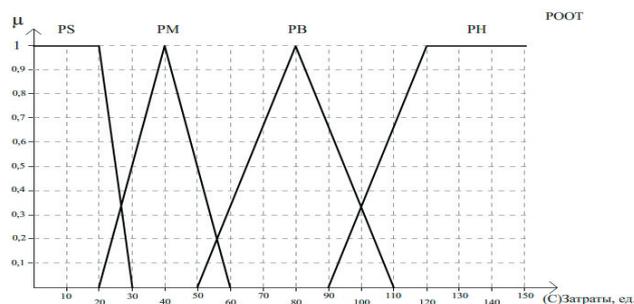


Рис. 1. Фаззификация лингвистической переменной $C_{роот}$

Входная лингвистическая переменная $P_{роот}$ - степень снижения присутствия СФ ПС. Расчет проис-

Таблица 3

Архитектурные элементы для технологии FOSD и COSD

FOSD			COSD		
Архитектурный элемент	Технология	Кол-во	Архитектурный элемент	Технология	Кол-во
Компонент	ООП	33	Компонент	ООП	33
	FOSD	11		COSD	7
	Сумма	44		Сумма	40
Коннектор	ООП	5	Коннектор	ООП	5
	FOSD	3		COSD	7
	Сумма	8		Сумма	12
Порт	ООП	52	Порт	ООП	53
	FOSD	19		COSD	14
	Сумма	71		Сумма	67

ходит на основе количественной метрики СФ CFD_{ratio} , см. выражение (1), P_{POOT} ограничена на универсуме X и принадлежит интервалу $[0;1]$. Терм-множество имеет вид: $P_{POOT} \in \{беспользная, малая, средняя, высокая, очень высокая\}$, или в сокращенной форме: $P_{POOT} \in \{Z, PS, PM, PB, PH\}$. Переменной P_{POOT} соответствует идентификатор β_2 в нечетких правилах продукций. Вид функции принадлежности аналогичен приведенному на рис. 1.

Выходная лингвистическая переменная E_{POOT} - эффективность применения ПООТ в УПС. E_{POOT} ограничена на универсуме X и принадлежит интервалу $[0;1]$. Терм-множество имеет вид: $E_{POOT} \in \{не эффективно, слабо эффективно, достаточно эффективно, эффективно, очень эффективно\}$, или в сокращенной форме: $E_{POOT} \in \{Z, PS, PM, PB, PH\}$. Переменной E_{POOT} соответствует идентификатор β_3 в нечетких правилах продукций.

Вид функции принадлежности аналогичен приведенному на рис. 1.

3. Характеристика входных данных

Рассмотрим предлагаемый подход на примере подсистемы управления персональными данными, состоящей из 13 java-классов, в ней присутствует реализация сквозной функциональности $СФ_1$ – фильтр персональных данных.

В табл. 2 приведены количественные характеристики рассматриваемой ООП подсистемы в соответствии с подходом, предложенным в [17]. В результате применения ПООТ в целевой подсистеме, ее функциональность с точки зрения конечного пользователя не изменилась, однако исходный код подвергся AOSD-, FOSD-, COSD-адаптации. Для расчета затрат на использование ПООТ в целевой подсистеме, все 3 модификации также представлены в соответствии с подходом [17]. В табл. 2, 3 приведены их количественные показатели.

Таблица 2

Архитектурные элементы для технологии ООП и AOSD

ООП			AOSD		
Архитектурный элемент	Технология	Кол-во	Архитектурный элемент	Технология	Кол-во
Компонент	ООП	49	Компонент	ООП	34
Коннектор	ООП	5		AOSD	13
Порт	ООП	82		Сумма	47
			Коннектор	ООП	5
				AOSD	16
				Сумма	21
			Порт	ООП	85
				AOSD	45
				Сумма	130

Каждый архитектурный элемент разделен на две части: ООП-базис, исходный код, который реализует основную функциональность подсистемы, исходный код $СФ_1$ выделен и локализован в ПООТ-надстройке.

В соответствии с выражением (1) для вычисления степени снижения СФ и подходу к оценке затрат на применение ПООТ приведенным в [17], получены количественные показатели, представленные в табл. 4.

Таблица 4

Входные данные для сквозной функциональности $СФ_1$

ПООТ	Затраты на реализацию ПООТ	Степень снижения СФ
ООП	41,45 А.Е.	48,0 %
AOSD	78,63 А.Е.	0,3 %
FOSD	41,93 А.Е.	21,0 %
COSD	39,58 А.Е.	0,5 %

4. Пример расчета и анализ результатов

Рассмотрим вариант расчета эффективности для исходной ООП подсистемы $E_{ООП} = \{41.45; 48\}$, табл. 4.

В соответствии с алгоритмом Мамдани на этапе агрегирования для лингвистической переменной $С_{ООП}$ значение функции принадлежности $\mu(C_{ООП})=0,93$, что отвечает значению термина – РМ. Для лингвистической переменной $P_{ООП}$, функция принадлежности имеет два значения: $\mu(P_{ООП})=0,9$, что соответствует терму РМ и $\mu(P_{ООП})=0,1$, терм – РВ.

Нечеткие правила продукционной модели, соответствующие значениям термов входных переменных – 10 и 14. По формуле логической конъюнкции, приведенной в [19], получим значения для правила 10: $T(C \wedge P) = \min\{0.93, 0.9\}$, т.о. $T(C \wedge P) = 0.9$ и для правила 14: $T(C \wedge P) = \min\{0.93, 0.1\}$, т.о. $T(C \wedge P) = 0.1$. Все остальные правила дают значение $T(C \wedge P) = 0$.

Поскольку агрегирование условий правила 10 дает в результате $T(C \wedge P) = 0.9$, а весовой коэффициент равен 1 (по умолчанию), то будет использовано значение 0,9 для получения результата активации подзаключений.

Агрегирование условий правила 14 дает в результате $T(C \wedge P) = 0.1$, а весовой коэффициент равен 1 (по умолчанию), то будет использовано значение 0,1 для получения результата активации.

На этапе *аккумуляции заключений*, правилу 10, соответствует множество PS, а правилу 14, соответствует множество PB. В результате *дефаззификации* получаем значение эффективности использования объектно-ориентированного подхода для реализации СФ₁, в целевой программной подсистеме: $E_{\text{ООП}} \approx 36\%$. Аналогичным образом можно рассчитать эффективность применения ПООТ: AOSD, FOSD, COSD.

Результаты расчетов приведены в табл. 5.

Таблица 5

Эффективности применения ПООТ в целевой подсистеме

ПООТ	Затраты на реализацию ПООТ $C_{\text{ПООТ}}, \text{А.Е.}$	Степень снижения СФ $P_{\text{ПООТ}}, \%$	Эффективность $E_{\text{ПООТ}}, \%$
ООП	41,45	48,0	36
AOSD	78,63	0,2	72
FOSD	41,93	21,0	70
COSD	39,58	0,5	93

Полученные результаты показывают, что подход ООП является не эффективным для реализации сквозной функциональности, наиболее эффективный подход – COSD, однако два других подхода показы-

вают эффективность не меньше 70%, что указывает на возможность их использования при сопровождении УПС, принимая во внимание их качественные преимущества перед COSD.

5. Выводы

В данной работе предложена методика оценки эффективности применения пост объектно-ориентированных технологий (ПООТ) для устранения проблемы сквозной функциональности (СФ) и рассмотрен пример ее использования при реинжиниринге унаследованной программной системы (УПС). Полученные результаты показали, что наименее эффективным средством для снижения уровня СФ является объектно-ориентированный подход, эффективность которого составляет $\approx 36\%$, в то время как эффективность применения рассмотренных ПООТ для устранения СФ превышает 70%, что указывает на возможность снижения общих затрат на сопровождение УПС, в которых присутствует СФ.

Предлагаемый подход будет использован на заключительном этапе разработки интегрированного CASE-средства для комплексной оценки эффективности использования пост объектно-ориентированных технологий.

Литература

1. Siobhán Clarke et al. Separating Concerns throughout the Development Lifecycle [Текст] // Proceedings of the AspectOriented Programming Workshop at ECOOP, June 14 – 18 1999, Lisbon, Portugal. – pp. 13 – 16.
2. Sven Apel et. al. On the structure of crosscutting concerns: Using Aspects of collaboration? [Текст] / Sven Apel, Don Batory, Marko Rosenmüller // Proceedings of GPCE Workshop on AOPLE'06, October 22 – 26, 2006, Portland, USA.
3. Соммервил И., Инженерия программного обеспечения. [Текст] : 6–е изд. / пер. с англ.; М.: Вильямс, 2002. – 624с.
4. Przybyłek Adam. Post object-oriented paradigms in software development: a comparative analysis [Текст] // Proceedings of the 1st Workshop on Advanced in Programming Languages at International Multiconference on Computer Science and Information Technology, October 15 – 17, 2007. Wisla, Poland. – pp. 1009-1020.
5. Aspect-Oriented Software Development [Электронный ресурс]. – Режим доступа: \www/ URL: <http://aosd.net> – 29.04.2013 г. Загл. с экрана.
6. Feature-Oriented Software Development [Электронный ресурс]. – Режим доступа: \www/ URL: <http://fosd.de> – 29.04.2013 г. Загл. с экрана.
7. Context-Oriented Software Development [Электронный ресурс]. – Режим доступа: \www/ URL: <http://www.hpi.uni-potsdam.de/hirschfeld/cop> – 29.04.2013 г. Загл. с экрана.
8. Ткачук Н. В., Нагорный К. А. Об одном подходе к оценке эффективности применения пост объектно-ориентированных технологий при сопровождении программных систем [Текст] / Ткачук Н.В., Нагорный К.А. // Проблемы программирования (Problems in Programming. Scientific Journal). ISSN 1727 – 4907 – К.: НАН України. – 2010. – № 2-3 (спец. выпуск). – С. 252-260.
9. Hirschfeld Robert et. al. Context-oriented Programming [Текст] / Robert Hirschfeld, Pascal Costanza, Oscar Nierstrasz // Journal of Object Technology. – 2008. vol. 7, no. 3. pp. 125 – 151.
10. Malte Appeltauer. Improving the Development of Context-dependent Java Applications with ContextJ [Текст] / Malte Appeltauer, Robert Hirschfeld, Hidehiko Masuhara // Proceedings of International Workshop on Context-Oriented Programming on ECOOP'09, July 6 – 10, 2009, Gnova, Italy.
11. Sven Apel. How AspectJ is Used: An Analysis of Eleven AspectJ Programs [Текст] / Sven Apel // Journal of Object Technology. – 2010. vol. 9, no. 1. pp. 117–142.
12. Adam Przybyłek. Where the Truth Lies: AOP and Its Impact on Software Modularity [Текст] / Adam Przybyłek // FASE 2011, LNCS 6603. – 2011. pp. 447–461.
13. Sandro Schulze et. al. Code Clones in Feature-Oriented Software Product Lines [Текст] / Sandro Schulze, Sven Apel, Christian Kastner // Proceedings of the GPCE'2010, October 10 – 13, 2010, Eindhoven, The Netherlands. – pp. 103 – 112.
14. Eduardo M. L. Figueiredo. Concern-Oriented Heuristic Assessment of Design Stability [Текст] : PhD thesis / Eduardo M. L. Figueiredo. – Lancaster University, UK., 2009. –237 p.

15. Eaddy Marc. Identifying, Assigning, and Quantifying Crosscutting Concerns [Текст] / Marc Eaddy, Alfred Aho, Gail C. Murphy // Proceedings of the First International Workshop on Assessment of Contemporary Modularization Techniques on the OOPSLA'07, 2007.
16. Eaddy Marc et. al. Do Crosscutting Concerns Cause Defects? [Текст] / Marc Eaddy et. al. // IEEE Transactions on Software Engineering / – 2008/ / vol. 34, no. 4, pp.497-515.
17. Нагорний, К. А. Архітектурні моделі та метрики оцінювання складності застосування пост об'єктно-орієнтованих технологій розробки програмних систем [Текст] / М. М. Литвинчук, К. А. Нагорний, М. В. Ткачук // Вісник ХНУ ім. В.Н. Каразіна, Серія «Математичне моделювання. Інформаційні технології. Автоматизовані системи управління». - 2012. - № 1015. - С.234-245.
18. Garlan David. ACME: An Architecture Description Interchange Language [Текст] / David Garlan, Robert Monroe, David Wile. // Proceedings of CASCON '97, November, 1997, Toronto, Canada. – pp.169-183.
19. Леоненков А.В. Нечеткое моделирование в среде MATLAB и fuzzyTECH [Текст] / Леоненков А.В.– СПб.: БЧВ-Петербург, 2005. – 736с.

У статті проведено дослідження адаптації інтерактивних сервісів порталу. Виявлено показники якості адаптації, елементи зовнішнього середовища для адаптуючого алгоритму, критерії ефективності адаптації. Запропоновано адаптуючий алгоритм, що змінює інтерактивні сервіси порталу, використовуючи дані діяльності окремого користувача. Адаптуючий алгоритм реалізований на мові Java і впроваджений в портал автовласників

Ключові слова: адаптивні алгоритми, адаптивні системи, інтерактивний сервіс, критерії ефективності

В статье проведено исследование адаптации интерактивных сервисов портала. Выявлены показатели качества адаптации, элементы внешней среды для адаптирующего алгоритма, критерии эффективности адаптации. Предложен адаптирующий алгоритм, изменяющий интерактивные сервисы портала, основываясь на деятельности конкретного пользователя. Адаптирующий алгоритм реализован на языке Java и внедрен в портал автовладельцев

Ключевые слова: адаптивные алгоритмы, адаптивные системы, интерактивный сервис, критерии эффективности

УДК 004.051

ИССЛЕДОВАНИЕ АДАПТАЦИИ ИНТЕРАКТИВНЫХ СЕРВИСОВ ПОРТАЛА АВТОВЛАДЕЛЬЦЕВ НА ОСНОВЕ АДАПТИВНЫХ АЛГОРИТМОВ

Т. Б. Шатовская

Кандидат технических наук, доцент*
E-mail: shatovska@gmail.com

Д. С. Негурица

Аспирант*

E-mail: neguritsa@gmail.com

*Кафедра программной инженерии
Харьковский национальный университет
радиоэлектроники
пр. Ленина, 14, г. Харьков, Украина, 61166

1. Введение

Современные информационные порталы автовладельцев предоставляют конечному пользователю огромное количество сервисов для оценки текущего состояния двигателей, трансмиссий, ходовой части, выполнения планового и внепланового обслуживания, бронирование времени на выполнение сервисных работ, покупка и заказ запчастей и т.д. Современное бурное развитие интернета и мобильных технологий позволяет автовладельцу получить доступ к интерактивному сервису, используя личное мобильное устройство или персональный компьютер. С учетом тенденций развития предоставляемых услуг на текущем

рынке, набор сервисов пополняется, чтобы привлечь новых потенциальных клиентов и удержать существующих.

С увеличением сервисов усложняется структура портала и следовательно уменьшается эффективность его использования. Конечный пользователь будет тратить больше времени на поиск и переход на требуемый ему сервис, что ухудшит его ценность. При достаточно сложных переходах возможно не только не привлечение новых клиентов, а и потеря существующих. Одним из явных решений данной проблемы является использование адаптивных алгоритмов. В отличие от технических систем, где адаптация возможна на основании различных технических решений средствами