

УДК 004.412:004.415.2

Особенности энергетических метрик UML диаграмм

В. М. Годунко, В. О. Мищенко, А. В. Пасека

*ИП Годунко В. М., Ростов-на-Дону, Россия,**Харьковский национальный университет им. В. Н. Каразина, Украина,**Харьковский национальный университет им. В. Н. Каразина, Украина*

В проектировании широко используются модели, разработанные средствами UML. Исследования сложности и стиля для таких графических языков имеют короткую историю. В данной работе изучены вопросы адекватности естественного варианта определения энергетических метрик для UML. Новизна связана с подсчётом примитивных характеристик (примитивов). Когда энергетические метрики вводились для языков программирования, например, Ада или Си, в определениях примитивов можно было исходить из традиции научных метрик Холстеда. Для графических языков это невозможно, и в работе выдвинуты определённые критерии адекватности. Доказана независимость энергетических метрик диаграмм классов от некоторых ранее известных метрик.

Ключевые слова: проектирование программных систем, UML, энергетические метрики, научные метрики Холстеда, графические языки, диаграмма классов.

У проектуванні широко використовуються моделі, розроблені засобами UML. Дослідження складності і стилю для таких графічних мов мають коротку історію. У даній роботі вивчені питання адекватності природного варіанту визначення енергетичних метрик для UML. Новизна пов'язана з підрахунком примітивних характеристик (примітивів). При введенні енергетичних метрик для мов програмування, скажімо, Ада чи Сі, можна було визначати примітиви на основі досвіду наукових метрик Холстеда. Для графічних мов це неможливо, і у роботі висунуто певні критерії адекватності. Доведено незалежність енергетичних метрик діаграм класів від деяких раніше відомих метрик.

Ключові слова: проектування програмних систем, UML, енергетичні метрики, наукові метрики Холстеда, графічні мови, діаграма класів.

Models developed by means of UML are widely used in the design. However, studies of complexity and style of graphical languages like UML have a short history. In this paper we study the adequacy of the natural version of the definition of energy metrics for UML. The question boils down to the calculation of primitive characteristics (primitives). When energy metrics were introduced to programming languages, such as Ada or C, the definitions of primitives could come from a tradition of scientific Halstead metrics. That is not possible for a graphic languages, and we are introduced certain criteria of adequacy. The independence of energy metrics of class diagrams from some previously known metrics is proved.

Key words: software design, UML, energy metrics, Halstead's software science metrics, graphic languages, class diagrams.

Введение

Разработка наукоёмких моделирующих программных систем, которые становятся всё более востребованными, является нетривиальной задачей и требует современных подходов к подготовке, испытаниям и использованию программного кода. Один из наиболее перспективных подходов к созданию кода обсуждаемых систем опирается на парадигму последовательного преобразования моделей [1]. На каждом этапе такого преобразования

необходимо принимать во внимание объективные показатели сложности и трудоёмкости требований и кодов. В частности, применяя генераторы кода, полезно каждый раз убеждаться в том, что разработка инструкций такому генератору намного дешевле, надёжнее или быстрее, чем непосредственная разработка требуемого кода [2]. На этапе проектирования сегодня широко используются модели, разработанные средствами UML. В отличие от текстов, подготовленных на универсальных языках программирования, для которых вопросы сложности и стиля исследуются около полувека, для языков, ориентированных на графическое представление требований, подобные публикации появились относительно недавно. Их актуальность возрастает по мере расширения сферы применений языка UML. Наиболее популярны формальные оценки для диаграмм классов UML моделей [3].

В этом сообщении рассмотрены только диаграммы классов и диаграммы компонент, в техническом плане примыкающих к классам. Целью работы была верификация определений метрик, разработанных в соответствии с принципами энергетического анализа программ и сравнение этих метрик с существующими.

1. Постановка задачи

Наиболее популярные, как нам кажется, метрики, которые принято ассоциировать с UML диаграммами, возникли раньше в форме ООП метрик и нацелены на отражение качества тех сущностей (например, классов), которые показаны на этих диаграммах [3]. Де-факто их рассматривали и как метрики качества самих диаграмм. Однако в данной работе нас интересует качество диаграмм UML в первую очередь. Ставится задача испытания одного естественного варианта определения энергетических метрик для UML. Опыт определения таких метрик для языков программирования [3], продолжавший традиции научных метрик Холстеда, здесь непосредственно не применим.

2. Некоторые известные метрики качества диаграмм классов

С. Чидамбер и К. Кемерер (1994) предложили шесть проектных метрик, для классов, из которых 3 подходят для диаграмм классов UML [3]:

- *Взвешенные методы на класс* (Weighted Methods Per Class - WMC): подробности

$$WMC = \sum_{i=1}^n C_i, \quad (1)$$

где C_i - «сложность i -го метода» из всех n методов $M_1 \dots M_n$, и мы здесь, следуя прецедентам «упрощения» [3], считаем её равной 1, то есть, $WMC = n$.

- *Высота дерева наследования* (Depth of Inheritance Tree - DIT) – определяется как длина максимального пути от данного класса до корневого класса в иерархии классов.

- *Количество детей* (Number of children - NOC) – это количество непосредственных наследников класса.

Последние две метрики являются характеристиками дерева наследования, предполагая его показанным на диаграмме. В целом метрики Чидамбера и Кемерера предназначены для оценки сложности проекта в отношении его

влияния на такие характеристики качества продукции, как пригодность для обслуживания и возможность многократного использования [4].

Метрики Ли и Генри, [3], учитывают внутреннюю информацию о классах:

- *Коцентрация абстракций* (data abstraction coupling - DAC) – это количество атрибутов класса, «типом которых является другой класс».
- *Количество локальных методов* (number of local methods - NOM).
- *Число описаний в классах* (number of properties - SIZE2) .

Все эти метрики прошли валидационные процедуры [3], однако не выяснялось, полна ли их система для прогноза качества проектов. Другими словами: что можно сказать о независимости от них других полезных метрик?

3. Энергетические метрики диаграмм UML

Энергетические метрики $W, A, E, Q(q)$ были созданы взамен общеизвестных научных метрик М.Холстеда в качестве характеристик совершенства процесса кодирования и надёжности продукции [5]. Описаны принципы и схемы их точного определения по исходным текстам программ. Всё реализовано для языка Ада и требует уточнения специфических деталей для других промышленных языков программирования. В данной работе речь идёт о реализации этого подхода в отношении «программ», которые имеют графическую форму представления.

В таком случае не очевидны способы формирования схемы программной системы: методы выделения её модулей, блоков и групп блоков. Мы даём такую естественную интерпретацию: *модуль* – оформленная диаграмма, *группа* – её связный подграф, *блок* – прямоугольный классификатор класса.

Число формальных параметров блока определяется как число отношений, в которых участвует класс, классификатор которого определяет этот блок. Следуя наиболее подходящему в энергетическом анализе из определений Холстеда [6],

$$\eta^* = 2 + \eta_2^* , \quad (2)$$

где η^* - потенциальный словарь блока;

η_2^* - число формальных параметров блока.

Программные символы определим как простые элементы UML. Отсюда подсчитываются примитивы длина и словарь.

На основании определения примитивов возможны определения простейших метрик, восходящие к М. Холстеду [5,6]:

$$V = V(M) = V(N, \eta) = N \log_2 \eta \quad (\text{объём модуля или блока}) \quad (3)$$

$$V^* = V(\eta^*, \eta^*) \quad (\text{потенциальный объём блока}) \quad (4)$$

$$D = V / V^* \quad (\text{трудность модуля}) , \quad (5)$$

где N и η - длина и словарь модуля M , η^* - потенциальный словарь блока, V^* – потенциальный объём модуля - сумма таких величин по всем группам, которые подсчитываются в зависимости от статуса группы. Если группа – «структурная», то V^* группы - это сумма V^* по блокам. Тогда:

$$V^*(M) = \sum V^*(B_{ji}) \quad (\text{для модуля } M) \text{ который состоит из блоков } B_{ji}), \quad (6)$$

где B_{ji} - i -й блок j -й группы модуля M .

Спецификационная энергия E определяется для групп, а затем - модуля:

$$E_j = V_j^* / \lambda^2, \quad E(M) = \sum E_j \quad (7)$$

где λ - нормирующий константа, зависящая от языка программирования.

Если схема программной системы нетривиальна, то энергетический анализ требует оценки метрики *объём разработки*, которая для модуля имеет вид:

$$W(M) = \sum V(M_i) + V(N, \eta') \quad (8)$$

где M_i - другие модули системы, от которых зависит данный модуль;

$V(M)$ - холстедовский объём модуля M ;

η' - полный словарь η модуля M , который определяется так, чтоб учесть контекст и порядок разработки (имеющиеся в [4] подробности здесь опустим).

Отметим, что энергетические метрики, в отличие от вышеупомянутых метрик UML/ОПП, вводятся как метрики диаграмм с любым числом классов или систем таких диаграмм. Сравнение с предшествующими метриками раздела 2 осмысленно только для диаграмм, поясняющих один класс.

4. Независимость энергетических метрик

Замечание 1. Пусть имеется две системы метрик M_1, \dots, M_k и N_1, \dots, N_l . Предположим, что метрика M_i зависит от метрик N_1, \dots, N_l . Это означает, что

$$M_i = f(N_1, N_2, \dots, N_l) \quad (9)$$

Тогда для любых двух диаграмм D_1 и D_2 должно быть

$$(\forall j = 1..l : N_j(D_1) = N_j(D_2)) \Rightarrow M_i(D_1) = M_i(D_2), \quad (10)$$

Если же удаётся привести пример таких двух диаграмм, что при некотором j

$$(\forall j = 1..l : N_j(D_1) = N_j(D_2)) \& M_i(D_1) \neq M_i(D_2), \quad (11)$$

то метрика M_i не может зависеть от метрик N_1, \dots, N_l . Если такие примеры находятся для всех M_i , то система метрик M_1, \dots, M_k не зависит от N_1, \dots, N_l .

Лемма о независимости. Базовые энергетические метрики V и V^* независимы от системы 3-х метрик Чидамбера и Кемерера, подходящих для диаграмм классов UML, плюс метрика числа описаний в классах от Ли и Генри.

Для доказательства достаточно подсчитать все указанные метрики диаграмм а) (поясняет класс A) и б) (поясняет B) на рис. 1 и воспользоваться замечанием 1.

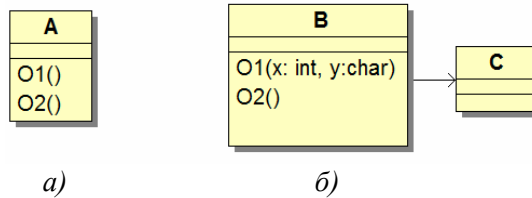


Рис. 1. Диаграммы, используемые в доказательстве леммы о независимости

Замечание 2. Утверждение леммы справедливо для и больших систем ранее известных метрик. Например, можно было бы добавить остальные метрики Ли и Генри. Мы воздержались от этого в виду расплывчатости их определений в [3,4].

5. Критерий адекватности, основанный на уравнении длины

Для верификации правил подсчёта программных символов (ПСим-лы) на UML диаграммах их подсчитывали две группы студентов. Одни действовали на основе определения предыдущего раздела, а другие имели полномочия составить и использовать стратегию подсчёта на собственное понимание. Использовались 20 UML диаграмм взятых из [7]. Оценка регрессионной зависимости длины модуля N от его словаря n для первой группы лучше следует уравнению длины программы Холстеда [5,6] (рис 2.), которое является, по сути, единственным объективным критерием адекватности определения ПСим.

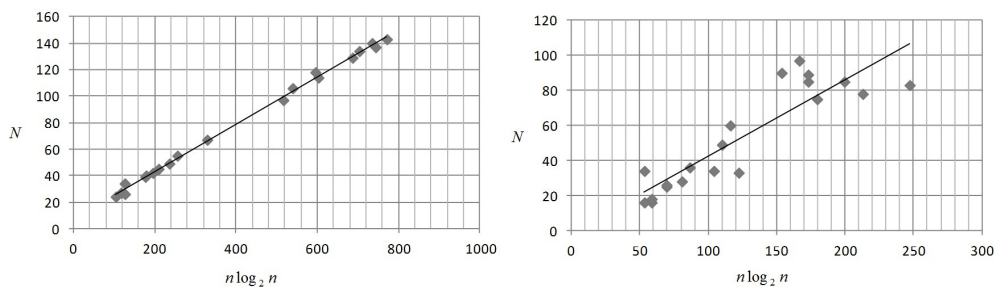


Рис. 2. Линеаризованная регрессия длины N на словарь n при разных определениях ПСим

6. Сбалансированность энергии и работы для диаграмм

Наряду с (7), спецификационная энергия модуля, состоящего из многих не сгруппированных блоков (типично для UML диаграмм), полагается равной [6]:

$$E_f = \begin{cases} (V^*)^3 \cdot \lambda^{-2} & \leftarrow n \leq 4 \\ (n/9 + 0.5) \cdot \left(V^* / (n/9 + 0.5) \right)^3 \cdot \lambda^{-2} & \leftarrow n \geq 5 \end{cases}, \quad (12)$$

где n – количество блоков в модуле, V^* – сумма потенциальных объёмов данных n блоков, λ – уровень языка программирования (в наших расчетах $\lambda = 1$).

О расхождении между спецификационной энергией программы и ее работой A свидетельствуют метрики (q) Q - (нормированного) информационного тепла:

$$A = D \cdot V, \quad Q = E - A, \quad q = Q / \max(E, A). \quad (13)$$

Проведен вычислительный эксперимент по выяснению распределения метрики q на реальных выборках диаграмм, взятых из [7,8].

Результаты рассматривались в плане верификации принятого метода оценки спецификационной энергии. Критерий – отражение теоретически предсказываемой тенденции к колебаниям между значениями E и A при соблюдении в среднем «нулевого» баланса по большим выборкам и по процессу разработки (к его концу). Набранная статистика (рис. 3) пока недостаточна для

суждений о таком балансе на каком-то уровне достоверности. Однако она не противоречит явно гипотезе о таком балансе, поскольку демонстрирует, что знак этой величины может оказываться разным. Вполне возможно, что для данных предметных областей существует прямая регрессии A на E , которая отлична от биссектрисы квадранта и служит отражением специфики этих областей [5].

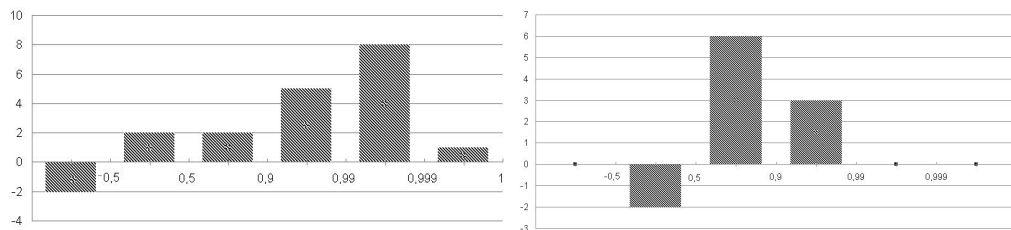


Рис. 3. Гистограммы распределения величины q для выборок из двух предметных областей: а) обучающие демонстрации [7]; б) параллельное программирование [8]

7. Распространение определений на диаграммы компонент

Все определения примитивов, принятые для диаграмм классов, можно сохранить и для диаграмм компонент. Применим наши критерии адекватности. В качестве основы выборки использовались примеры из [8], которые пришлось дополнить другими, взятыми из Интернета, для увеличения числа хотя бы до 16.

Соответствие уравнению длины (рис. 4) немного ухудшилось по сравнению с

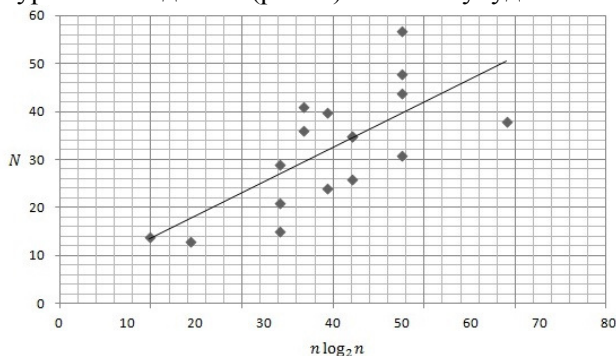


Рис. 4. Линеаризованная регрессия длины N на словарь n для диаграмм компонент

вариантом б) рис. 1, но считать его нарушенным нет оснований. Простое объяснение: примеры диаграмм компонент отличаются большим разнообразием, чем для классов, и связь длины программы с её словарём более стохастична.

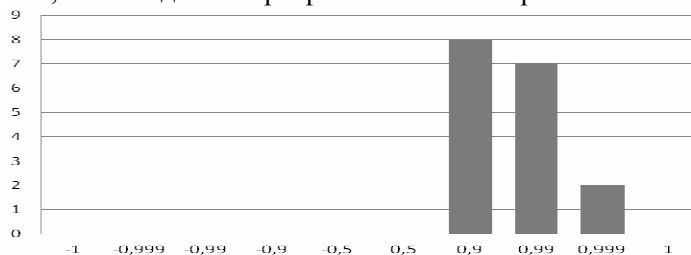


Рис. 5. Дисбаланс между энергией и работой по выборке из диаграмм компонент

Нарушен баланс в среднем между спецификационной энергией и работой (рис. 5). Если это так и для других выборок, то определение спецификационной энергии для диаграмм компонент придётся усовершенствовать.

Заклучение

Разработан метод верификации определения энергетических метрик для UML диаграмм, что позволяет использовать известные возможности таких метрик для оценки качества указанных диаграмм в проектах программных систем. Верификация определений основана на согласии с уравнением длины программы Холстеда и согласии с тезисом о балансе в среднем по выборкам между спецификационной энергией и работой кодирования. Применительно к естественным определениям для диаграмм классов данный метод показал удовлетворительность гипотезы их адекватности, а в приложении к диаграммам компонент – необходимость дополнительного исследования.

В дальнейшем можно применить факторный анализ для прояснения проблем определения энергетических метрик для диаграмм компонент. Полезно автоматизировать оценку метрик диаграмм классов. Для этого можно использовать XMI модель для UML [9] и привлечь инструменты Matreshka [10].

ЛИТЕРАТУРА

1. OMG's MetaObject Facility [Электронный ресурс]. – Режим доступа: <http://www.omg.org/mof/>.
2. Годунко В.М., Мищенко В.О., Резник М.М., Штефан Д.В. Динамическая генерация HTML страниц для Ada web серверов // Радиоэлектронные и компьютерные системы. – 2012. – № 57. – С. 225 – 229.
3. Calero C., Genero M., Piattini M. A survey of metrics for UML class diagrams // Journal of Object Technology. – 2005. – 9. – P. 59-92.
4. Орлов С.А. Технологии разработки программного обеспечения: [учеб. пособие] / С. А. Орлов – СПб.: Питер, 2003. – 480 с.
5. Мищенко В.О. Энергетический анализ программного обеспечения с примерами реализации для Ада-программ. – Х.: ХНУ имени В.Н. Каразина, 2007. – 128 с.
6. Мищенко В.О. CASE-оценка критических программных систем. Том 1. Оценка качества / Мищенко В.О., Поморова О. В., Говорущенко Т. А. / под ред. Харченко В. С. – : Х: Нац. аэрокосмический ун-т «Харьк. авиац. ин-т», 2012. – 201 с.
7. ISO/IEC 19505-1 [Электрон. ресурс]. – Режим доступа: <http://www.omg.org/spec/UML/ISO/19505-1/PDF/>.
8. A. Burns, A. Wellings. Concurrent and real-time programming in Ada. // Cambridge University Press. – 2007. – P. 461.
9. OMG MOF 2 XMI Mapping Specification [Электронный ресурс]. – Режим доступа: <http://www.omg.org/spec/xmi/2.4.1/pdf>. – 09.08.2011.
10. Матрешка: Преобразование Ada в UML - Ada-Ru Community -16 Апр. 2012 [Электронный ресурс]. – Режим доступа: <http://ada-ru.qtada.com/archives/445>.