

УДК 004.41:519.6+004.412:004.05

В.О. МИЩЕНКО

*Харьковский национальный университет им. В.Н. Каразина, Украина***МЕТРИКИ РАСШИРЕНИЙ ПРОГРАММ ВЫЧИСЛИТЕЛЬНОГО НАЗНАЧЕНИЯ**

Для небольших программных систем компьютерного моделирования физических процессов автор разработал метод проектирования, позволяющий поддерживать качество продукции в условиях экономии на полноте документирования требований и принятых решений. Такая экономия, вынуждаемая дефицитом ресурсов при разработке оригинального программного обеспечения (ПО) данного класса, ранее обычно компенсировалась только комментариями или ссылками на научную литературу. В нашем подходе роль спецификации играет система ссылок в исходном тексте программы на предварительно структурированные тексты по теории используемых методов. Созданы метрики для моделей качества ПО, разрабатываемого на основе такого подхода.

Ключевые слова: компьютерное моделирование, спецификации, разработка, схема программы, программные метрики, модель качества.

Проблема и предыдущие работы

Программами вычислительного назначения будем называть такие программные системы, которые позволяют в порядке опытно-конструкторских работ или в исследовательских целях проводить компьютерное моделирование сложных физических процессов на основе наукоемких численных методов.

Данная работа исходит из опыта более узкой предметной области, в которой рассматриваемая проблема представляется особенно актуальной. Это математическое моделирование процессов распространения и рассеяния волн на основе методов дискретных особенностей (МДО) [1-3]. Приложения связаны с приемом или генерацией электромагнитных волн (или пучков) или с их передачей с помощью волноведущих конструкций, а также с некоторыми задачами акустики, механики твердого тела [1]. Быстрое развитие теории и алгоритмов МДО весьма редко сопровождается появлением соответствующих отчуждаемых систем компьютерного моделирования. Построенные для апробации новых методов программы часто не предъявляются или разработаны как код плюс общие пояснения. Стандартная оценка их качества крайне затруднена [4].

Причина этого в том, что реализацию программных систем моделирования на основе новых подходов обычно выполняют авторы этих подходов. Те, экономя свое время, фиксируют только программный код, а, используя математические пакеты, не делают и того. Если бы при проектировании и кодировании тщательно велась документация, то было бы выше качество программ, проще изменения, но трудоемкость процесса была бы намного

большей.

Ситуацию можно изменить на основе подхода автора [5]. Вместо трудоемкого полного документирования небольшого проекта предложено снабжать программный код адресациями к нужным разделам научных (так их назовем) текстов (книг, отчетов, статей) при условии, что этим текстам придана определенная семантическая структура (см. [6]), подобная той, которой обладают программы.

1. Актуальность задачи, цель, постановка

Определение 1. *Научным обоснованием программы назовем размеченную согласно [6] систему текстов, нужных для ее разработки. Расширение программы – это она вместе с ее научным обоснованием и точными ссылками на него – адресацией.*

Весьма актуальна разработка средств контроля качества программ создаваемых методом [5]. Пусть, например, научных материалов, отобранных и структурированных в качестве научного обоснования, для понимания всей программы не достаточно. Что служило бы индикатором такой ситуации?

Целью данной работы является построение модели и надежно вычисляемых метрик качества для разработки небольших наукоемких вычислительных программ методом расширений [5], имея в виду актуальное применение к программным системам моделирования сложных дифракционных процессов.

Достижение этой цели требует такой постановки задачи исследования. Дать более цельное, чем в [5] (где рассмотрен пример) описание средств и схемы метода расширений, определить метрики, обосновать их выгоды для управления качеством.

вом ПО.

2. Метод расширения программ

Тексты научного и технического содержания, в которых имеются описания и обоснования определенных структур данных, алгоритмов, формульных выражений и т.п., превращаются в материалы для научных обоснований с помощью семантической структуризации [6, 8]. При этом *создаются размещаемые модули*, которые, грубо говоря, представляют собой *аннотации* либо *реализации* – тексты (или их части), размеченные с помощью специального языка. Рассмотрим смысл этой структуризации для целей подхода [5] к построению ПО.

Материалами для научных обоснований служат опубликованные статьи, книги и т.п., а также (при дефиците нужных публикаций) документы, специально созданные при подготовке ПО. Если материал невелик (статья), то должен быть представлен двумя размещаемыми модулями, из которых один – аннотация, а другой – реализация этого материала. При этом материал получает форму: решение (если в расширенных программах ожидаются ссылки только на материал в целом) или контекст (если ссылки будут ко внутреннему содержанию) или прототип (метод, по которому может создаваться решение, или схема, по которой может создаваться контекст). Внутри более сложных по структуре материалов – контекстов можно выделять внутренние решения и контексты. Чем крупнее и важнее материал, тем внутренних контекстов может быть больше. Если материал велик (книга), то должен быть разбит на разделы и представляется модулем аннотации контекста (как правило, без внутренних решений и контекстов) плюс система аннотаций и реализаций дочерних контекстов, которые представляют разделы (подразделы), трактуемые подобно статьям.

Простой контекст стандартно выглядит так:

```
//контекст// Название //это//
Кратко //:// резюме //есть//
  Идентификация Аннотация //://
//результат// Название_первого //://
...
//результат// Название_последнего //://
//все// Название //://
```

Данные курсивом понятия следует замещать текстом по смыслу. Служебные слова и символы окружают «//». Обычным шрифтом – слова, подбираемые по смыслу. Конструкция «Кратко //:// резюме //есть// ... //://» вводит *справочную информацию*, причем (см. [8], правила 36-38) текст до «//://» это – определяемый *элемент*, после – *термин* из предопределенного списка, а после слова «//есть//» – вы-

ражение (точнее сказать, «наполнение»). Например, при желании подчеркнуть, что использована цитата из данного материала, можно выбрать варианты:

```
Цитируемое //:// резюме //есть// ... «...»//:// ,
О содержании //:// цитата //есть// ... «...» //:// .
```

Конструкция «//результат// ... //://» – простейший пример аннотации решения, данного как *результат*. В общем случае требуется *пояснение решения*:

```
//результат// Название
//(// 1 //исходя из// термин //:// Фактов и т.п. //://
  2 //получаем// термин //:// Пояснение.//)// //://,
```

где секции пояснений можно повторять, а их номера не указывать.

Всякий размещаемый модуль может начинаться с предписания контекста, имеющего вид:

```
//в контексте// Название материала //н// ...
Название материала //://
//в терминах// Название контекста //н// ...
  Название контекста //://,
```

где название материала подразумевает такой уже помещенный в базу проекта модуль аннотации, от которого зависит данный модуль. Те из названных материалов, которые имеют форму контекста, следует упоминать после слова «//в терминах//».

Теперь о реализациях (размещаемых модулей или их внутренних контекстов и решений). Они включают тексты материалов, куда добавлены в соответствии с аннотацией метки, определяющие соответствие участков текста ранее определенным конструкциям из аннотаций. Вот, например, шаблон реализации небольшого контекста:

```
//контекст// //подробно// Название //это//
Текст, сопровождаемый метками
//результат// Название_первого //это//
//начало// Текст, сопровождаемый метками
//все// Название_первого //://
...
//результат// Название_последнего //это//
//начало//...
//все// Название_последнего //://
Текст, сопровождаемый метками
//начало// Текст, сопровождаемый метками
//все// Название //://.
```

Характерной меткой в модуле реализации является адресация используемого результата:

```
Название //(//
  1 //:// Соответствующее пояснение //н//
  2 //:// Соответствующее пояснение //://
  ... .//)// //://,
```

где номер дает соответствие между пояснением использования и ранее аннотированным пояснением

решения. Еще можно адресоваться к определениям и элементам справочной информации (см. [8]).

Средства поддержки разработки трактуют эти адресации как вызовы процедур, использование типов или объектов из дополнительных модулей (на языке программирования), *сопряженных* модулям научного обоснования. Как строить сопряженные модули на языке Ада см. [7], а на других языках – аналогично. Сопряженные модули не имеют алгоритмического эффекта, и, считай, не берут лишней памяти. Но они позволяют, например, автоматизировать оценку определяемых ниже метрик.

Простейшие способы документирования программ, например, с помощью комментариев в исходных текстах, тоже можно рассматривать как расширения. Но наш метод состоит в планомерном модульном расширении программ. Вот его схема.

На основе знания общих требований к программе сформировать список решаемых ею задач и функций, обеспечивающих эти задачи (для программ, реализующих моделирование дифракции, типовой список см. в [9]). Далее, ориентируясь на задачи, построить эскиз схемы программной системы (СПС, см. [7, 10]). Придать модулям внутренне блоки соответственно требуемым функциям. По СПС генерировать каркас будущей программы [10]. Заполнять его кодом. При реализации структур данных и алгоритмов, определенных в научном обосновании, предварительно отмечать их использование. Для этого записывать (комментарием) адресации используемых результатов, определений, элементов, как это делается в модулях реализации материалов.

3. Метрический контроль расширений

Определим на расширениях программ такие функции со значениями в интервале (0; 1), которые можно интерпретировать как меры желательных свойств этих расширений (метрики), причем 0 – свойство отсутствует, а 1 – присутствует полностью. Упомянутые далее работа программирования и спецификационная энергия [7] суть обобщения для сложных программ хорошо известной «работы в программировании» Холстеда и ее оценки посредством величины уровня языка программирования.

Определение 2. Энергетической оценкой научного обоснования программы (ЭОНОП) называется зависящая от языка реализации спецификационная энергия (см. [7]) соответствующей системы сопряженных модулей этого научного обоснования.

Определение 3. Метрика «Согласованность энергий программы и ее обоснования» (СЭПО) *предназначена* для оценивания расширений вычислительных программ при условии их разработки

методом предыдущего раздела. Эта метрика оценивает различие сложности структуры научного обоснования и программы.

Примитивы:

$E(P)$ – спецификационная энергия программы P ,
 $E_s(P)$ – ЭОНОП относительно языка P . (1)

Формула применения:

$$X = 1 - (2 \cdot |E(P) - E_s(P)|)^q / (E(P) + E_s(P))^q \quad (2)$$

где по умолчанию $q = 2$; зависимости от языка нет.

Интерпретация: При $X = 0$ научное обоснование отсутствует, а при достаточно близких к 0 значениях оно «свалено» в один неструктурированный модуль, что затрудняет его использование. Значения X , близкие к 1, свидетельствуют о том, что программа и ее научное обоснование имеют близкие показатели структурной сложности, что хорошо – это необходимый признак хорошей согласованности. Если из двух расширений одной программы у одного из них СЭПО явно больше, чем у другого, то много шансов, что работать с ним удобнее.

Идея: Оценивать числовой атрибут метрики не менее трех раз (вначале, по ходу и по окончании кодирования программы). При тщательной и опережающей (по отношению к кодированию) подготовке научного обоснования значения X будут расти (или не убывать) по направлению к 1. При существенных отклонениях от монотонности (или, если последняя оценка $X \ll 1$) то тщательность работы под вопросом.

Выгода: Если между программой и ее научным обоснованием нет примерного соответствия по интегральному показателю сложности, рассчитанному по элементам структуры, то, вероятно, допущены просчеты. Метрика дает шанс их своевременно устранить, сигнализируя о несоответствии.

Нам потребуется понятие формальных символов (tokens) в смысле Холстеда для языка «научной прозы» (см. [11]). Число таких символов приравнивается числу N слов в тексте, а размер словаря полагается

$$n = k \cdot m \quad (3)$$

где m – число разных использованных слов; k – поправка (умолчание $k = 0.4$, см. [11], с. 112).

Не сложно подсчитать символы в таблицах и формулах, а для всех типов рисунков иметь правила.

Определение 4. Назовем работой над текстом научного обоснования величину работы программирования в смысле ([7], с. 67-74) для соответствующей системы размещаемых модулей, длины и словари которых оцениваются, как выше, а потенциальные объемы – по сопряженным модулям.

Определение 5. Метрика «Степень подробности обоснования» (СПО) применяется при том же

условии, что и метрика в определении 4. Она *предназначена* для оценки степени подробности изложения в научном обосновании программы.

Примитивы:

$A(P)$ – работа программирования программы P ,
 $A_s(P)$ – работа над текстом обоснования P . (4)

Формула:

$$X = \begin{cases} \frac{x}{2q} & \Leftarrow x < q \\ 1 - \frac{q}{2x} & \Leftarrow x \geq q \end{cases} \quad \left(x = \frac{A_s(P)}{A(P)} \right), \quad (5)$$

где по умолчанию $q = 2$; в общем случае

$$x = q \Rightarrow X = 0.5, \quad x = 5q \Rightarrow X = 0.9. \quad (6)$$

Интерпретация: Достаточно малые значения числового атрибута соответствуют ситуации, когда объем словесно-формульных описаний вычислительных методов мал в сравнении с их реализацией, что не соответствует смыслу «обоснований» (они фиктивны или фрагментарны), а достаточно близкие к 1 значения соответствуют большим объемам словесных описаний, что подозрений не вызывает. В промежутке, чем больше значение данной метрики, тем относительно подробнее обоснование.

Идея: Проводить оценку после того, как тестируемое значение СЭПО стабилизировалось вблизи 1. В таком случае потенциальные объемы в выражениях работы для программы и обоснования в среднем примерно равны, и отношение работ в формуле данной метрики примерно соответствует среднему отношению объемов модулей (программы и ее обоснования). Все в порядке, если это отношение мало (то есть, $X \approx 1$). Обратная ситуация приемлема только, если обоснование само написано на формальном языке. Выбор $q = 2$ учитывает и такую возможность.

Выгода: Метрика выявляет дефект обоснования, при котором система модулей аннотации адекватна архитектуре программы, но модули реализации – пустышки или конспективное изложение теории.

Определение 6. Метрика «Охват адресации» (ОА) применяется тогда же, когда метрика СЭПО и *предназначена* для оценки степени охвата адресацией модулей СПС данной программы.

Примитивы:

M – число модулей СПС данной программы,
 M_a – число модулей с адресацией. (7)

Формула:

$$X = \begin{cases} 1 & \Leftarrow q \leq x \\ M_a / (q \cdot M) & \Leftarrow x < q \end{cases}, \quad \left(x = \frac{M_a}{M} \right), \quad (8)$$

где по умолчанию $q = 0.9$.

Интерпретация: $X = 0$ неприемлемо (научное обоснование не используется), а, чем X больше, тем полнее модули охвачены адресацией вплоть

до ее практической полноты при $X = 1$.

Идея: Если известно, что почти вся программа служит вычислительным целям, то малость X означает небрежность ее оформления или отсутствие в ее научном обосновании нужных сведений.

Если наряду с вычислительными, широко представлены не вычислительные функции, их можно сосредоточить в одном модуле, и метрика останется применимой.

Выгода: При кодировании с адресациями к научному обоснованию метрика выявит явные пробелы в адресации, которые можно проанализировать и исправить.

Определение 7. Метрика «Детальность адресаций» (ДА) применима одновременно с СЭПО и *предназначена* для оценки того, насколько в среднем «обеспечены» адресациями элементы детальной структуры программы – блоки СПС.

Примитивы:

M_a – число модулей, в которых есть адресации,

A_i – число адресаций в i -м из этих модулей,

B_i – число внутренних блоков i -го модуля. (9)

Формула:

$$X = \min \left(1, \left(\frac{\sum_{i=1}^{M_a} A_i}{\sum_{i=1}^{M_a} B_i} \right)^q \right), \quad (10)$$

где умалчиваемое значение $q = 2$; при подсчете внутренних блоков модуля нужно учесть все: автономные и сгруппированные подпрограммы, остатки синтаксически оформленных групп, остаточный блок модуля (подробнее, и тем – точнее, см. [7]).

Интерпретация: $X \approx 0$, если и только если модули СПС программы с адресациями имеют сложную внутреннюю структуру, а адресаций в них мало. $X = 1$ выделяет отличную ситуацию, когда каждая единица внутренней структуры модулей с адресациями имеет в среднем не менее одной адресации. $X \approx 1$ приемлемо.

Адресации в программе с большим значением X в целом более детальны, чем в программе с меньшим значением.

Идея: Оценка метрики имеет смысл даже до того, как полностью определится охват модулей адресацией, хотя особо желательна по завершении разработки исходного кода. Оценка ДА особенно важна, если оценка метрики СЭПО около 1.

В этом случае структуры программы и ее научного обоснования согласованы. Тогда в среднем каждый элемент программной структуры обязан найти себе адресацию, и только 1 – приемлемый результат. Иначе либо необходимые адресации вы-

полнены не полностью, либо обоснование, несмотря на хорошую энергетическую согласованность с программой, является недостаточным по смыслу.

Выгода: На всех этапах кодирования сложной программы можно обнаружить неочевидную неполноту адресаций и принять меры.

4. Подмодель качества для расширений

Разработка программных систем методом расширений программ требует включения специальной компоненты в модель качества таких систем. Дело в том, что по замыслу материалы для научных обоснований должны размещаться в базе данных для многократного использования. Тогда разработчик сможет адресоваться в программе ко многим материалам, не размечая и не размещая их сам (разве только какую-то свежую статью). В этом – источник экономии времени на подготовке спецификаций. В этом и угроза качеству, если «чужая» разметка неадекватна нуждам программы (например, большая статья могла быть ранее оформлена не как контекст с результатами, а как результат). Такое несоответствие случайно с точки зрения данного разработчика.

Метрики СЭПО и СПО, проверяя соответствия, имеют много шансов обнаружить случайные нестыковки нужд разработчика с целями проводившейся ранее структуризации. После исправления (обычно – углубления) структуризации, метрики ОА и ДА становятся датчиками качества самой разработки.

Таблица 1

Энергетические оценки обоснования по его СПС

Блок	Форма	Пояснений	V^*	$\lambda^2 E_s$
Вк.9.А	результ.	2	16	4096
Вк.9.В	результ.	2		
Вк-rest	контекст	0	2	8
Вк.10	результ.	2	8	512
Рр.А	результ.	2	16	4096
Рр.В	результ.	2		
Рр-rest	контекст	0	2	8

Пример применения метрик. Научное обоснование проекта: книга Вк, главы 9-10, и статья Рр. Последствия их структуризации показаны в табл. 1. Требуется разработать программу MN, которая позволит решать задачи: ввода-вывода (Ю), подготовки расчетов (РЕ), расчета (МС), оформления результатов (RV). Отображение задач и функций на архитектуру этой программы характеризует табл. 2. На основе табл. 1-2 и формул (1)-(2) находим:

СЭПО: $E_s = 8710, E = 19843, X = 1 - 0.61 = 0.39$ (11)

Это плохо. Табл. 1 подсказывает, внутренняя структура Вк.10 не проявлена. После переработки ее модуля-аннотации в контекст аналогично Вк.9 имеем:

СЭПО: $E_s = 12312, E = 19843, X = 0.78$. (12)

Таблица 2

Оценка спецификационной энергии программы

Блок	Форма	Парам-в	V^*	$\lambda^2 E$
Ю.Г	проц.	2	16	4096
Ю.Р	проц.	2		
Ю-rest	пакет	2	8	512
РЕ	проц.	3	11.61	1564.8
МС.Ф	проц.	2	16	4096
МС.И	проц.	2		
МС-rest	пакет	4	15.51	3731.1
RV	функ.	2	8	512
MN	проц.	4	15.51	3731.1

Расчет A, A_s (4) помимо N, n и V^* всех модулей требует знания порядка разработки модулей [7]. Считая порядок естественным, мы имеем для СПО:

$A = 1.26 \cdot 10^6, A_s = 23.9 \cdot 10^6, x = 18.9, X = 0.95$. (13)

Один из 5 модулей не зависит от обоснования:

ОА: $X = 4 / (0.9 \cdot 5) = 0.89$. (14)

что приемлемо. По табл.3 и 2 вычисляем:

ДА: $X = 6 / 9 = 0.67$. (15)

Это невысокий результат. Проверка необходимости адресаций по смыслу в Ю-rest, МС.Ф МС-rest, где их нет, привела к добавлению двух, и тогда ДА= 0.89.

Таблица 3

Данные об адресациях к научному обоснованию

Модуль:	Ю	РЕ	МС	MN
Адресаций	2	1	2	1

Замечания. От программиста требуется только *навык* оценивания энергетических метрик. Настройка *шкал* всех метрик – за счет параметра q .

Заключение

Для ранее опубликованного в концептуальной форме метода расширения программ дано цельное описание средств и схемы применения. Указаны преимущества в сравнении с традиционными приемами разработки вычислительных программ для систем компьютерного моделирования сложных дифракционных процессов на основе МДО.

Для вычислительных программ разработаны новые метрики внутреннего качества, которые формируют модель контроля качества для страховки от угроз этому качеству, характерных для метода расширений. Разъяснения интерпретации и идейных основ этих метрик содержат обоснования тех необходимых свойств, которые не требуют более позднего рассмотрения по данным о применении.

Литература

1. Лифанов И.К. Метод сингулярных интегральных уравнений и численный эксперимент (в математической физике, аэродинамике, теории упругости и дифракции) / И.К. Лифанов. – М.: ТОО «Янус», 1995. – 520 с.

2. Гандель Ю.В. Новый подход к решению смешанных краевых задач для уравнений Лапласа и Гельмгольца / Ю.В. Гандель, И.К. Лифанов // Дифференциальные уравнения. – Т. 34. – № 9. – С. 1246-1253.

3. Гандель Ю.В. Парные и гиперсингулярные интегральные уравнения задач дифракции электромагнитных волн на плоских решетках и экранах / Ю.В. Гандель // Труды XI Международного симпозиума МДОЗМФ-2003. – Харьков-Херсон, 2003. – С. 53-58.

4. Гахов А.В. Поиск математической модели при анализе связи между видами качества расчетных программ / А.В. Гахов, В.О. Мищенко // Радиоэлектронные и компьютерные системы. – 2008. – №6. – С. 214-218.

5. Мищенко В.О. Построение программных систем моделирования дифракции на идеально проводящих экранах, лежащих в диэлектрическом полупространстве / В.О. Мищенко // Вісник Харк. нац. ун-ту. –

2004. – № 833. – Вип. 10. – С. 170-184.

6. Мищенко В.О. Семантическая структуризация текстовых данных в базе данных, поддерживающей распределенную разработку программного обеспечения / В.О. Мищенко // Вестник ХГТУ. – 2002. – № 1. – С. 304-307.

7. Мищенко В.О. Энергетический анализ программного обеспечения с примерами реализации для Ада-программ / В.О. Мищенко. – Х.: ХНУ им. В.Н. Каразина, 2007. – 119 с.

8. Мищенко В.О. Прикладной язык семантической структуризации научных текстов как некоторая проекция языка Ада / В.О. Мищенко // Вісник Харк. нац. ун-ту. – 2004. – № 605. – Вип. 3. – С. 90-105.

9. Gahov A.V. The Validation of the Software that Was Developed for Calculations Related to the Design of Antennas / A.V. Gahov, V.O. Mishchenko // Радіоелектронні і комп'ютерні системи. – 2007. – № 6. – Р. 180-185.

10. Теоретические основы, методы и инструментальные средства анализа, разработки и верификации гарантоспособных информационно-управляющих систем для аэрокосмических объектов и комплексов критического применения. В 4-х частях. ГР №. 0106U001071. – Х.: ХАИ, 2008. – 871 с.

11. Холстед М.Х. Начала науки о программах / М.Х. Холстед. – М.: Финансы и статистика, 1981. – 128 с.

Поступила в редакцию 25.02.2009

Рецензент: д-р техн. наук, проф. В.С. Харченко, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков, Украина.

МЕТРИКИ РОЗШИРЕНЬ ПРОГРАМ ОБЧИСЛЮВАЛЬНОГО ПРИЗНАЧЕННЯ

В.О. Мищенко

Для невеликих програмних систем комп'ютерного моделювання фізичних процесів автор розробив метод проектування, що дозволяє підтримувати якість продукції в умовах заощадження на повноті документування вимог та прийнятих рішень. Таке заощадження, що до нього призводить дефіцит ресурсів при розробці оригінального програмного забезпечення (ПЗ) даного класу, було звичайно компенсоване лише коментарями та посиланнями на наукову літературу. У нашому підході роль специфікацій грає система посилань у вихідному тексті програми на попередньо структуровані тексти з теорії методів, що використовуються. У статті створено метрики для моделей якості ПЗ, яке може розроблятися на основі такого підходу.

Ключові слова: комп'ютерне моделювання, специфікації, розробка, схема програми, програмні метрики, модель якості.

MEASURES FOR EXTENSIONS OF PROGRAMS FOR NUMERICAL

V.O. Mishchenko

For small software systems of computer modeling of physical processes the author has developed a designing method that is supporting the product quality in conditions of economy on documentation completeness. This unfortunate economy is explained by resources scarcity of the development of original software of that class. Previously, the economy was usually compensated by only comments or links to a scientific literature. In our approach the role of detailed specifications is played by the system of links to pre-structured texts on the theory of methods used. The metrics have been developed for quality models of the software that is developed using this approach.

Keywords: computer-aided modeling, specifications, design, scheme of program system, program metrics, quality model.

Мищенко Виктор Олегович – канд. физ.-мат. наук, доцент, Харьковский национальный университет им. В.Н. Каразина, Харьков, Украина, e-mail: mischenko@univer.kharkov.ua.