

СОЗДАНИЕ МЕТОДОЛОГИИ ПРОЕКТИРОВАНИЯ CASE-СРЕД НА БАЗЕ СИТУАЦИОННОГО МЕТОДА

Рассмотрены принципы построения методологии MetaCASE для проектирования CASE-сред на основе ситуационного метода. Методология ориентирована на разработку CASE-сред, адаптируемых к ситуации применения. В рамках подхода определен набор базовых концепций инженерии ситуационного метода, используемых в MetaCASE. Описаны основные методологические решения по построению архитектуры и способам адаптации созданных CASE-сред. Определены пути реализации и развития методологии.

Введение

Предметом рассмотрения данной статьи является концептуально новый подход к созданию методологий разработки программных систем (ПС). Существующие методологии характеризуются разнообразием подходов и претензией на универсальность, но зачастую эффективно поддерживают лишь отдельные подпроцессы разработки ПС, что не отвечает требованиям заказчика и проблемным доменам [1]. Поэтому в последние годы проявилась тенденция к созданию гибких методологий, удовлетворяющих конкретным требованиям заказчиков, посредством сборки из компонентов действующих методологий [2].

Компонент методологии – это самодостаточная часть методологии, определяющая способ трансформации одного или нескольких артефактов разработки ПС в целевой артефакт.

Концепции компонента соответствуют части методологии, когерентные в отношении как используемого метода представления процесса разработки (например, объектного или структурного), так в отношении самого объекта – подпроцесса разработки ПС. Метамоделю компонента включает описание отдельного подпроцесса разработки ПС, артефактов разработки (промежуточных продуктов на разных фазах процесса), методики управления подпроцессами и необходимых ресурсов (среды разработки, исполнителей). Данное опи-

сание может быть текстовым или формализованным в зависимости от наличия соответствующих нотаций и механизмов их поддержки в используемом методе разработки.

Главный акцент в инженерии методологии делается на адаптацию создаваемой методологии к условиям применения, базирующуюся на ее интерпретации как сформированной целевой совокупности когерентных частей – компонентов [3]. Реальность такого подхода подтверждается возможностью разбиения существующих методологий на отдельные части. Например каждую из методологий Rational Unified Process (RUP), Object-Oriented Process, Environment и Notation (OPEN) и Process Framework (OPF) можно представить в виде множества компонентов [4,5,6]. Это позволяет проводить поиск и выборку отдельных компонентов, их настройку на конкретные проблемные характеристики с последующим объединением в новую методологию. Такое направление работ активно разрабатывается в проекте Method Tailoring [7], ориентированном на адаптацию обобщенного методологического каркаса к конкретным проектам разработки программных систем.

В данной статье термин «методология» заменяется термином «метод», «компонент» – «фрагментом метода», «инженерия методологии» – «инженерией метода». Термин «методология» используется для систематизированного описания, объяснения и

оценки основных аспектов методики разработки CASE-сред в системе MetaCASE [8].

В статье рассматривается *инженерия ситуационного метода* (ИСМ), которая ориентирована на формализацию инженерии методов, что отражено в ряде работ, посвященных этому направлению [9, 10]. Предлагаемый нами подход представляется как попытка использования ИСМ для итеративного проектирования CASE-сред, осуществляющих методическую и инструментальную поддержку процессов разработки ПС. Методологии такого назначения обычно реализуются системами MetaCASE.

Концепции инструментальной поддержки разработки соответствуют отдельные инструменты (редакторы, компиляторы, отладчики, генераторы программного кода и документации, и т.д.) и инструменты CASE, автоматизирующие, например, методы структурного анализа или проектирования программных артефактов.

Инструмент CASE определяется как компьютерно-ориентированный продукт, предназначенный для поддержки одного или нескольких методов программной инженерии в рамках процесса разработки ПС.

Обычно инструменты CASE реализуют фиксированные методы, и наиболее распространенными среди них являются объектно-ориентированная разработка ПС, структурное программирование Джексона, метод Джордана, прототипно-ориентированная разработка программных средств. Комбинация методов, а также поддерживающих их отдельных инструментов и инструментов CASE, образует среду CASE, которая предназначена для компьютерно-ориентированного, методологического и инструментального обеспечения процесса разработки программных систем.

Известным недостатком инструментов CASE является их жесткая привязка к реализуемым методам, ограничивающая диапазон их использования, что при динамичности и разнообразии запросов на их применение вынуждает создавать новые CASE, приводя к значительным затратам ресурсов.

Решением этой проблемы является создание CASE-сред, гибкость которых предопределяется возможностью их адаптации к условиям применения и разработки. Проблему адаптации CASE-сред можно свести к реконфигурации реализуемого ими метода и к варьированию ассоциированным множеством инструментов поддержки. Так как основной акцент при создании и адаптации CASE-сред делается на работу с методами, то целесообразность решения этой задачи посредством ИСМ является очевидной.

1. Базовые концепции инженерии методов

Основной концепцией инженерии методов является *метод*.

Определение 1.1. Метод представляет собой упорядоченный набор нормативных предписанных действий, выполняемых акторами для достижения целей при проектировании программных средств.

Концептуально метод определяется *структурой*, состоящей из трех взаимосвязанных частей – процесса, нотаций и концепций [11]. *Процесс* описывает способ выполнения операций с учетом некоторых правил и рекомендаций разработчиком проекта. *Нотации* предназначены для документирования результатов применения метода. *Концепции* представляют собой понятийные категории, которые включаются в процесс и нотацию для описания домена применения метода.

Заметим, что данное определение созвучно приведенному в [11,с.9]: «метод представляет собой способ доступа к выполнению проекта разработки системы, состоящий из инструкций и правил, базированный на специфическом способе мышления и структурируемый систематическим образом на отдельные действия процесса с фиксацией промежуточных продуктов разработки».

Различие концепций *метода* и *процесса* состоит в следующем: метод предписывает способ проведения работ, а процесс

описывает полную деятельность по разработке ПС. Метод используется на всем протяжении процесса или этапов разработки ПС.

2. Базовые элементы инженерии ситуационного метода

Под *ситуационным* понимается метод, конструируемый для процесса проектирования ПС таким образом, чтобы полностью удовлетворять конкретной ситуации его применения. В основу инженерии метода положено его структурное представление в виде множества когерентных частей – *фрагментов*, связи и порядок применения которых регламентируются его конкретизированной конфигурацией.

Адаптация метода к ситуациям применения обеспечивается в ИСМ реконфигурацией ситуационного метода, модификацией его контента путем изменения состава фрагментов и их инструментальной поддержки, а также его параметрической настройкой.

Конфигурация ситуационного метода определяет обобщенную архитектуру проектируемой CASE-среды, а собранный ситуационный метод ее конкретизирует для целевого проекта ПС.

ИСМ использует базу методов, состоящую из множества фрагментов наиболее распространенных методов проектирования ПС. Самой серьезной методологической проблемой использования таких фрагментов при сборке методов является возможность их концептуальной несогласованности.

Таким образом, методология Meta-CASE базируется на последовательном применении и развитии следующих аспектов ИСМ: построения конфигурации ситуационного метода как ядра архитектуры проектируемой среды CASE, сборки ситуационного метода и его адаптации к требованиям конкретного применения.

До настоящего времени выбор метода разработки ПС для нового домена или в случае нестандартной проектной ситуации сводился к поиску в огромном множестве

уже существующих методов, что редко приводило к приемлемому результату. Зачастую неудовлетворительный результат поиска вынуждал разрабатывать собственный метод.

Эта проблема решается с помощью ИСМ, суть которой состоит в замене разработки конкретного метода построением параметризованной *конфигурации* структурированного метода, релевантной predeterminedному множеству ситуаций его применения, и сборки на ее основе ситуационного метода. Адаптация ситуационного метода к требованиям конкретного проекта ПС осуществляется посредством его реконфигурации, изменения состава компонентов и настройки параметров. В простейшем случае адаптация сводится к выбору пути внутри схемы конфигурации.

Определение 2.1. Конфигурация ситуационного метода представляет собой структурированное описание метода, удовлетворяющее predeterminedному множеству ситуаций применения.

Такая конфигурация описывает обобщенную структуру процесса проектирования ПС и отображается ориентированным графом целей и подцелей, вершинам которого сопоставлены критерии применения фрагментов метода. Конкретизация значений критериев отражает реальную ситуацию использования метода и служит базой для его формирования. Ситуационный метод создается посредством *сборки* фрагментов метода, характеристики которых удовлетворяют критериям применимости конфигурации метода с их конкретными значениями [12]. Сборка завершается связыванием фрагментов метода и компонентов инструментальной поддержки.

В основу инженерии ситуационного метода положены:

- представление метода как структуры, объединяющей множество концептуально когерентных параметризованных частей – его компонентов. Порядок применения компонентов задается сценарием, отображенным ориентированным графом целей проектирования;

- ситуація застосування методу визначається характеристиками домена застосування, середовища розробки проекту ПС і набором вимог замовника.

Ситуація методу відображає контекст використання ситуаційного методу і визначається двома складовими: ситуацією застосування, відповідною домену призначення, і ситуацією середовища розробки проектів ПС за допомогою методу, асоційованого з доменом його функціонування.

Визначення 2.2. Ситуація методу – це сукупність абстракцій доменів його призначення і функціонування.

Різниця в ситуаціях визначається різницями їх характеристик.

Визначення 2.3. *Характеристика ситуації* представляє собою детерміноване властивість ситуації розробки, фокусуюче увагу розробника на визначеній проблемі або аспекті, які повинні бути оброблені конфігурацією методу.

Ситуація відображає контекст використання методу, визначає концептуальне середовище проблемної області проекту, вимоги замовника і характеристики середовища розробки проекту [13]. На основі аналізу ситуації формується множина критеріїв застосування методу і його фрагментів.

Компоненти ситуаційного методу. В методології конструювання ситуаційного методу в ролі компонентів розглядаються його когерентні частини – *фрагменти*, реалізуючі передбачені дії для визначених аспектів загальної проблеми, розв'язаної методом.

Подібно методу, кожен його фрагмент складається з опису способів роботи (процеса), нотаций і концепцій. Фрагменту відповідають модель продукту і модель процесу, які пов'язані між собою реалізаційними рекомендаціями. Клас продуктів, фігуруючих в процесі розробки, представляється з допомогою *модель продукту*, а процес розробки

продукту з використанням цього фрагмента методу – з допомогою *модель процесу*.

Фрагмент розглядається як найменший значущий компонент методу, адресований визначеному аспекту проблеми. Концепція фрагмента методу визначена в [9, с.112] як «... описання методу інженерії ПС або будь-якої його когерентної частини». Виходячи з цього визначення, фрагментом методу вважається як повний метод, наприклад Object Modeling Technique (OMT), так і єдинична концепція, використовувана всередині OMT, наприклад «об'єкт», що дозволяє говорити про представлення фрагментів методу на різних *рівнях гранулярності*. Описання фрагмента методу містить набір передбачених ситуаційних факторів його застосування. Прикладами фрагментів методу можуть служити “*use case*” або «UML-діаграма», які є частинами цілого.

Таким чином, фрагменти – це базові будівельні блоки ІСМ, допускаючі повторне використання. Вони відрізняються за цілями, наприклад логічне проектування, специфікація потоків даних і т.д. В ролі фрагментів можуть використовуватися: методи, технологічні засоби і інструменти, класифіковані як фрагменти методів різних *типів*, залежних від їх характеристик і вибраного рівня *гранулярності*. Визначальними характеристиками фрагментів в ІСМ вважаються ситуаційні фактори.

В якості прикладів можна привести фрагменти методу проектування ПС на визначеному рівні гранулярності, відповідаючі за етапи аналізу або тестування загального життєвого циклу розробки ПС.

Кожний фрагмент методу повинен бути описаний на одному з мов інженерії методів.

Цепочки і альянси фрагментів. Фрагмент може бути використаний окремо і незалежно від інших фрагментів, але в ІСМ він використовується як частина ланки або альянсу методів.

Цепочка фрагментов реалізує *вертикальну інтеграцію* методів між різними рівнями абстракції розробки, наприклад рівнями концептуального і об'єктного моделювання.

Альянсу фрагментов відповідає *горизонтальна інтеграція* методів одного і того ж рівня абстракції розробки.

Методи і каркаси. В основі процесу побудови структури методу лежить використання допоміжної концепції *каркаса* методу, який представляє собою орієнтований граф *целей і підцелей*, ідентифікованих в результаті аналізу ієрархії цілей проекту ПС. Вершини каркаса відповідають *целям*, а дугам – *стратегіям*, що визначають навігацію від вихідної цілі до кінцевої. Кожній цілі каркаса методу сопоставляється ситуація, що визначає стан частин розробляваної ПС на певній стадії застосування методу.

Формуванню каркаса передують семантичний аналіз множини ситуацій застосування методу, який завершується визначенням виразительного множини характеристик ситуацій з наступним їх цільовим комбінуванням. Каркас відображає модель процесу проектування ПС, наприклад *водопадну* або *итеративну* моделі. Крім того, каркас визначає *домен* методу, в якому останній використовується.

Пакет конфігурації є елементом ИСМ і відповідає попередній обобщенній конфігурації ситуаційного методу, наступна адаптація якої задовольняє конкретним характеристикам проекту розробки ПС або їх комбінації.

Определение 2.4. Пакет конфігурації – це конфігурація певного базового ситуаційного методу, що задовольняє діапазону значень релевантних характеристик.

Пакет конфігурації формується із каркаса методу шляхом визначення для стратегій *передписаних* дій і зв'язування характеристик цих дій з їх

целями. Він служить базою для формування ситуаційного методу, здійснюваного шляхом вибірки з бази методів і наступної збирання окремих фрагментів і сопоставлення передписаним діям і їх характеристикам.

Определение 2.5. Ситуаційний метод – це тонко налаштований і адаптований пакет конфігурації, що використовується в проекті [14].

База методів. Ядром ИСМ є база методів, в якій зберігаються фрагменти методів різних типів. Крім того, її компонентами можуть бути: описання типових пакетів конфігурації, технологічних засобів і інструментів. Описання компонентів бази методів містять характеристики, що визначають їх призначення, ситуацію використання, способи і ступінь застосування.

Базовий метод розглядається як стартовий при конструюванні ситуаційного методу. В його ролі використовуються стандартні або комерційні методи, успішно апробовані на практиці, або методи власної розробки.

Фрагмент також можна розглядати як базовий метод на даному рівні його гранулярності. Він, в свою чергу, може складатися з інших фрагментів більш тонкої гранулярності.

3. Аспекти застосування ИСМ в методології MetaCASE

В методології MetaCASE інженерія ситуаційних методів повинна використовуватися для рішення двох ключових проблем проектування адаптованих серед CASE:

- формування архітектури CASE-серед на базі попередньо спроектованої конфігурації ситуаційного методу і наступної проблемної адаптації середовища до конкретної ситуації домена призначення шляхом реконфігурації цього методу;

- адаптації CASE-середовища до середовища розробки ПС шляхом параметричної на-

стройки ситуационного метода и корректировки его контента.

В разрабатываемой методологии MetaCASE используются все базовые элементы ИСМ. *Ситуация* разработки CASE-сред определяет контекст проектирования как самих сред, так и разработки в них ПС. *Конфигурации* метода соответствует ситуационный метод, адекватный проектируемой CASE-среде, а *контент* метода формируется путем выбора из базы фрагментов методов, релевантных предписанным действиям пакета.

В состав базы методов MetaCASE входят типовые пакеты конфигураций, фрагменты методов с их описаниями, методики и руководства по их применению. В описаниях фрагментов методов определяются используемые в них концепции, характеристики их применимости и методической функциональности. Степень эффективности использования такой базы для сборки ситуационных методов зависит от степени дискретности ее компонентов, т.е. фрагментов методов. Поэтому предусловием ведения базы методов является задание способа *фрагментации методов*.

Проблема согласования конфликтующих определений одинаковых концепций в смежных фрагментах методов решается вводом понятия *суперметода*, для которого описывается общее пространство концепций, выбираемое посредством определения наименьшего общего знаменателя концепций методов данного домена. Другим решением проблемы является ввод в структуру метода промежуточных компонентов, согласующих концепции смежных фрагментов метода.

Определение концепций методов осуществляется их моделированием. Для описания моделей данных метода используются диаграммы типа «*сущность-связь*», в частности ERR. В том случае, когда ER-диаграмм недостаточно, выразительность модели может быть усилена логикой предикатов первого порядка.

Инженерия ситуационного метода реализуется в методология MetaCASE следующей последовательностью шагов:

1. Определение *ситуации* применения CASE-среды на основе описания требований заказчика, характеристик домена назначения, целей проекта и среды разработки ПС.

2. Построение на основе анализа ситуации применения CASE-среды *каркаса* метода и *пакета конфигурации* ситуационного метода как базы формирования архитектуры CASE-среды.

3. Определение *контента* метода путем выбора из *базы методов* фрагментов методов, релевантных этому пакету конфигурации и покрывающих основные фазы жизненного цикла процесса проектирования ПС в CASE-среде.

4. Сборка ситуационного метода с целью конкретизации методологической архитектуры CASE-среды.

5. Согласование между собой выбранных фрагментов методов на концептуальном уровне.

6. Адаптация CASE-среды к требованиям конкретного проекта ПС и среды его разработки путем реконфигурации и параметрической настройки ситуационного метода и связывания фрагментов методов с инструментальной поддержкой.

Предложенная схема методологии является эффективной, поскольку предлагает строгую инженерную поддержку для построения CASE-среды и обеспечивает возможность адаптации к условиям применения как проблемного характера, так и задаваемым средой разработки ПС. В качестве полезных побочных эффектов такого подхода можно рассматривать возможность *типизации конфигураций* ситуационных методов и *повторного использования* компонентов базы методов.

Ниже анализируются вопросы усиления эффективности использования инженерной поддержки методологии MetaCASE и возможные пути их решения.

Фрагментация методов осуществляется посредством выделения когерентных

множеств концепций и технологий, ориентированных на охват конкретного аспекта проектирования ПС, и описания фрагментов с указанием релевантных наборов ситуационных факторов, регламентирующих ситуацию использования фрагмента. Целью фрагментации является формирование контента базы методов методологии.

Ситуационные факторы содержат характеристики фрагмента метода и определяют его применимость к специфической ситуации. И, наоборот, для заранее определенной ситуации ситуационный фактор указывает фрагмент метода, наиболее приемлемый в этой ситуации. Способ спецификации ситуационных факторов, в простейшем случае, сводится к указанию назначения и характеристик, примерные значения которых приведены ниже:

Назначение = {моделирование, дизайн интерфейса, компиляция программ, редактирование текстов, тестирование программ,...}.

Характеристики = {объектный или структурный подход к моделированию, язык диаграмм, тестирование правильности решений, управление версиями при редактировании,...}.

Для более формализованного описания ситуационных факторов требуется разработка специального языка инженерии методов.

Эвристики. Ситуационные факторы связываются с фрагментами методов с помощью *эвристик*, представляющих собой эмпирические правила, которые определяют как фрагмент метода, так и ситуацию его использования. Применяются правила вида IF...THEN... с логическим выражением, регламентирующим определение конкретной проектной ситуации в IF-части, и фрагментом метода, который должен быть подключен в этой ситуации, в THEN-части, например:

IF сложность данных модели высока
THEN модель следует описывать на естественном языке

Выбор фрагментов метода все еще остается главной проблемой, так как формального способа обоснования эффективности выбора пока не существует.

Выбору фрагментов метода предшествуют анализ ситуации разработки проекта ПС (рисунок) и построение каркаса метода проектирования. Характеристики ситуации определяются в результате оценки совокупности целей и подцелей проекта, а их связывание с вершинами каркаса завершает формирование пакета конфигурации ситуационного метода.

Сборка метода. Процесс сборки базируется на правилах, обеспечивающих *качество* ситуационного метода. Метод должен соответствовать ситуации процесса разработки проекта ПС и одновременно быть непротиворечивым, результативным и надежным. В основе процесса сборки метода лежит *стратегия*, реализующая подход «сверху–вниз» или «снизу–вверх». Так, например, при использовании базового метода следует применять стратегию «сверху вниз».

Гарантию качества сборки обеспечивает удовлетворение следующих критериев:

- *концептуальная согласованность* – непротиворечие определений одинаковых концепций смежных фрагментов;
- *методическая полнота* – наличие всех операций, технологических приемов и концепций фрагментов, из которых конструируется метод;
- *методическая минимальность* – наличие операций, технологических приемов и концепций, минимально необходимых для разработки ПС по этому методу.

Методическая полнота обеспечивается формальным путем, использующим логику предикатов первого порядка. Методическая минимальность достигается корректной фрагментацией методов.

База методов. Контент базы методов должен ориентироваться на методологическую поддержку инженерии ситуационных методов и быть представлен в *case-форме*, формулирующей специфические

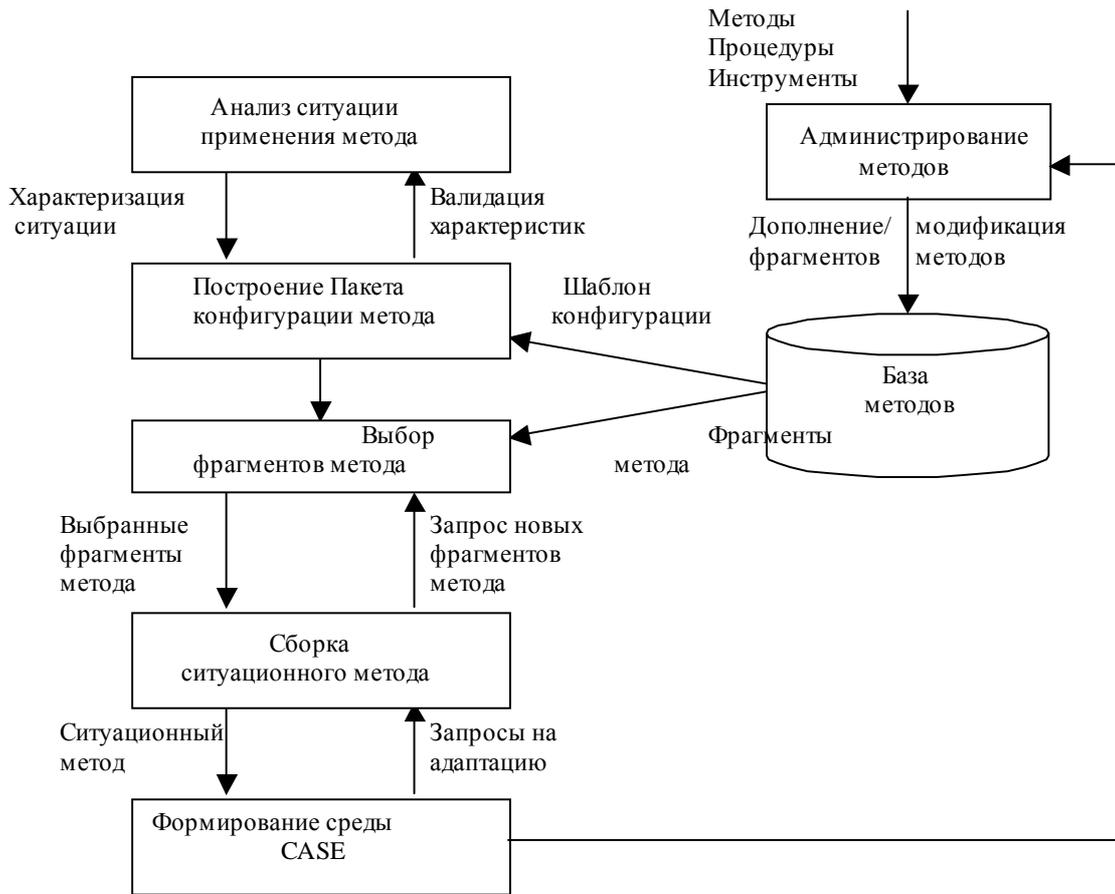


Рисунок. Процесс инженерии ситуационного метода

случаи практики проектирования CASE-сред.

База методов должна быть реализована как база данных. Роль *контента* базы методов играют такие наборы: описаний фрагментов методов, ситуационных факторов, эвристик, описаний типовых предписанных действий, типовых пакетов конфигураций, инструментов и т.д.

Выводы

Предложен новый подход к проектированию CASE-сред, основанный на их адаптации к ситуации применения. Адаптационное проектирование является новым, еще не достаточно разработанным направлением в создании методологий метатехнологических систем типа MetaCASE. Концептуальную основу создаваемой методологии MetaCASE составляет *инженерия*

ситуационных методов, а предложенное нами ее расширение позволило разделить адаптацию CASE-сред на две стадии: проблемную – на этапе проектирования конфигурации ситуационного метода, и функциональную – на этапе эксплуатации среды. Так, реконфигурация ситуационного метода, являющегося ядром архитектуры CASE-сред, позволяет удовлетворить спектр требований домена назначения, а параметрическая настройка метода используется для адаптации CASE-среды к среде разработки программных систем.

Эффективность разрабатываемой методологии MetaCASE определяется гибкостью процесса проектирования CASE-сред, адаптацией проблемной и функциональной архитектуры к условиям применимости, а также повторным использованием конфигураций и компонентов мето-

дов. Такие возможности отсутствуют в существующих MetaCASE. Поэтому реализация предложенной методологии MetaCASE позволит обеспечить экономию времени и трудозатрат при разработке новых CASE-сред.

Перспективы развития данной работы определены в области разработки архитектуры базы методов, языка инженерии ситуационных методов и формализации подхода к фрагментации методов.

1. *Avison D.E. Fitzgerald G.*, Information Systems Development: Methodologies, Techniques and Tools. Third ed. // McGraw Hill, 2003.
2. *Моренцов Е.И.* Метатехнология и концепции создания специализированных технологий программирования // Проблемы программирования. – 1998. – Вып. 3. – С. 15-22.
3. *Henninger S.*, et al. Supporting Adaptable Methodologies to Meet Evolving Project Needs // In XP/Agile Universe, 2002.
4. *Desmond D'Souza & Alan Wills*, Objects, Components and Framework with UML–The Catalysis Approach // Addison-Wesley–Longman, 1998.
5. *Henderson-Sellers B.*, Method engineering for OO systems development // Communications of the ACM, 46(10), 2003 – P. 73-78 .
6. *Firesmith D.G. and B. Henderson-Sellers*, The OPEN Process Framework // London: Addison Wesley, 2002.
7. Software Development Method Tailoring at Motorola. *Fitzgerald B., Russo N., O'Kane T.* // Communications of the ACM, 46(4), 2003 – P. 65-70 .
8. *Зинькович В.М., Моренцов Є.І.* Концепції та засоби метатехнології для створення проблемно-орієнтованих технологій програмування // Проблемы программирования. – 2002. – № 3-4. – С. 74-82.
9. Situational Method Engineering for information system project approaches. *Harmsen F., Brinkkemper S. and Han Oei.* // In Proceedings of the CRIS'94 conference, Maastricht, The Netherlands, September 1994.
10. *Harmsen A.F.* Situational Method Engineering. Doctoral Dissertation // Utrecht University, The Netherlands, 1997.
11. Object-Oriented Systems Analysis And Design Using UML. *Bennett S., McRobb S., Farmer R.*, // McGraw-Hill, London, UK, 2002.
12. Meta-modelling based assembly techniques for situational method engineering. *Brinkkemper S., Saeki M., Harmsen F.* // Information Systems, vol. 24 (3), 1999 – P. 209–228.
13. *Rolland C., Prakash N.* A proposal for context-specific method engineering. // In S. Brinkkemper, K. Lyytinen & R. J. Welke (Eds.), Method engineering: Principles of method construction and tool support (P. 191-208). Atlanta: Chapman & Hall, 1996.
14. *Brinkkemper S.* Method engineering: engineering of information systems development methods and tools. // Information and Software Technology 38, 1996 – P. 275–280.

Получено 21.03.05

Про авторів

Зинькович Валерий Митрофанович,
старший научный сотрудник
Моренцов Евгений Иванович
кандидат технических наук,
старший научный сотрудник

Место работы авторов:

Институт программных систем
НАН Украины,
03187, г.Киев,
Проспект Академика Глушкова, 40
Тел. (044) 526 3098, 526 4286