

СИСТЕМНЫЙ АНАЛИЗ, УПРАВЛЕНИЕ И ОБРАБОТКА ИНФОРМАЦИИ

УДК 004.41

В. Е. ГВОЗДЕВ, А. Е. КОЛОДЕНКОВА

ФОРМАЛИЗАЦИЯ ПРОЦЕДУРЫ ПРОЕКТИРОВАНИЯ АРХИТЕКТУР ПРОГРАММНОЙ СИСТЕМЫ

Рассматривается подход к формализации проектирования архитектур сложной программной системы. В основе подхода лежит построение матриц вида «объект-свойство», а также выделение на их основе непересекающихся классов объектов. *Программная система ; матрица взаимосвязей ; иерархическая классификация (дендрограмма) ; архитектура требований ; архитектура информационных задач ; архитектура данных*

ВВЕДЕНИЕ

Одним из основных вопросов, возникающих на ранних стадиях проектирования программной системы, является проектирование архитектуры. В известной литературе [1, 2], посвященной программным проектам, термин «архитектура программной системы» имеет множество толкований. В настоящей работе под архитектурой программной системы, следуя [3], понимается совокупность существенных решений, определяющих: организацию программной системы, выбор структурных элементов системы и их интерфейсов.

Программной системе можно поставить в соответствие множество архитектур: функциональную, задач, программных компонентов, организационную, обеспечивающую и др. Проектирование упомянутых архитектур, как и любая задача декомпозиции системы на подсистемы, имеет множество решений. При этом полностью формализовать процедуру проектирования архитектур вряд ли возможно. Окончательный выбор зависит от множества обстоятельств, к которым относятся: взаимосвязанность различных архитектур, необходимость обеспечения преемственности развития уже существующих программных систем, доступные трудовые ресурсы; личные пристрастия проектировщиков и многие другие. Вместе с тем, чем больше вариантов построения архитектур будет рассмотрено, тем выше обоснованность окончательного решения.

1. КОНЦЕПТУАЛЬНАЯ ОСНОВА

Концептуальной основой предлагаемого формализованного подхода к построению системы архитектур, соответствующей сложной программной системе, является описанная в литературе [4] «трехслойная архитектура приложений» (рис. 1).

Каждому из выделенных слоев соответствует своя архитектура требований; архитектура задач; архитектура данных. Этим архитектурам на логическом уровне ставится в соответствие иерархическая система, представленная на рис. 2. Здесь $КТ_1, КТ_2, \dots, КТ_N, \dots, КТ_M$ – непересекающиеся классы требований; $КЗ_1, КЗ_2, \dots, КЗ_L, \dots, КЗ_K$ – непересекающиеся классы информационных задач; $КД_1, КД_2, \dots, КД_Q, \dots, КД_R$ – классы данных, обеспечивающие решение информационных задач.



Рис. 1. Уровни абстракции в рамках трехслойной архитектуры приложений

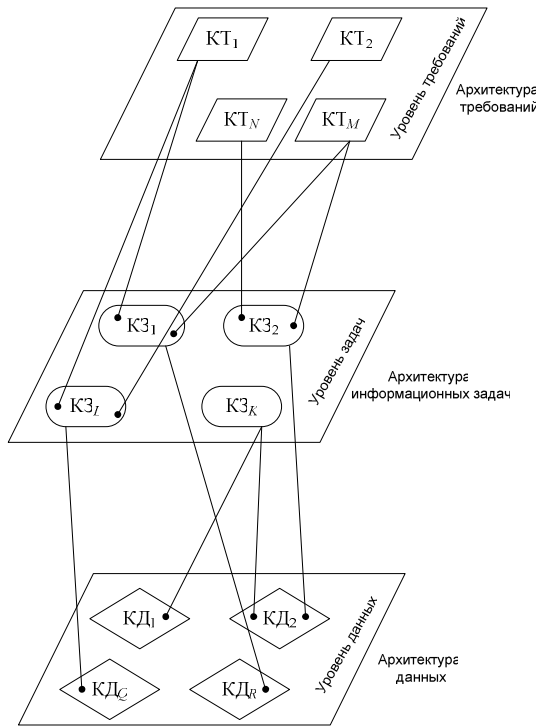


Рис. 2. Иерархическая схема архитектур, соответствующих разным уровням абстракции

2. МЕТОДИЧЕСКАЯ ОСНОВА

В известных работах [1, 2] подчеркивается важность разработки формальных моделей, соответствующих разным этапам жизненного цикла программного проекта. В настоящей работе рассматривается один из возможных подходов к формализации проектирования различных архитектур, соответствующей программной системе. Для того чтобы подчеркнуть, что речь идет не о программных модулях, а о программной компоненте, реализующей функционально законченный этап обработки исходных данных, в дальнейшем в качестве синонима «программной компоненты» используется термин «задача».

В работе [5] описаны матрицы проекций, которые, по сути, являются матрицами взаимосвязей и описывают связь между различными объектами программной системы. Однако в упомянутой работе не описаны подходы к выделению классов объектов.

В настоящей работе предлагается подход к выделению классов объектов на основе методов кластерного анализа.

Ограничения подхода:

1) предполагается, что заданы требования к функциональным возможностям проектируемой программной системы, причем требования

имеют одинаковую значимость с точки зрения заказчика и не меняются в ходе проекта;

2) предполагается, что все выделенные задачи, связанные с реализацией требований, являются новыми, то есть у разработчиков отсутствуют типовые решения, которые можно использовать для реализации проекта;

3) предположение о том, что число выделяемых подсистем заранее известно;

4) предлагаемый подход не ориентирован на использование в ситуации, когда новая программная разработка должна быть увязана с уже существующими (наследуемыми) программными системами.

3. ПРОЦЕДУРА ФОРМИРОВАНИЯ АРХИТЕКТУР ПРОГРАММНОЙ СИСТЕМЫ

Описание формальной процедуры продемонстрировано на примере формирования архитектуры требований, задач. Предложенный подход может быть использован и при проектировании других классов архитектур.

Формирование архитектуры основано на выделении классов требований, которые между собой связаны через классы информационных задач, расположенных на последующем уровне иерархии. Возможность представления требований в виде совокупности задач (visual table of contents) обосновано в литературе [6].

В качестве концептуальной основы формирования классов требований на начальных стадиях проекта целесообразно принять выделение подсистем на основе сходства требований.

Технологическую основу формирования архитектуры требований программной системы составляет построение и обработка матриц взаимосвязей требований и реализующих их задач. Формирование упомянутых матриц основано на построении списка задач, реализующих j -е требование; формированию на их основе списка задач, реализующих все требования

$\{3\} = \bigcup_{j=1}^N \{3\}_j$; формированию матриц вида (табл. 1).

Таблица 1
Сформированная матрица взаимосвязей «требования-задачи»

Требования \ Задачи	З ₁	З ₂	...	З _к
T ₁	+			
T ₂	+	+		
...
T _N	+	+		+

Здесь $\{Z\}_j$ – множество задач, решаемых при реализации j -го требования $j \in \{1, 2, \dots, N\}$; K – количество элементов $\{Z\}$, т. е. общее число задач, необходимых для решения всех N требований.

Необходимость решения k -й задачи $k \in \{1, 2, \dots, K\}$ при реализации j -го требования обозначается знаком «+» (при пересечении k -го столбца и j -й строки матрицы).

Заметим, что допускается множество вариантов формирования моделей состава задач, реализующих требования. При выделении задач следует стремиться к типизации задач, относящихся к различным требованиям.

На основе матрицы взаимосвязей рассчитываются меры сходства (различия) для всех пар требований. Полученные после вычислений значения соответствующих мер сводятся в квадратные матрицы, симметричные относительно главной диагонали (так называемые матрицы мер сходства) [6].

В литературе описаны различные подходы к построению дендрограмм на основе матриц мер сходства, а также методы обработки матриц мер сходства [7, 8]. Выбор конкретных мер зависит, в первую очередь, от конкретного исследования, а также от шкалы измерений [7]. Поэтому решение, получаемое на основе формальной процедуры, носит рекомендательный характер и может корректироваться при принятии окончательного решения с учетом мнений заказчика, владельца проекта, менеджера по продукту, менеджера программного проекта, разработчиков программных продуктов и по другим соображениям.

Основу формализации информационной поддержки формирования архитектуры информационных задач составляет формирование матриц взаимосвязей «задачи-данные».

Под решением информационной задачи понимается получение законченного информационного продукта, например, отчет, таблица, график и прочие.

Формирование упомянутой матрицы основано на построении списка данных, реализующих j -задачу (например, в виде IDEF0, DFD модели); формированию на их основе списка задач, реализующих все задачи $\{D\} = \bigcup_{j=1}^K \{D\}_j$; формированию матриц вида (табл. 2).

Здесь $\{D\}_j$ – множество данных, использованных при решении j -й задачи $j \in \{1, 2, \dots, K\}$; Q – количество элементов $\{D\}$, т.е. общее число

данных, необходимых для решения всех K задач.

Таблица 2

Сформированная матрица взаимосвязей «задачи-данные»

Задачи \ Данные	Д ₁	Д ₂	...	Д _Q
Z ₁	+			
Z ₂	+	+		
...
Z _K	+	+		+

Необходимость решения i -х данных при решении j -й задачи обозначается знаком «+» (при пересечении i -го столбца и j -й строки матрицы).

Основу формализации информационной поддержки формирования архитектуры данных составляет формирование матриц взаимосвязей «данные-имена данных».

Формирование упомянутых матриц основано на построении списка имен данных, реализующих j -е данные; формированию на их основе списка имен данных, реализующих все данные $\{ИД\} = \bigcup_{j=1}^Z \{ИД\}_j$; формированию матриц вида (табл. 3).

Таблица 3

Сформированная матрица взаимосвязей «данные-имена данных»

Данные \ Имена данных	ИД ₁	ИД ₂	...	ИД _Z
Д ₁	+			
Д ₂	+	+		
...
Д _Q	+	+		+

Здесь $\{ИД\}_j$ – множество имен данных, решаемых при реализации j -х данных $j \in \{1, 2, \dots, Q\}$; Z – количество элементов $\{ИД\}$, т.е. общее число имен данных, необходимых для решения всех Q данных.

4. ПРИМЕР

Дано:

а) пусть имеется пять требования $T = \{T_1, T_2, T_3, T_4, T_5\}$, которым ставится восемь задач $Z = \{Z_1, Z_2, Z_3, Z_4, Z_5, Z_6, Z_7, Z_8\}$;

б) пусть имеется восемь задач $\{Z\} = \{Z_1, Z_2, Z_3, Z_4, Z_5, Z_6, Z_7, Z_8\}$, которым ставятся данные $\{D\} = \{D_1, D_2, D_3, D_4, D_5, D_6\}$;

Требуется:

1) выбрать обоснованную процедуру для выделения классов требований;

2) выделить сходства классов задач.

Решение:

1) а) составим матрицу взаимосвязей требований и реализующих их задач (табл. 4).

Таблица 4
Матрица взаимосвязей «требования-задачи»

Требования \ Задачи	З ₁	З ₂	З ₃	З ₄	З ₅	З ₆	З ₇	З ₈
T ₁	+		+		+	+		
T ₂	+	+	+					+
T ₃		+		+	+			+
T ₄	+			+	+	+	+	
T ₅	+		+		+	+		+

б) посредством методологии IDEF0 составим матрицу взаимосвязей задач и реализующих их данных (табл. 5).

Таблица 5
Матрица взаимосвязей «задачи-данные»

Задачи \ Данные	Д ₁	Д ₂	Д ₃	Д ₄	Д ₅	Д ₆
З ₁	+		+		+	
З ₂	+	+		+		+
З ₃	+		+			
З ₄		+	+		+	
З ₅	+			+		+
З ₆			+	+	+	
З ₇	+	+		+	+	
З ₈	+	+	+		+	+

2) а) Результаты, получаемые при обработке матрицы (см. табл. 4) методом Жаккара и коэффициентом ассоциативности, представлены на рис. 3–4.

На рис. 3 представлена дендрограмма, построенная с помощью метода Уорда.

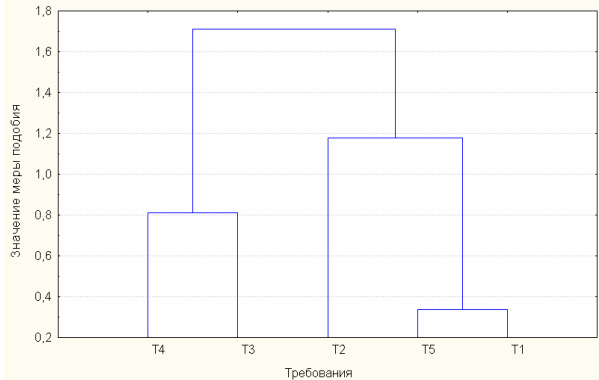


Рис. 3. Дендрограмма, построенная на основе матрицы меры сходства (мера сходства Жаккара)

На рис. 4 представлена дендрограмма, построенная с помощью центроидного метода.

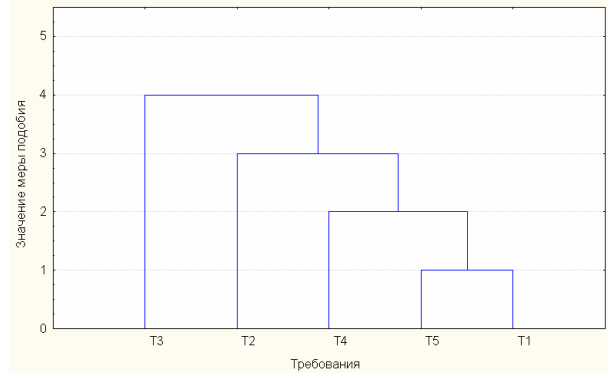


Рис. 4. Дендрограмма, построенная на основе матрицы меры сходства, сформированной с использованием коэффициента ассоциативности

$$\text{вида } S_{i,j} = \frac{P_{11}^{(i,j)}}{P_{11}^{(i,j)} + 2(P_{10}^{(i,j)} + P_{01}^{(i,j)})}$$

Здесь $P^{(i,j)} = P_{00}^{(i,j)} + P_{01}^{(i,j)} + P_{10}^{(i,j)} + P_{11}^{(i,j)}$, где

$P_{00}^{(i,j)}$ – доля задач из всего множества задач

при реализации всех N требований, не решаемых при реализации i -го и j -го требований;

$P_{01}^{(i,j)}$ – доля задач из всего множества задач

при реализации всех N требований, не решаемых при реализации i -го требования и решаемых при реализации j -го требования;

$P_{10}^{(i,j)}$ – доля задач из всего множества задач

при реализации всех N требований, решаемых при реализации i -го требования и не решаемых при реализации j -го требования;

$P_{11}^{(i,j)}$ – доля задач, решаемых при реализации

i -го и j -го требований из всего множества задач при реализации всех N требований.

Для данной исходной матрицы (см. табл. 4) целесообразно выделить следующие классы требований:

1) мера сходства Жаккара:

- метод одиночной связи: $Кл_1 = \{T_3, T_4\}$, $Кл_2 = \{T_1, T_5\}$, $Кл_3 = \{T_2\}$;

- метод полной связи: $Кл_1 = \{T_3, T_4\}$, $Кл_2 = \{T_1, T_5\}$, $Кл_3 = \{T_2\}$;

- центроидный метод: $Кл_1 = \{T_3, T_4\}$, $Кл_2 = \{T_1, T_5\}$, $Кл_3 = \{T_2\}$;

- метод Уорда: $Кл_1 = \{T_3, T_4\}$, $Кл_2 = \{T_1, T_5\}$, $Кл_3 = \{T_2\}$;

- метод медиан: $Кл_1 = \{T_3, T_4\}$, $Кл_2 = \{T_1, T_5\}$, $Кл_3 = \{T_2\}$;

2) коэффициент ассоциативности $S =$

$$= \frac{P_{11}}{P_{11} + 2(P_{10} + P_{01})}$$

- метод одиночной связи: $Кл_1 = \{T_3, T_4\}$, $Кл_2 = \{T_1, T_5\}$, $Кл_3 = \{T_2\}$;

- метод полной связи: $Кл_1 = \{T_3, T_4\}$, $Кл_2 = \{T_1, T_5\}$, $Кл_3 = \{T_2\}$;

- центроидный метод: $Кл_1=\{T_1, T_2, T_4, T_5\}$, $Кл_2=\{T_3\}$;
- метод Уорда: $Кл_1=\{T_3, T_4\}$, $Кл_2=\{T_1, T_5\}$, $Кл_3=\{T_2\}$;
- метод медиан: $Кл_1=\{T_1, T_3, T_4, T_5\}$, $Кл_2=\{T_2\}$.

б) Результаты, получаемые при обработке матрицы (см. табл. 5) методом Жаккара и коэффициентом ассоциативности, представлены на рис. 5-6.

На рис. 5 представлена дендрограмма, построенная с помощью метода Уорда.

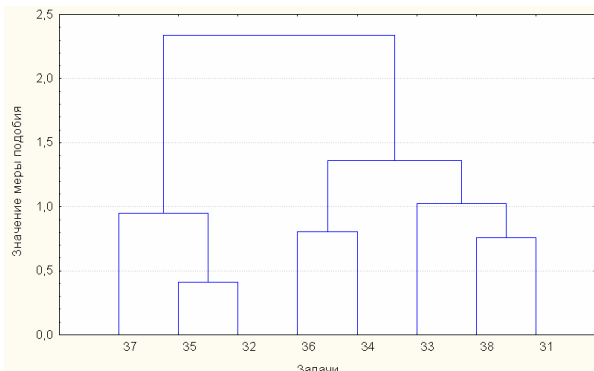


Рис. 5. Дендрограмма, построенная на основе матрицы меры сходства (мера сходства Жаккара)

На рис. 6 представлена дендрограмма, построенная с помощью метода одиночной связи.

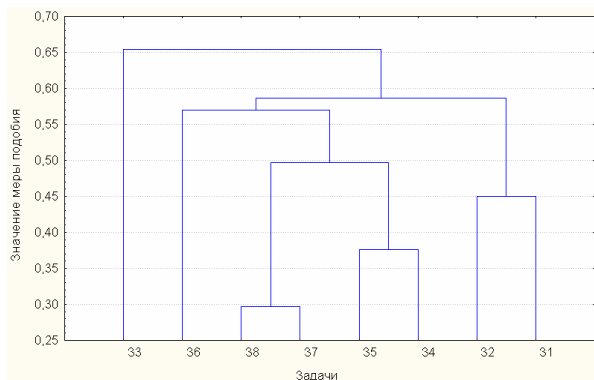


Рис. 6. Дендрограмма, построенная на основе матрицы меры сходства, сформированной с использованием коэффициента ассоциативности

$$\text{вида } S_{(i,j)} = \frac{2(p_{11}^{(i,j)} + p_{01}^{(i,j)})}{p^{(i,j)} + p_{11}^{(i,j)} + p_{00}^{(i,j)}}$$

Здесь $p^{(i,j)} = p_{00}^{(i,j)} + p_{01}^{(i,j)} + p_{10}^{(i,j)} + p_{11}^{(i,j)}$,

где $p_{00}^{(i,j)}$ – доля данных из всего множества данных при реализации всех L задач, не решаемых при реализации i -й и j -й задач;

$p_{01}^{(i,j)}$ – доля данных из всего множества данных при реализации всех L задач, не решаемых при реализации i -й задачи и решаемых при реализации j -й задачи;

$p_{10}^{(i,j)}$ – доля данных из всего множества данных при реализации всех L задач, решаемых при реализации i -й задачи и не решаемых при реализации j -й задачи;

$p_{11}^{(i,j)}$ – доля данных, решаемых при реализации i -й и j -й задач из всего множества данных при реализации всех L задач.

Для данной исходной матрицы (см. табл. 5) целесообразно выделить следующие классы задач:

1) мера сходства Жаккара:

- метод одиночной связи: $Кл_1=\{З_1, З_3, З_8\}$, $Кл_2=\{З_2, З_5, З_7\}$, $Кл_3=\{З_4\}$, $Кл_4=\{З_6\}$;

- метод полной связи: $Кл_1=\{З_4, З_6\}$, $Кл_2=\{З_1, З_8\}$, $Кл_3=\{З_2, З_5, З_7\}$, $Кл_4=\{З_3\}$;

- центроидный метод: $Кл_1=\{З_2, З_5, З_7\}$, $Кл_2=\{З_1, З_4, З_8\}$, $Кл_3=\{З_6\}$, $Кл_4=\{З_3\}$;

- метод Уорда: $Кл_1=\{З_4, З_6\}$, $Кл_2=\{З_1, З_3, З_8\}$, $Кл_3=\{З_2, З_5, З_7\}$;

- метод медиан: $Кл_1=\{З_2, З_4, З_5, З_7, З_8\}$, $Кл_2=\{З_6\}$, $Кл_3=\{З_1\}$, $Кл_4=\{З_3\}$;

2) коэффициент ассоциативности $S = \frac{2(p_{11} + p_{01})}{p + p_{11} + p_{00}}$:

- метод одиночной связи: $Кл_1=\{З_1, З_2\}$, $Кл_2=\{З_4, З_5\}$, $Кл_3=\{З_7, З_8\}$, $Кл_4=\{З_6\}$, $Кл_5=\{З_3\}$;

- метод полной связи: $Кл_1=\{З_1, З_2\}$, $Кл_2=\{З_4, З_5\}$, $Кл_3=\{З_7, З_8\}$, $Кл_4=\{З_6\}$, $Кл_5=\{З_3\}$;

- центроидный метод: $Кл_1=\{З_4, З_5, З_6, З_7, З_8\}$, $Кл_2=\{З_1\}$, $Кл_3=\{З_2\}$, $Кл_4=\{З_3\}$;

- метод Уорда: $Кл_1=\{З_1, З_2\}$, $Кл_2=\{З_7, З_8\}$, $Кл_3=\{З_3, З_4, З_5, З_6\}$;

- метод медиан: $Кл_1=\{З_4, З_5, З_6, З_7, З_8\}$, $Кл_2=\{З_1\}$, $Кл_3=\{З_2\}$, $Кл_4=\{З_3\}$.

Метод иерархической классификации подбирается проектировщиком индивидуально для исследуемой предметной области с учетом ее специфики [7]. Единых правил выбора не существует. Главным критерием для выбора метода классификации может являться хорошая интерпретируемость получаемых результатов, не противоречащих физическому смыслу изучаемой предметной области.

5. ЗАКЛЮЧЕНИЕ

Таким образом, в настоящей статье рассматривается формальный подход, обеспечивающий информационную поддержку проектирования архитектур, соответствующих программной системе, основанный на использовании известных методов кластерного анализа.

В силу неоднозначности подходов к выбору схожести задач, обработки матриц «объект-свойство», получаемые решения носят рекомендательный характер.

СПИСОК ЛИТЕРАТУРЫ

1. **Макконенелл, С.** Сколько стоит программный проект / С. Макконенелл. М. : «Русская редакция». СПб. : Питер, 2007. 297 с.
2. **Бэбьюли, Ф.** Управление проектом / Ф. Бэбьюли. М. : ФАИР-ПРЕСС, 2002. 208 с.
3. **Крылов, Е. В.** Техника разработки программ : Кн 2. Технология, надежность и качество программного обеспечения / Е. В. Крылов, В. А. Острейковский, Н. Г. Типикин. М. : Высшая школа, 2008. 469 с.
4. **Васкевич, Д.** Стратегии клиент/сервер. Руководство по выживанию для специалистов по реорганизации бизнеса / Д. Васкевич. Киев : Дианетика, 1996. 396 с.
5. **Грекул, В. И.** Проектирование информационных систем : учеб. пособие для студентов вузов, обучающихся по спец. в обл. инф. технологий / В. И. Грекул, Г. Н. Денищенко, Н. Л. Коровкина. М. : Интернет-ун-т инф. технологий, 2005. 304 с.
6. **Майерс, Г.** Надежность программного обеспечения / Г. Майерс. М. : Мир, 1980. 360 с.
7. **Андрейчиков, А. В.** Анализ, синтез и планирование решений в экономике / А. В. Андрейчиков, О. Н. Андрейчикова. М. : Финансы и статистика, 2002. 364 с.

8. **Кульба, В. В.** Оптимизация структур распределенных баз данных в АСУ / В. В. Кульба, А. Г. Мамиконов, С. А. Косяченков, И. А. Ужастов. М. : Наука, 1990. 240 с.

ОБ АВТОРЕ



Гвоздев Владимир Ефимович, зав. каф. автоматиз. проектир. инф. систем. Дипл. инж. электрон. тех-ки (УАИ, 1978). Д-р техн. наук по АСУ (УГАТУ, 2000). Иссл. в обл. АСУ, открытых инф. систем, прикл. статистики, теории надежности, контроля и управ. сост. окр. среды.



Колоденкова Анна Евгеньевна, доц. той же каф. Дипл. инж.-системотехн. по АСОИУ (УГАТУ, 2004). Канд. техн. наук (УГАТУ, 2007). Иссл. в обл. прикл. статистики, системн. анализа, контроля и управления территор. объектов, информац. систем, технологии системн. модел-я.