

ИНФОРМАТИКА И ИНФОРМАЦИОННЫЕ ПРОЦЕССЫ/INFORMATICS AND INFORMATION PROCESSES

DOI: <https://doi.org/10.60797/IRJ.2026.171.29> EDN: ARYWEG

ДИАГНОСТИКА СОСТОЯНИЙ ДЛЯ АДАПТИВНОГО УПРАВЛЕНИЯ ОШИБКАМИ В РАСПРЕДЕЛЕННЫХ СИСТЕМАХ

Научная статья

Принев М.А.^{1,*}, Леденева Т.М.²¹ORCID : 0009-0004-9704-6896;^{1,2} Воронежский государственный университет, Воронеж, Российская Федерация

* Корреспондирующий автор (metshislav[at]rambler.ru)

Предложена: 12.05.2026; Принята: 05.08.2026; Опубликовано: 17.09.2026

Аннотация

В статье представлены подходы к разработке инструментов мониторинга состояний сложной распределенной системы и управления ошибками, которые возникают в процессе ее функционирования. В предположении, что система представляет собой «черный ящик», для идентификации ошибок используются наблюдаемые значения показателей. Рассмотрены возможные типы требований к этим значениям и построены оценочные функции (d-оценки), которые позволяют оценить отклонение от нормы. Для идентификации состояния системы и составляющих ее объектов разработан комплекс лингвистических шкал, которые позволяют сформировать как количественные, так и качественные (лингвистические) оценки. Предложена типизация ошибок, возникающих в системе, и протоколы диагностики состояния объектов и системы в целом. Для управления ошибками разработана многоагентная система. Изложены принципы функционирования механизма назначения компенсирующих воздействий, который позволяет автоматизировать процесс разрешения инцидентов в системе. Предложен алгоритм адаптивного управления ошибками, основанный на корректировке рейтингов правил назначения компенсирующих воздействий.

Ключевые слова: управление ошибками, оценка состояния системы, инцидент, критическая ошибка, компенсирующее воздействие.

STATE DIAGNOSIS FOR ADAPTIVE ERROR CONTROL IN DISTRIBUTED SYSTEMS

Research article

Prinev M.A.^{1,*}, Ledeneva T.M.²¹ORCID : 0009-0004-9704-6896;^{1,2} Voronezh State University, Voronezh, Russian Federation

* Corresponding author (metshislav[at]rambler.ru)

Suggested: 12.05.2026; Accepted: 05.08.2026; Published: 17.09.2026

Abstract

The article presents approaches to developing tools for monitoring the state of a complex distributed system and controlling errors that occur during its operation. Assuming that the system is a 'black box', observed indicator values are used to identify errors. Possible types of requirements for these values are examined, and evaluation functions (d-scores) are constructed to estimate deviations from the norm. To identify the state of the system and its constituent objects, a set of linguistic scales has been developed, enabling both quantitative and qualitative (linguistic) assessments to be made. A classification of errors occurring in the system is suggested, along with protocols for diagnosing the state of the objects and the system as a whole. A multiagent system has been developed for error management. The principles for the operation of the mechanism for assigning compensatory actions are set out, which allows the process of resolving incidents within the system to be automated. An algorithm for adaptive error management is proposed, based on the adjustment of the ratings of the rules for assigning compensatory actions.

Keywords: error control, system state assessment, incident, critical error, compensating action.

Введение

Актуальной задачей для сложных систем с распределенной структурой является автоматизация процессов поддержания работоспособности системы. Под *системой с распределенной структурой* будем понимать совокупность большого числа компонентов, объединенных для достижения общей цели и обладающих высоким уровнем автономности и связности. Примерами таких систем могут служить системы с сервисной архитектурой (SOA), робототехнические системы и комплексы, то есть информационные и программно-аппаратные системы, обладающие такими свойствами, как распределенность, автономность компонентов, децентрализация, в которых глобальная цель достигается за счет кооперации и взаимосвязанности компонентов [1], [2], [3].

Под *объектом* Ob_i будем понимать структурную единицу системы (сервис, программный или программно-аппаратный модуль и т.д.), выполняющую определенную функцию, по сути, представляющую собой автономный функциональный элемент. Система состоит из множества объектов, которые находятся между собой в некоторых отношениях. Подмножества объектов могут образовывать подсистемы данной системы. Совокупность объектов и отношений между ними образует *структуру* системы.

В системах с распределенной структурой инструменты управления и мониторинга играют ключевую роль, обеспечивая контроль за состоянием и производительностью объектов, а также управление конфигурацией и обработку ошибок, выполняемую, как правило, командой разработчиков. Под *ошибкой* подразумевается внутреннее состояние системы, которое является отклонением от состояния, удовлетворяющего функциональным и нефункциональным спецификациям системы, тогда как сбой системы приводит к тому, что система перестает выполнять свои функции. *Управление ошибками* (error management) — это процесс выявления, анализа, исправления и предотвращения ошибок. В системах с распределенной структурой он включает принципы, инструменты и алгоритмы поддержания работоспособности, которые позволяют обрабатывать сбои в распределенных системах, состоящих из независимых объектов. Автоматизация этих процессов позволит значительно повысить эффективность и востребованность подобных систем. Примерами автоматизированных инструментов для мониторинга, визуализации, масштабирования и автоматического перезапуска компонентов в системах с сервисной архитектурой могут служить *Prometheus*, *Grafana*, *Kubernetes* [4], [5], [6].

Наряду с инструментальными решениями, в научной литературе активно развиваются теоретические подходы к управлению ошибками в распределенных системах. Существует ряд классификаций и обзоров, охватывающих методы отказоустойчивости, самоадаптации и активного управления отказами [7], [8], [9]. В работе [7] представлена систематическая классификация подходов к обеспечению отказоустойчивости в распределенных и облачных средах, выделяющая реактивные, проактивные, адаптивные и гибридные стратегии; в [8] систематизированы работы по проектированию самоадаптивных систем с использованием методов теории управления; в [9] рассмотрены методы активного управления отказами в динамических системах.

В реальных сложных распределенных системах построение релевантной аналитической модели требует полной и прозрачной структуры системы, что в условиях сложных распределенных сред зачастую принципиально недостижимо. Более того, многообразие классов ошибок и уникальность конфигураций делают любую предопределенную модель принципиально неполной. Известно, что даже базовые вероятностные модели [10], [11] сталкиваются с фундаментальными ограничениями применимости к реальным сложным системам, что служит прямым обоснованием необходимости перехода к предлагаемому адаптивному подходу, не требующему полного знания внутренней структуры объекта управления.

Это формирует принципиальную предпосылку для подхода к управлению ошибками, основанного на косвенном адаптивном воздействии на основную систему, трактуемую как «черный ящик», что актуализирует решения, связанные с диагностикой объектов в условиях неопределенности.

В статье представлены результаты исследования, направленного на разработку информационной среды для моделирования многоагентной системы (MAC) адаптивного управления ошибками для распределенных информационных систем. В рамках данного исследования разработан формальный аппарат для оценки состояния объектов по наблюдаемым параметрам в условиях неопределенности (на основе концепции «черного ящика»), обеспечивающий унифицированную обработку разнородных данных и формализацию пороговых значений для выявления инцидента и критической ошибки.

Для достижения цели были решены следующие задачи:

- классифицированы типы требований к значениям показателей;
- построены оценочные функции для количественной оценки отклонений; разработаны лингвистические шкалы для интерпретации состояний;
- формализованы пороги инцидента и критической ошибки;
- предложена архитектура модели MAC, использующей эти формализации для адаптивного управления ошибками.

Диагностика состояния системы

Процесс функционирования системы сводится к процессам функционирования объектов в их взаимосвязи, при этом каждому объекту Ob_j присуще некоторое состояние $Situ(Ob_j)$. Совокупность состояний объектов предопределяет состояние всей системы $Situ(S)$. В процессе функционирования объекта (системы) можно выделить *нормальное* состояние, которое соответствует эталонной работоспособности, и состояния, вызванные различными нарушениями в работе или внешними факторами, которые обуславливают отклонения от нормального состояния. Введем универсальную лингвистическую переменную $Situ = \text{«Состояние»}$ для формального описания состояний системы и составляющих ее объектов. Переменная принимает значение в шкале

$$Scale_{Situ} = \begin{cases} \text{норма (N)} \\ \text{слабое отклонение от нормы (Slight Deviation, SID)} \\ \text{сильное отклонение от нормы (Strong Deviation, StD)} \\ \text{существенное отклонение от нормы (Significant Deviation, SiD)}. \end{cases}$$

С учетом введенных кратких обозначений определим универсальную лингвистическую шкалу в виде $Scale_{Situ} = \{N, SID, StD, SiD\}$. Например, $Situ(Ob_j) = SiD$ означает, что состояние объекта Ob_j определяется как «существенное отклонение от нормы». Без ограничения общности будем считать, что для $Scale_{Situ}$ каждый терм задается нечетким числом [12] с прямоугольной функцией принадлежности $\mu_{Si}(x)$ (треугольной или трапециевидной). Данную шкалу будем использовать для идентификации всех типов ошибок, при необходимости корректируя универсальную шкалу [0,1].

В предположении, что каждый показатель является количественным, при идентификации ошибки, иницированной показателем, необходимо учитывать требование к его значению. Эти требования можно классифицировать следующим образом:

- требование первого типа — значение показателя не должно превосходить заданного значения ε ;

- требование второго типа — значение показателя должно превосходить заданное значение ε ;
- требование третьего типа — значение показателя не должно выходить за границы заданного диапазона $[\varepsilon', \varepsilon'']$

Зафиксируем объект Ob и рассмотрим для него значение некоторого показателя x , предполагая, что $x, \varepsilon, \varepsilon', \varepsilon'' \in [0, 1]$ (данное требование можно обеспечить за счет использования функций нормирования [13]). Поскольку требования к значениям показателя уникальны, то для оценки степени нарушения требования введем следующие величины:

- для требования первого типа (норма $x \leq \varepsilon$);

$$d_1(x, \varepsilon) = \begin{cases} 1, x \leq \varepsilon \\ \frac{\varepsilon(1-x)}{x(1-\varepsilon)}, x > \varepsilon \end{cases} \quad (1)$$

- для требования второго типа (норма $x \geq \varepsilon$);

$$d_2(x, \varepsilon) = \begin{cases} 1, x \geq \varepsilon \\ \frac{x(1-\varepsilon)}{\varepsilon(1-x)}, x < \varepsilon \end{cases} \quad (2)$$

- для требования третьего типа (норма $\varepsilon' \leq x \leq \varepsilon''$).

$$d_3(x, \varepsilon) = \begin{cases} 1, \varepsilon' \leq x \leq \varepsilon'' \\ d_2(x, \varepsilon'), x < \varepsilon' \\ d_1(x, \varepsilon''), x > \varepsilon'' \end{cases} \quad (3)$$

Таким образом, для каждого показателя с помощью величины $d \in [0, 1]$ можно оценить, нарушено ли требование к его значению: при $d = 1$ не нарушено; если $d < 1$, то требование нарушено, причем значение величины d показывает, как велико это нарушение. Оценки, задаваемые формулами (1) – (3), будем называть *d-оценками*.

На $[0, 1]$ введем частную лингвистическую шкалу, которая учитывает требование к значению показателя, используя термы $\{N, SlD, StD, SiD\}$. Для этого примем следующие соглашения:

1) для показателя с требованием первого типа носитель термина N есть промежуток $[0, \varepsilon)$, а на промежутке $(\varepsilon, 1]$ определены три термина SlD, StD, SiD с симметричными треугольными функциями принадлежности (a, l, r) , которые пересекаются в точках с ординатой 0,5;

2) для показателей с требованием второго типа носитель термина N есть промежуток $(\varepsilon, 1]$, а на промежутке $[0, \varepsilon)$ определены три термина SlD, StD, SiD с симметричными треугольными функциями принадлежности, которые пересекаются в точках с ординатой 0,5;

3) для показателей с требованием третьего типа носитель термина N есть промежуток $[\varepsilon', \varepsilon'']$, а на промежутках $[0, \varepsilon')$ и $[\varepsilon'', 1]$ определены по три термина SlD, StD, SiD с симметричными треугольными функциями принадлежности, которые пересекаются в точках с ординатой 0,5.

Определим, например, шкалу $\{N, SlD, StD, SiD\}$ для показателя первого типа. В этом случае терм N представляется трапецевидным числом $N = ([0, \varepsilon], 0, \frac{1-\varepsilon}{3})$, термы SlD, StD, SiD задаются треугольными нечеткими числами: $SlD = (\frac{2\varepsilon+1}{3}, \frac{1-\varepsilon}{3}, \frac{1-\varepsilon}{3})$, $StD = (\frac{\varepsilon+2}{3}, \frac{1-\varepsilon}{3}, \frac{1-\varepsilon}{3})$, $SiD = (1, \frac{1-\varepsilon}{3}, 0)$.

За счет совмещения шкал количественная оценка степени нарушения требований получает качественную интерпретацию, принимая значение некоторого термина, которому соответствует максимум функции принадлежности.

Пусть, например, к показателю предъявляется требование первого типа и $\varepsilon = 0,4$, тогда $a_{SlD} = 0,6, a_{StD} = 0,8, a_{SiD} = 1$. Если $x = 0,75$, то максимум функции принадлежности достигается для термина StD , а это означает, что отклонение по данному показателю от нормы *сильное*.

Таким образом, для каждого показателя можно определить степень нарушения требования как в количественной, так и в лингвистической форме.

Заметим, что у каждого показателя имеется своя частная лингвистическая шкала. Переход к универсальной лингвистической шкале осуществляется за счет процедуры унификации [14]. В результате каждый терм частной лингвистической шкалы представляется нечетким подмножеством универсальной лингвистической шкалы.

Пусть $S = \{S_r\}_{r=\overline{0,L}}$ и $S_U = \{S_k^U\}_{k=\overline{0,N}}$ – две лингвистические шкалы. Функция преобразования лингвистической шкалы S в лингвистическую шкалу S_U определяется следующим образом [13]:

$$\varphi_{S \rightarrow S_U}(S_i) = \left\{ \left(S_k^U / \max \left\{ \mu_{S_i}, \mu_{S_k^U} \right\} \right) \right\}_{k=\overline{0,N}}$$

Таким образом, базовую лингвистическую шкалу можно рассматривать как универсальный инструмент, позволяющий представить разнородную информацию унифицированным способом без потери качества исходной информации. Информация о состоянии объекта может быть представлена в числовой шкале в виде числа, в частной лингвистической шкале — в виде термина этой шкалы, а также в форме нечеткого подмножества термов универсальной лингвистической шкалы.

Предположим, что функционирование любого объекта Ob_j описывается набором показателей (параметров) $Ob_j = (P_1^j, \dots, P_{m_j}^j)$, каждый из которых принимает свое значение в некоторой количественной шкале. В общем случае объекту можно сопоставить *векторную оценку* его состояния. Предположим, что для каждого показателя задано требование к значениям, которое соответствует нормальному состоянию объекта.

Под *ошибкой, иницизированной показателем*, будем понимать нарушение требования к значению этого показателя.

Пусть $\{Ob_{j_1}, \dots, Ob_{j_N}\}$ — множество объектов, образующих систему. Состояние системы определяется состоянием ее объектов.

Под *ошибкой, иницизированной объектом*, будем понимать нарушение работоспособности данного объекта, которое приводит к нарушению качества функционирования системы.

Ошибка, иницизированная системой, возникает при нарушении работоспособности ее объектов в их взаимодействии.

Таким образом, с помощью полученной формализации состояний объекта можно конкретизировать понятия инцидента и критической ошибки.

Инцидент — это некритическая ошибка, соответствующая переходу статуса объекта в зону слабого отклонения. Его возникновение является сигналом для автоматического запуска механизма компенсации со стороны надстройки. Формально порог возникновения инцидента, h_{inc} , может быть определен как точка пересечения функций принадлежности к «норме» и к «сильному отклонению».

Критическая ошибка возникает при переходе в зону существенного отклонения, порог h_{cr} , что указывает на высокую вероятность скорого сбоя. Порог h_{cr} может быть определен как точка пересечения функций принадлежности к «сильному отклонению» и к «существенному отклонению». Такая ситуация может требовать эскалации, уведомления команды разработчиков и, потенциально, остановки работы для глубокого анализа.

Значения функций принадлежности порога инцидента и критической ошибки позволяют определить интервал использования процесса адаптивного управления ошибками, на котором возможно разрешение инцидента:

$$h_{cr} \leq \mu(x) \leq h_{inc}$$

Возможны ситуации, когда комбинация отклонений от нормального состояния приводит к нарушению работы объектов и системы в целом, что приводит к возникновению проблемы оценивания сложившейся ситуации в целом. Решением могут служить различные протоколы диагностики состояния объекта:

1. Выявление показателей, для которых нарушение требований является критичным и нарушает работу системы. Если в процессе мониторинга хотя бы для одного такого показателя возникнет существенное отклонение от нормы (*Sid*), то можно говорить о критической ошибке для объекта.

2. Определение системы приоритетов на множестве показателей, когда каждому показателю приписывается вес, отражающий его значимость для объекта. Данный подход позволяет учитывать предыдущий случай, когда накопление отклонений для менее значимых показателей может привести к критической ошибке.

3. Если все показатели равноценны, то для описания возникающих ситуаций необходимо сформулировать *принцип* учета отклонений: если требование нарушается (*хотя бы по одному* показателю, или *по всем* показателям, или *по большинству* показателей, или *по крайней мере по m из n показателей* — конкретный вариант определяется системой), то возникает отклонение в функционировании объекта. Такое приближенное количество показателей называется лингвистическим квантором и формально задается функцией квантификации [15].

В первом случае необходимо диагностировать те показатели, которые являются критическими, и, если обнаружено отклонение от нормы, то применять компенсирующее воздействие. В остальных случаях для учета требований к значению показателей можно использовать процедуру агрегирования векторной оценки в скалярную величину [16].

Таким образом диагностика работоспособности объектов и распределенной системы в целом будет осуществляться по наблюдаемым параметрам, полагая основную систему «черным ящиком», но при этом фиксируя отклонения статусов объектов от нормы при помощи предложенных моделей.

Разработка многоагентной системы для адаптивного управления ошибками

Адаптивное управление ошибками представляет собой динамический самонастраиваемый процесс поддержания работоспособности сложных распределенных систем, основанный на автоматическом выборе и применении к объекту компенсирующих воздействий из библиотеки алгоритмов компенсации. Выбор инициируется при обнаружении отклонения текущего статуса объекта от эталонного значения. Ключевым механизмом процесса назначения компенсирующих воздействий является алгоритм эволюционной оптимизации назначенных им рейтингов, отражающих относительную прогностическую оценку вероятности успешного разрешения ошибки.

Адаптивное управление ошибками позволяет автоматизировать процесс разрешения инцидентов, возникающих в процессе работы сложных распределенных систем, используя модели и методы обработки приближенной информации, базирующиеся на принципах диагностики объектов, изложенных выше. Это обеспечивает универсальный подход к обработке разнородных данных о состоянии системы [14], [15].

В контексте решаемой задачи объект может быть представлен кортежем:

$$Ob_j = (att, st, st1, st2),$$

где *att* — векторная оценка объекта, отражающая его признаки, качества и свойства или их отсутствие;

st — векторная оценка объекта, отражающая его признаки, качества и свойства или их отсутствие;

st1 — векторная оценка объекта, отражающая статус его компонентов в исходном состоянии;

st2 — векторная оценка объекта, отражающая статус его компонентов в измененном состоянии.

Компенсующее воздействие (КВ) в данном контексте представляет собой последовательность команд, направленных на поддержание работоспособности объекта, представленного в формате, совместимом с программным обеспечением объекта (точкой входа). Оно назначается при выявлении отклонений, доступных для мониторинга параметров компонентов объекта $st1_i$, от эталонных значений st_i при заданном промежутке допустимых значений падения статусов; результативность применения оценивается по полученным значениям измененного статуса $st2_i$.

Модель многоагентной системы (MAC) [17] является универсальной надстройкой к распределенной информационной системе. В дальнейших рассуждениях такую систему будем называть основной (ОС) и рассматривать как «черный ящик». Входной информацией для MAC являются наблюдаемые параметры ОС, объекты являются точками входа для подключения MAC.

Для формального описания архитектуры модели MAC введем ориентированный агентный граф $AG = (V, E)$, где:

V — множество вершин, соответствующих функциональным компонентам MAC;

$E \subseteq V \times V$ — множество направленных дуг, отражающих потоки сообщений и данных (рис. 1).

Вершины графа AG идентифицируются следующим образом:

$v_1^1, v_1^2, \dots, v_1^n$ — агенты-представители (A_{D_i});

v_2 — агент компенсации (A_{Calc});

v_5 — агент шина (A_{MB});

v_6^1 — база данных;

v_6^2 — библиотека алгоритмов компенсации.

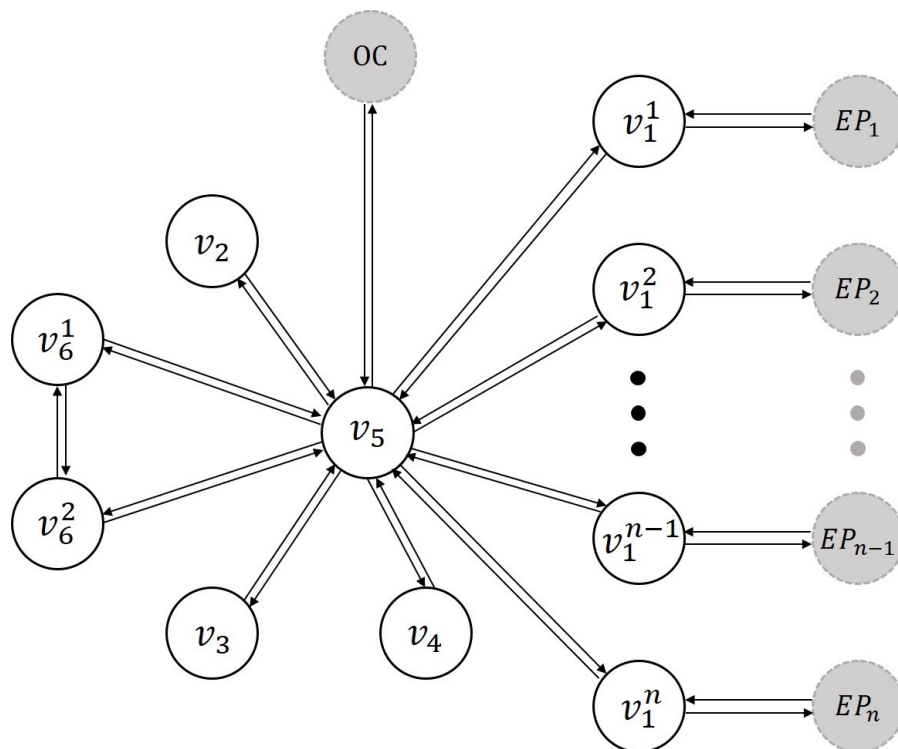


Рисунок 1 - Агентный граф AG
DOI: <https://doi.org/10.60797/IRJ.2026.171.29.1>

Взаимодействие с ОС и точками входа осуществляется через агентов-представителей и канал уведомлений. Внешние сущности (ОС и точки входа EP_i) на рисунке обозначены серым цветом и не входят в V , но участвуют во взаимодействии. Для осуществления адаптивного управления ошибками модель MAC должна выполнять следующие задачи:

- получение статусов показателей текущего состояния объекта;
- масштабирование;
- выявление инцидентов и их разрешение;
- мониторинг и уведомление об инцидентах и критических ошибках.

Выявление инцидента происходит в соответствии с принципами диагностики объектов, рассмотренными выше. Разрешение инцидентов и поддержание работоспособности ОС происходит за счет работы с компенсирующими воздействиями, обеспечивающейся алгоритмом адаптивного управления ошибками в распределенных информационных системах, представленном на рис. 2.

Каждому компенсирующему воздействию присуще не менее одного значения терма, по которым осуществляется выбор назначаемого КВ в соответствии со значением текущего статуса объекта и выбранной стратегией. Для конъюнктивной стратегии выбирается самое низкое из возможных значений терма, для дизъюнктивной — самое высокое, компромиссная стратегия подразумевает выбор промежуточного значения терма или его случайное назначение. На рис. 3 представлен пример определения термов для текущего показателя статуса объекта.

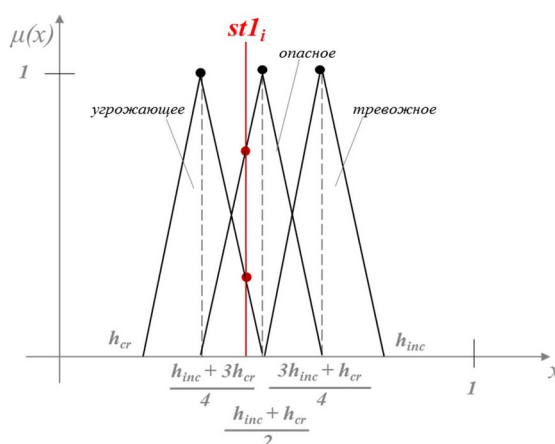


Рисунок 3 - Пример определения термов для текущего показателя

DOI: <https://doi.org/10.60797/IRJ.2026.171.29.3>

Благодаря использованию термов, решаются две задачи в процессе работы хранилища данных, а именно, обеспечивается больший охват задействованных компенсирующих воздействий, что приводит к ускорению эволюционной коррекции рейтинга, а также появляется возможность выбора пессимистичной, оптимистичной или компромиссной стратегии подбора компенсирующего воздействия, что увеличивает гибкость и адаптивность системы.

Одинаковые кластеры присущи КВ, ориентированным на решение схожих проблем, обусловленных определенным набором атрибутов объекта. Такие правила являются наименее универсальными, но при этом они могут стать наиболее точным инструментом для разрешения инцидента в соответствии с атрибутами объекта.

При получении значения текущего статуса объекта, удовлетворяющего условию $h_{cr} \leq stl_i \leq h_{inc}$ начинается трехуровневый подбор компенсирующего воздействия для разрешения инцидента. С коррекцией рейтинга после каждой итерации. На первом уровне выбирается КВ с наивысшим текущим рейтингом. Если КВ на первом уровне привело к разрешению инцидента, работа алгоритма прекращается, иначе переходим на второй уровень, на котором происходит выбор КВ по терму и рейтингу, что позволяет решить проблему, исходя из критерия значения текущего статуса объекта по выбранной стратегии. Для второго этапа назначается несколько итераций, если первое выбранное правило не сработало. Если второй этап не привел к решению проблемы, начинается работа на третьем уровне, когда запрашиваются КВ по рейтингу и кластеру, то есть происходит отработка критерия выбора КВ по наиболее подходящим для объекта параметрам. Для работы на третьем уровне назначается количество итераций для отбора компенсирующего воздействия в каждом из кластеров, приоритетных для объекта с последующей корректировкой рейтинга. Работа хранилища данных продолжается до момента разрешения инцидента или до истечения времени, удовлетворяющего условию: $t < t_{cr}$, где t_{cr} — промежуток времени, назначенный для работы хранилища данных.

Авторами разработана программная реализация предложенного в статье алгоритма работы с компенсирующими воздействиями «Универсальный модуль назначения компенсационных воздействий для адаптивного управления ошибками в распределенных информационных системах», позволяющая эмулировать различные сценарии для проведения экспериментального исследования. В ходе эксперимента была исследована динамика адаптации модели. Для этого на эмуляторе было обработано 100 инцидентов по сценарию, предусматривающему изменение условий — введение новых объектов после обработки сорока инцидентов, визуализация результата эксперимента представлена на рис. 4.

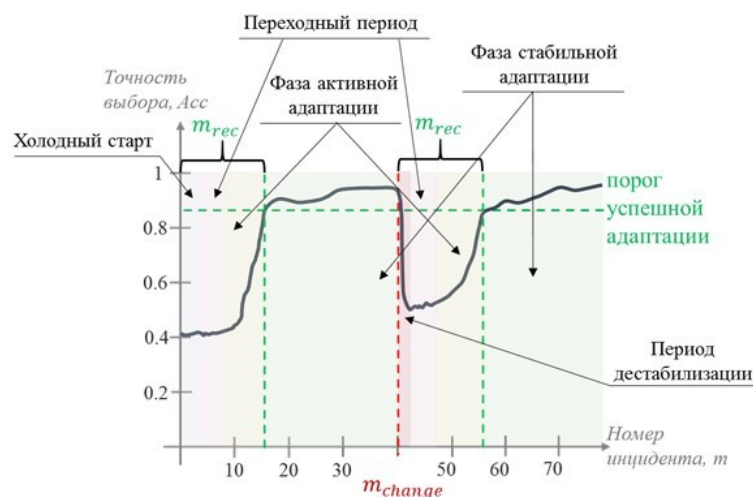


Рисунок 4 - Динамика адаптации модели при изменении условий

DOI: <https://doi.org/10.60797/IRJ.2026.171.29.4>

Полученный график показывает, что в зоне стабильной адаптации доля верных выборов, равная отношению корректно разрешенных инцидентов к их общему количеству, не опускается ниже 85%, изменение условий приводит к кратковременному спаду доли верных выборов, минимальное значение Acc при спаде больше, чем при холодном старте.

Заключение

Адаптивная система управления ошибками позволяет решить задачу автоматизации поддержания работоспособности сложных систем, обладающих распределенной структурой. Использование представленных в статье подходов к оценке состояния объектов и всей системы в целом позволяет использовать как количественную информацию, так и качественную, выраженную лингвистическими терминами для диагностики инцидента или критической ошибки. Хранилище данных системы обладает свойством самоорганизации и адаптивности. Работа хранилища данных имеет заданные временные ограничения, что повышает значимость вычислительной мощности и быстродействия системы. Решить эту проблему можно за счет использования многоагентных технологий [18], [19], [20], что обеспечит декомпозицию задач распределение мощностей и увеличение быстродействия системы за счет логики взаимодействия агентов и возможности распределенного решения задач.

Получены следующие результаты исследования, обладающие научной новизной:

- формализованы d -оценки для трех типов требований, отличающиеся универсальностью по отношению к типам показателей и позволяющие количественно оценить отклонение статуса объекта от нормы без построения аналитической модели;
- построены частные и универсальная лингвистические шкалы, отличающиеся возможностью преобразования разнородных данных в единый формат и позволяющие интерпретировать состояния объекта как в количественной, так и в качественной форме;
- определены формальные пороги инцидента h_{inc} и критической ошибки h_{cr} , отличающиеся объективностью задания через точки пересечения функций принадлежности и позволяющие четко разграничить зоны для автоматического и ручного вмешательства;
- предложены протоколы агрегирования отклонений, отличающиеся гибкостью выбора стратегии учета показателей и позволяющие учитывать как приоритеты, так и равнозначность показателей при диагностике объекта;
- разработана архитектура модели МАС, отличающаяся адаптивностью и универсальностью и позволяющая поддерживать работоспособность распределенных информационных систем без доступа к их внутренней логике.

Предложенный в статье формальный аппарат для оценки состояния объектов по наблюдаемым параметрам (в условиях неопределенности, на основе концепции «черного ящика») использован для построения архитектуры модели МАС, использующей эти формализации для адаптивного назначения компенсирующего воздействия для разрешения инцидентов. Проведен вычислительный эксперимент, показавший точность доли верных выборов КВ ~ 85% при минимальных интеграционных затратах, что подтверждает эффективность рассматриваемого в статье подхода.

Конфликт интересов

Не указан.

Рецензия

Все статьи проходят рецензирование. Но рецензент или автор статьи предпочли не публиковать рецензию к этой статье в открытом доступе. Рецензия может быть предоставлена компетентным органам по запросу.

Conflict of Interest

None declared.

Review

All articles are peer-reviewed. But the reviewer or the author of the article chose not to publish a review of this article in the public domain. The review can be provided to the competent authorities upon request.

Список литературы / References

1. Данилов И.Г. Сервис-ориентированная архитектура как основа для построения современных распределённых систем / И.Г. Данилов // Известия ЮФУ. Технические науки. — 2010. — № 7(108). — С. 168–173.
2. Solarte Martinez G.R. SOA Distributed Systems Architecture / G.R. Solarte Martinez // Scientia et Technica. — 2021. — Vol. 26, № 3. — P. 328–334.
3. Петренко А.А. Сравнение типов архитектуры систем сервисов / А.А. Петренко // Системные исследования и информационные технологии. — 2015. — № 4. — С. 48–62.
4. Sukhija N. Towards a Framework for Monitoring and Analyzing High Performance Computing Environments Using Kubernetes and Prometheus / N. Sukhija, E. Bautista. — Leicester : IEEE, 2019. — P. 257–262. — DOI: 10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00087
5. Ларионов В.С. Анализ работы веб-серверов с использованием пакета grafana / В.С. Ларионов, И.В. Дунин // Форум молодых ученых. — 2017. — № 11(15). — С. 544–549.
6. Фирсов Н.А. Мониторинг состояния инфраструктуры при помощи Prometheus & Grafana / Н.А. Фирсов // Современные тенденции и инновации в науке и производстве : материалы XII Международной научно-практической конференции, Междуреченск, 26 апреля 2023 года. — Междуреченск : Кузбасский государственный технический университет имени Т.Ф. Горбачева, 2023. — С. 461–463.
7. Kirti M. Fault-tolerance approaches for distributed and cloud computing environments: A systematic review, taxonomy and future directions / M. Kirti, A.K. Maurya, R.S. Yadav // Concurrency and Computation: Practice and Experience. — 2024. — Vol. 36, № 2. — DOI: 10.1002/cpe.8081
8. Patikirikoral T. A systematic survey on the design of self-adaptive software systems using control engineering approaches / T. Patikirikoral, A. Colman, J. Han et al. // 7th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2012). — Zurich, Switzerland : IEEE, 2012. — P. 33–42. — DOI: 10.1109/SEAMS.2012.6224389
9. Abbaspour A. A Survey on Active Fault-Tolerant Control Systems / A. Abbaspour, S. Mokhtari, A. Sargolzaei et al. // Electronics. — 2020. — Vol. 9, № 9. — P. 1513. — DOI: 10.3390/electronics9091513
10. Cha J.H. Generalizations of the Poisson Process and Their Reliability Applications / J.H. Cha // Reliability Analysis and Maintenance Optimization of Complex Systems; edited by Zhao Q.Q. Chung I.H. Zheng J. Kim J. — Cham : Springer, 2025. — DOI: 10.1007/978-3-031-70288-4_6
11. Карасев В.В. Идентификация логико-вероятностных моделей риска структурно-сложных систем с группами несовместных событий / В.В. Карасев, Е.Д. Соложенцев // Автоматика и телемеханика. — 2002. — № 3. — С. 97–113.
12. Пегат А. Нечёткое моделирование и управление / А. Пегат. — Москва : БИНОМ. Лаборатория знаний, 2013. — 798 с.
13. Леденева Т.М. Обработка нечеткой информации / Т.М. Леденева. — Воронеж : ВГУ, 2006. — 232 с.
14. Леденева Т.М. Моделирование оценочных систем на основе принципа многоальтернативности / Т.М. Леденева, С.Л. Подвальный // Системы управления и информационные технологии. — 2014. — № 3-1(57). — С. 155–161.
15. Леденева Т.М. Основы нечеткого моделирования : учебно-методическое пособие / Т.М. Леденева. — Воронеж : Издательский дом ВГУ, 2022. — 127 с.
16. Grabisch M. Aggregation functions: Means / M. Grabisch, J.-L. Marichal, R. Mesiar et al. // Information Science. — 2011. — № 181. — P. 1–22.
17. Prinev M.A. Implementation of Fuzzy Error Management in Systems with a Service Architecture Using Multi-Agent Technologies / M.A. Prinev // 2025 7th International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA). — Lipetsk, Russian Federation : IEEE, 2025. — P. 450–454. — DOI: 10.1109/SUMMA68668.2025.11302295
18. Леденева Т.М. Многоагентные системы: модели агентов и постановки задач, обеспечивающие их характерные свойства (часть 1) / Т.М. Леденева, М.А. Принев // Вестник Воронежского государственного университета. Серия: Системный анализ и информационные технологии. — 2024. — № 3. — С. 50–73. — DOI: 10.17308/sait/1995-5499/2024/3/50-73
19. Принев М.А. Многоагентные системы: обзор современных подходов к моделированию и проектированию (часть 2) / М.А. Принев, Т.М. Леденева, В.В. Гаршина // Вестник Воронежского государственного университета. Серия: Системный анализ и информационные технологии. — 2024. — № 4. — С. 167–190. — DOI: 10.17308/sait/1995-5499/2024/4/167-190
20. Принев М.А. Проблемы выбора и исследования эффективности способа интеграции распределённых систем / М.А. Принев // Актуальные проблемы прикладной математики, информатики и механики : сборник трудов Международной научной конференции, Воронеж, 12-14 декабря 2022 года. — Воронеж : Воронежский государственный университет, 2023. — С. 889–893.

Список литературы на английском языке / References in English

1. Danilov I.G. Servis-orientirovannaya arxitektura kak osnova dlya postroeniya sovremennyx raspredelyonnyx sistem [Service-oriented architecture as a basis for building modern distributed systems] / I.G. Danilov // Proceedings of SFedU. Technical Sciences. — 2010. — № 7(108). — P. 168–173. [in Russian]
2. Solarte Martinez G.R. SOA Distributed Systems Architecture / G.R. Solarte Martinez // Scientia et Technica. — 2021. — Vol. 26, № 3. — P. 328–334.



3. Petrenko A.A. Sravnenie tipov arxitektury' sistem servisov [System research and information technologies] / A.A. Petrenko // System research and information technologies. — 2015. — № 4. — P. 48–62. [in Russian]
4. Sukhija N. Towards a Framework for Monitoring and Analyzing High Performance Computing Environments Using Kubernetes and Prometheus / N. Sukhija, E. Bautista. — Leicester : IEEE, 2019. — P. 257–262. — DOI: 10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00087
5. Larionov V.S. Analiz raboty' veb-serverov s ispol'zovaniem paketa grafana [Analysis of web server performance using grafana] / V.S. Larionov, I.V. Dunin // Young Scientists Forum. — 2017. — № 11(15). — P. 544–549. [in Russian]
6. Firsov N.A. Monitoring sostoyaniya infrastrukturi pri pomoshchi Prometheus & Grafana [Monitoring infrastructure state with Prometheus & Grafana] / N.A. Firsov // Sovremennye tendencii i innovacii v nauke i proizvodstve [Modern Trends and Innovations in Science and Industry] : materials of the XII International Scientific and Practical Conference, Mezhdurechensk, April 26, 2023. — Mezhdurechensk : Kuzbasskii gosudarstvennii tekhnicheskii universitet imeni T.F. Gorbacheva, 2023. — P. 461–463. [in Russian]
7. Kirti M. Fault-tolerance approaches for distributed and cloud computing environments: A systematic review, taxonomy and future directions / M. Kirti, A.K. Maurya, R.S. Yadav // Concurrency and Computation: Practice and Experience. — 2024. — Vol. 36, № 2. — DOI: 10.1002/cpe.8081
8. Patikirikoral T. A systematic survey on the design of self-adaptive software systems using control engineering approaches / T. Patikirikoral, A. Colman, J. Han et al. // 7th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2012). — Zurich, Switzerland : IEEE, 2012. — P. 33–42. — DOI: 10.1109/SEAMS.2012.6224389
9. Abbaspour A. A Survey on Active Fault-Tolerant Control Systems / A. Abbaspour, S. Mokhtari, A. Sargolzaei et al. // Electronics. — 2020. — Vol. 9, № 9. — P. 1513. — DOI: 10.3390/electronics9091513
10. Cha J.H. Generalizations of the Poisson Process and Their Reliability Applications / J.H. Cha // Reliability Analysis and Maintenance Optimization of Complex Systems; edited by Zhao Q.Q. Chung I.H. Zheng J. Kim J. — Cham : Springer, 2025. — DOI: 10.1007/978-3-031-70288-4_6
11. Karasev V.V. Identifikaciya logiko-veroyatnostny'x modelej riska strukturno-slozhny'x sistem s gruppami nesovmestny'x sobyt'ij [Identification of logical-probabilistic risk models for structurally complex systems with groups of incompatible events] / V.V. Karasev, E.D. Solozhencev // Automation and Remote Control. — 2002. — № 3. — P. 97–113. [in Russian]
12. Pegat A. Nechyotkoe modelirovanie i upravlenie [Fuzzy modeling and control] / A. Pegat. — Moscow : BINOM. Laboratoriya znaniy, 2013. — 798 p. [in Russian]
13. Ledeneva T.M. Obrabotka nechetkoi informatsii [Fuzzy information processing] / T.M. Ledeneva. — Voronezh : VGU, 2006. — 232 p. [in Russian]
14. Ledeneva T.M. Modelirovanie ocenochny'x sistem na osnove principa mnogoal'ternativnosti [Modeling of evaluation systems based on the multi-alternative principle] / T.M. Ledeneva, S.L. Podval'nyj // Control Systems and Information Technologies. — 2014. — № 3-1(57). — P. 155–161. [in Russian]
15. Ledeneva T.M. Osnovi nechetkogo modelirovaniya [Fundamentals of Fuzzy Modeling] : undamentals of fuzzy modeling / T.M. Ledeneva. — Voronezh : Izdatelskii dom VGU, 2022. — 127 p. [in Russian]
16. Grabisch M. Aggregation functions: Means / M. Grabisch, J.-L. Marichal, R. Mesiar et al. // Information Science. — 2011. — № 181. — P. 1–22.
17. Prinev M.A. Implementation of Fuzzy Error Management in Systems with a Service Architecture Using Multi-Agent Technologies / M.A. Prinev // 2025 7th International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA). — Lipetsk, Russian Federation : IEEE, 2025. — P. 450–454. — DOI: 10.1109/SUMMA68668.2025.11302295
18. Ledeneva T.M. Mnogoagentnie sistemi: modeli agentov i postanovki zadach, obespechivayushchie ikh kharakternie svoystva (chast 1) [Multi-agent systems: agent models and problem statements ensuring their characteristic properties (part 1)] / T.M. Ledeneva, M.A. Prinev // Vestnik Voronezhskogo gosudarstvennogo universiteta. Seriya: Sistemnyi analiz i informatsionnye tekhnologii [Bulletin of Voronezh State University. Series: System Analysis and Information Technologies]. — 2024. — № 3. — P. 50–73. — DOI: 10.17308/sait/1995-5499/2024/3/50-73 [in Russian]
19. Prinev M.A. Mnogoagentny'e sistemy: obzor sovremenny'x podkhodov k modelirovaniyu i proektirovaniyu (chast' 2) [Multi-agent systems: a review of modern approaches to modeling and design (part 2)] / M.A. Prinev, T.M. Ledeneva, V.V. Garshina // Bulletin of Voronezh State University. Series: System Analysis and Information Technologies. — 2024. — № 4. — P. 167–190. — DOI: 10.17308/sait/1995-5499/2024/4/167-190 [in Russian]
20. Prinev M.A. Problemi vibora i issledovaniya effektivnosti sposoba integratsii raspredelyonnikh sistem [Problems of selection and efficiency study of distributed systems integration method] / M.A. Prinev // Aktual'nye problemy prikladnoy matematiki, informatiki i mekhaniki [Current Problems in Applied Mathematics, Computer Science, and Mechanics] : Proceedings of the International Scientific Conference, Voronezh, December 12–14, 2022. — Voronezh : Voronezhskii gosudarstvennii universitet, 2023. — P. 889–893. [in Russian]