

Научная статья

УДК 004.72

URL: <https://trudymai.ru/published.php?ID=185658>

EDN: <https://www.elibrary.ru/WQXZCT>

АРХИТЕКТУРА ОРГАНИЗАЦИИ ВЗАИМОДЕЙСТВИЯ БОРТОВЫХ ПРОГРАММНЫХ СИСТЕМ БЕСПИЛОТНЫХ ЛЕТАТЕЛЬНЫХ АППАРАТОВ В ГРУППЕ И С ПУНКТОМ УПРАВЛЕНИЯ В УСЛОВИЯХ НЕСТАБИЛЬНОГО ИНФОРМАЦИОННОГО ОБМЕНА

Иван Александрович Хорсик¹, Роман Родионович Шатовкин²✉

^{1,2} МИРЭА – Российский технологический университет,

Москва, Россия

¹ horsik.ivan@mail.ru

² shatovkin@yandex.ru ✉

Аннотация. Обоснована актуальность задачи разработки архитектуры организации взаимодействия бортовых программных систем беспилотных летательных аппаратов в группе и с пунктом управления в условиях нестабильного информационного обмена при отсутствии ретрансляторов. Определены состав распределенной системы «группа беспилотных летательных аппаратов под управлением пункта управления» и принадлежность ее элементов уровням иерархии, а также целевые и коммуникативные функции элементов системы. С учетом этих особенностей проведена постановка и исследованы существующие подходы к решению обозначенной задачи. На основе

результатов исследования разработана архитектура организации взаимодействия бортовых программных систем беспилотных летательных аппаратов в группе и с пунктом управления, основанная на их представлении в виде узлов распределенной иерархической системы доминантного типа, учитывающая целевые и коммуникативные функции этих узлов на различных уровнях рассматриваемой системы, условия и особенности применения группы беспилотных летательных аппаратов в горной местности, а также обоснованные требования к организации взаимодействия, отличающаяся тем, что представляет собой сервис-ориентированную архитектуру с кластерным разбиением групп беспилотных летательных аппаратов согласно реализуемому функционалу и одноранговым представлением взаимодействия узлов внутри кластера разведывательных беспилотных летательных аппаратов-не лидеров, что позволяет обеспечить организацию устойчивого взаимодействия в условиях нестабильного информационного обмена при отсутствии ретрансляторов.

Ключевые слова: группа беспилотных летательных аппаратов, пункт управления, нестабильный информационный обмен, бортовая программная система, архитектура организации взаимодействия

Для цитирования: Хорсик И.А., Шатовкин Р.Р. Архитектура организации взаимодействия бортовых программных систем беспилотных летательных аппаратов в группе и с пунктом управления в условиях нестабильного информационного обмена // Труды МАИ. 2025. № 143. URL: <https://trudymai.ru/published.php?ID=185658>

ARCHITECTURE OF INTERACTION ORGANIZATION ONBOARD SOFTWARE SYSTEMS OF UNMANNED AERIAL VEHICLES IN A GROUP AND WITH A CONTROL CENTER IN CONDITIONS OF UNSTABLE INFORMATION EXCHANGE

Ivan A. Khorsik¹, Roman R. Shatovkin²✉

^{1,2} MIREA – Russian Technological University

Moscow, Russia

¹ horsik.ivan@mail.ru

² shatovkin@yandex.ru✉

Abstract. The relevance of the task of developing an architecture for organizing the interaction of onboard software systems of unmanned aerial vehicles in a group and with a control center under conditions of unstable information exchange in the absence of repeaters is substantiated. The composition of the distributed system "a group of unmanned aerial vehicles under the control of a control center" and the belonging of its elements to the hierarchy levels, as well as the target and communicative functions of the system elements are determined. Taking into account these features, the formulation and existing approaches to solving the designated problem are studied. Based on the results of the study, an architecture has been developed for organizing the interaction of on-board software systems of unmanned aerial vehicles in a group and with a control center, based on their representation in the form of nodes of a distributed hierarchical system of a dominant type, taking into account the target and communicative functions of these nodes at various levels of the system under consideration, the conditions and features of using a group of unmanned aerial vehicles in mountainous terrain, as well as reasonable requirements for organizing

interaction, characterized in that it is a service-oriented architecture with a cluster partitioning of groups of unmanned aerial vehicles according to the implemented functionality and a peer-to-peer representation of the interaction of nodes within a cluster of reconnaissance unmanned aerial vehicles that are not leaders, which allows for the organization of stable interaction in conditions of unstable information exchange in the absence of repeaters.

Keywords: group of unmanned aerial vehicles, control center, unstable information exchange, on-board software system, architecture of interaction organization

For citation: Khorsik I.A., Shatovkin R.R. Architecture of interaction organization onboard software systems of unmanned aerial vehicles in a group and with a control center in conditions of unstable information exchange. *Trudy MAI*. 2025. No. 143. (In Russ.). URL: <https://trudymai.ru/eng/published.php?ID=185658>

Введение

В настоящее время беспилотные летательные аппараты (БПЛА) применяются в народном хозяйстве для решения достаточно широкого круга задач – поисково-спасательные операции, мониторинг окружающей среды, доставка грузов и т.д. Как правило, решение таких задач подразумевает групповое применение БПЛА.

Вопросам группового применения БПЛА посвящено достаточно большое количество публикаций. Это работы отечественных ученых – В.И. Меркулова [1–3], Н.В. Кима [4], Д.В. Моисеева [5], В.С. Моисеева [6–8], П.Е. Подлипьяна [9], А.В. Ананьева [10, 11], а также зарубежных ученых – М.Ф. Годвина, С.К. Спрая, Дж. К. Хедрика [12], П. Дж. Фахлстрема, Т.Дж. Глисона [13], В. Мюллера, Ф. Рейнерта, Д. Паллмера [14].

Однако подходы к групповому применению БПЛА, предлагаемые в этих работах, либо не рассматривают вопросов организации взаимодействия бортовых

программных систем БпЛА в группе и с пунктом управления (ПУ), либо подразумевают наличие БпЛА-ретрансляторов (что не всегда возможно в условиях ограниченности выделяемого ресурса), обеспечивающих их взаимодействие. Поэтому достигаемый в них положительный эффект нивелируется при отсутствии БпЛА-ретрансляторов в условиях нестабильного информационного обмена, обусловленного попаданием БпЛА в зоны отсутствия связи, что является характерным при проведении поисково-спасательных операций в горной местности. Следовательно, вопрос организации взаимодействия бортовых программных систем БпЛА в группе и с ПУ в условиях нестабильного информационного обмена при отсутствии БпЛА-ретрансляторов является актуальным и практически важным.

Решение данного вопроса подразумевает, в первую очередь, разработку соответствующей архитектуры, способной обеспечить взаимодействие бортовых программных систем БпЛА в группе и с ПУ в обозначенных условиях.

Цель статьи – разработка архитектуры организации взаимодействия бортовых программных систем БпЛА в группе и с ПУ, позволяющей в условиях нестабильного информационного обмена, обусловленного применением группы БпЛА под управлением ПУ при проведении поисково-спасательной операции в горной местности, обеспечить их устойчивое взаимодействие при отсутствии БпЛА-ретрансляторов.

Постановка задачи

Бортовые программные системы БпЛА в группе, принимающей участие в поисково-спасательной операции под управлением ПУ, вполне правомерно представить элементами (или узлами) распределенной вычислительной системы «группа БпЛА под управлением ПУ», каждый из которых в параллели с другими элементами решает свою задачу в интересах достижения общей цели по спасению людей, терпящих бедствие.

Состав распределенной системы «группа БпЛА под управлением ПУ», принадлежность ее элементов (узлов) уровням иерархии, а также их целевые и коммуникативные функции представлены в таблице 1.

Таблица 1 – Состав распределенной системы «группа БпЛА под управлением ПУ», принадлежность ее элементов (узлов) уровням иерархии и их функции

Уровень иерархии системы	Элементы (узлы) системы	Функции	
		Целевые	Коммуникативные
Высший	ПУ	1. Постановка задачи. 2. Определение способа и параметров решения задачи: – задание района и определение площади проведения поисково-спасательных работ; – расчет требуемого ресурса для решения задачи – количества разведывательных БпЛА, а по результатам разведки – и транспортных БпЛА; – задание способа поиска и параметров движения БпЛА с учетом рельефа и погодных условий. 3. Выработка решений по формированию маршрутов БпЛА	1. Прием и анализ полученной от разведывательного БпЛА-лидера группы видовой и служебной информации. 2. Формирование и выдача команд управления разведывательному БпЛА-лидеру группы (команд непосредственно БпЛА-лидеру группы и команд для ретрансляции транспортным БпЛА)
Средний	Разведывательный БпЛА-лидер	1. Обнаружении терпящих бедствие людей и фиксации их физического состояния по доступным внешним признакам	1. Взаимный обмен видовой и служебной информацией с БпЛА в своей группе. 2. Сбор и обобщение видовой и служебной информации от разведывательных БпЛА-не лидеров группы и служебной информации от транспортных БпЛА. 3. Выдача обобщенной видовой и служебной информации на ПУ. 4. Выдача команд управления и служебной информации БпЛА-не лидерам в группе. 5. Ретрансляция команд управления ПУ транспортным БпЛА
Низший	Группа из N разведывательных БпЛА-не лидеров		1. Взаимный обмен видовой и служебной информацией с БпЛА в своей группе. 2. Прием команд управления (от ПУ – для БпЛА-лидера и от БпЛА-лидера – для БпЛА-не лидеров)
	Группа из K транспортных БпЛА	1. Доставка предметов первой необходимости	1. Прием ретранслированных команд управления ПУ от разведывательного БпЛА-лидера группы. 2. Выдача служебной информации БпЛА-лидеру группы

Группа разнородна по своему составу и включает как разведывательные, так и транспортные БпЛА. Основная задача разведывательных БпЛА – обнаружение терпящих бедствие людей и фиксация их физического состояния по доступным внешним признакам; а основная задача транспортных БпЛА – доставка предметов первой необходимости.

Группа БпЛА применяется в поисково-спасательной операции в горной местности, что обуславливает сбои в связи между отдельными БпЛА в группе друг с другом и с ПУ, то есть нестабильный информационный обмен.

Кроме того, группа является иерархической системой доминантного типа: ПУ осуществляет управление разведывательным БпЛА-лидером, который, в свою очередь, управляет N разведывательными БпЛА-не лидерами и K транспортными БпЛА.

Обозначенные характеристики, функционал элементов (узлов), условия и особенности применения группы БпЛА являются исходными данными при обосновании требований и разработке архитектуры организации взаимодействия бортовых программных систем БпЛА в группе и с ПУ в условиях нестабильного информационного обмена.

Исследование существующих подходов к решению задачи

Исследование существующих концепций проектирования распределенных систем. Основными концепциями проектирования распределенных вычислительных систем в настоящее время являются: концепция ACID (Atomicity, Consistency, Isolation, Durability); концепция BASE (Basically Available, Soft state, Eventually consistent); концепция CAP (Consistency, Availability, Partition tolerance).

Концепция ACID использует набор свойств: атомарность, согласованность, изолированность, долговечность [15]. Atomicity (атомарность) – это свойство гарантирует, что если транзакция не может быть выполнена полностью, то все внесенные изменения, будут отменены. Consistency (согласованность) – это свойство гарантирует, что система всегда находится в согласованном состоянии до и после выполнения транзакции. Isolation (изолированность) – это свойство транзакций в системе, которое гарантирует, что выполнение одной транзакции не будет затронуто выполнением другой транзакции. Оно означает, что каждая транзакция должна быть изолирована от других транзакций, выполняющихся параллельно. Durability (долговечность) – это свойство транзакций, которое гарантирует, что после успешного завершения транзакции изменения данных будут сохранены даже в случае сбоя системы.

В целом, концепция ACID делает основной упор на надежность данных и их целостность, за счет чего обеспечивает их согласованность в любой момент времени. Однако, эта концепция может стать узким местом в производительности при работе с большими объемами данных.

Концепция BASE использует набор свойств: в основном доступный, мягкое состояние, постепенно согласованный [16]. Basically Available (в основном доступный) – это свойство обозначает, что система должна быть доступна для обработки запросов и предоставления данных в любое время, даже в случае сбоев в работе или разделения на несколько частей. Soft state (мягкое состояние) – это свойство обозначает, что состояние системы может быть временно несогласованным, то есть данные могут находиться в процессе изменения и обновления. Eventually consistent (в конечном итоге согласованный) – это свойство обозначает, что в системе может быть временная

несогласованность данных, но в конечном итоге все они будут согласованы. То есть после выполнения операции изменения данных, не все копии данных в системе будут мгновенно обновлены, и в течение некоторого времени различные копии данных могут содержать разные версии.

В целом, концепция BASE делает основной упор на доступность данных и их масштабируемость, позволяя достигнуть этих целей за счет компромисса в отношении согласованности данных. Допускается, что данные могут быть не совсем точными или актуальными в некоторый момент времени. Однако, при работе системы данные обновляются и постепенно их состояние становится согласованным.

Концепция CAP определяет, что распределенная вычислительная система не может одновременно обеспечивать три свойства: согласованность, доступность и устойчивость к разделению [17]. Consistency (согласованность) – свойство, которое определяет, что каждый узел в распределенной системе должен иметь доступ к одному и тому же состоянию данных в любой момент времени. Если система обладает свойством согласованности, то любые изменения данных, которые происходят в одном узле, должны быть немедленно доступны во всех остальных узлах системы. Availability (доступность) – свойство, которое определяет, что каждый запрос, отправленный к системе, должен получить ответ, даже если произошел сбой в системе. Если система обладает свойством доступности, то каждый узел должен быть доступен для обработки запросов в любое время, даже если другие узлы системы недоступны. Partition Tolerance (устойчивость к разделению) – свойство, которое определяет, что система должна продолжать функционировать, даже если происходит ее разделение на две или более частей. Если система обладает свойством

устойчивости к разделению, то каждый узел должен продолжать работать независимо от других узлов, даже если эти узлы временно не могут связаться друг с другом.

В целом, эта концепция делает основной упор на доступность данных и их отказоустойчивость. Однако, для достижения этих целей может потребоваться отказ от согласованности данных в некоторых случаях, что приводит к возможным проблемам в процессе работы системы.

Таким образом, существующие концепции проектирования распределенных вычислительных систем позволяют учесть различные требования к создаваемой системе в зависимости от ее особенностей и условий применения. При этом возможно ориентироваться как на одну из представленных концепций, так и определить требуемое сочетание свойств разных концепций в качестве компромисса, удовлетворяющего особенностям и условиям применения системы.

Исследование существующих архитектурных подходов к построению распределенных систем. Наиболее популярными архитектурными подходами к построению распределенных вычислительных систем в настоящее время являются: клиент-серверная архитектура; многоуровневая архитектура; одноранговая (peer-to-peer) архитектура; сервис-ориентированная архитектура (SOA – Service-oriented architecture).

Клиент-серверная архитектура – это централизованная модель организации вычислительных систем, в которой задачи распределены между клиентами и серверами (рисунок 1) [18, 19].



Рисунок 1 – Клиент-серверная архитектура

Клиенты – это устройства или приложения, которые запрашивают информацию или услуги у сервера. Серверы – компьютеры, которые предоставляют запрашиваемую информацию или услуги клиентам.

Преимущества подхода: масштабируемость: система легко адаптируется под увеличение нагрузки; централизованное управление: все данные хранятся на сервере, что упрощает их контроль и защиту; безопасность: возможность централизованно управлять доступом к данным; гибкость: легко добавлять новые функции и услуги без значительных изменений в существующей системе.

Недостатки подхода: зависимость от сервера: если сервер выходит из строя, клиенты не смогут получить доступ к услугам или данным; сложность настройки: настройка и управление сервером могут требовать значительных усилий и знаний; сетевые задержки: взаимодействие между клиентом и сервером может быть замедлено из-за сетевых задержек, особенно при больших объемах данных.

Многоуровневая архитектура – это централизованная модель организации вычислительных систем, в которой сервер разделен на несколько специализированных узлов, каждый из которых выполняет определенные задачи (рисунок 2) [18, 19].

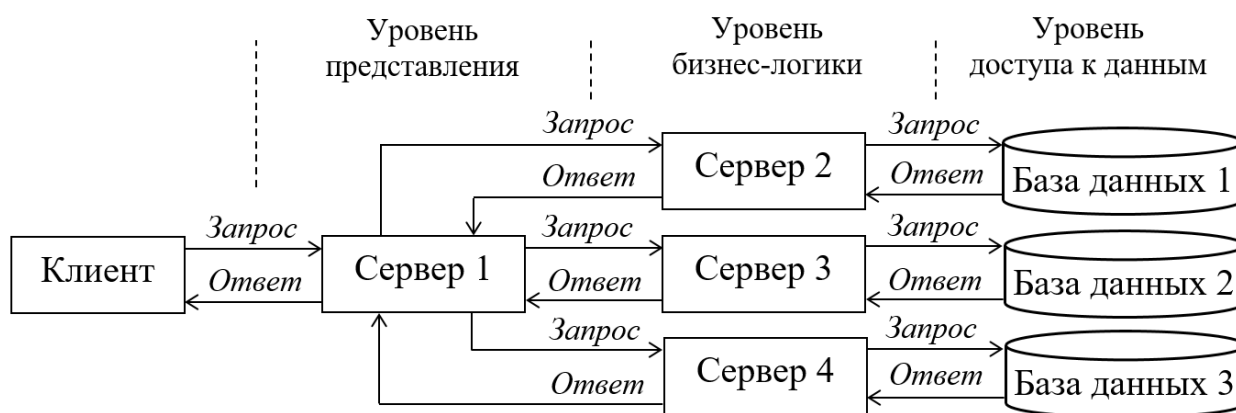


Рисунок 2 – Многоуровневая архитектура

Преимущества подхода: гибкость: возможность изменять один уровень без

необходимости изменения всей системы; безопасность: четкое разделение уровней позволяет эффективно контролировать доступ к критически важным данным и функциям; упрощение тестирования и отладки: проблемы можно локализовать в конкретном уровне; возможность параллельной разработки разных уровней командами специалистов с различной экспертизой.

Недостатки подхода: низкая скорость работы: много информации проходит от слоя к слою, не используя бизнес-логику; затрудненный поиск ошибок: прежде чем попасть в базу данных, информация проходит через все уровни, и, если что-то повреждается, приходится анализировать каждый уровень по отдельности.

Одноранговая архитектура – это децентрализованная модель организации вычислительных систем, в которой все узлы системы равноправны и обладают одинаковыми функциями, при этом каждый узел (устройство) может выступать как клиентом, так и сервером (рисунок 3) [18, 19].

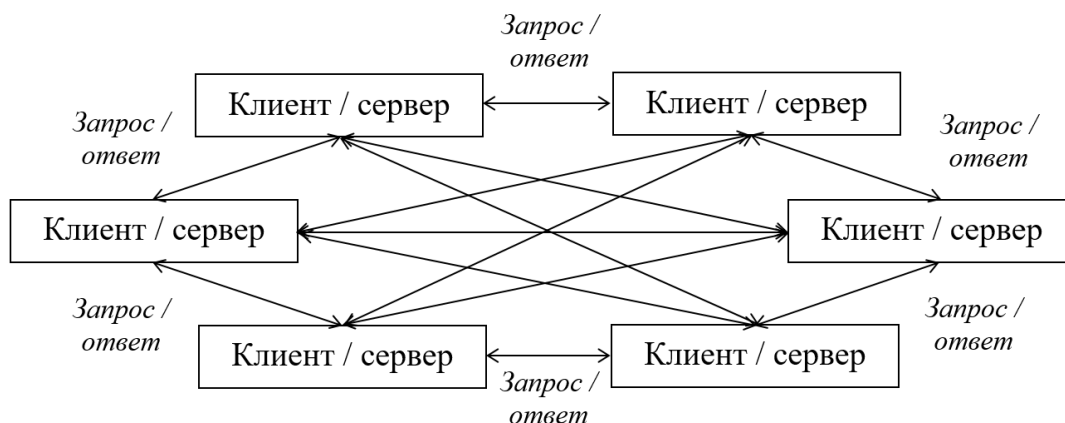


Рисунок 3 – Одноранговая архитектура

Преимущества подхода: высокая отказоустойчивость: отсутствие единой точки отказа минимизирует риск простоя системы при выходе из строя отдельных узлов; масштабируемость и распределение нагрузки: обмен информацией может происходить

одновременно с несколькими узлами; анонимность и безопасность данных: передача происходит напрямую между узлами, что обеспечивает конфиденциальность.

Недостатки подхода: сложности с безопасностью: отсутствие централизованного управления усложняет обеспечение безопасности системы; ограниченная пропускная способность: если система имеет много активных узлов, это замедляет передачу данных.

Сервис-ориентированная архитектура – это модульный подход, базирующийся на обеспечении удаленного использования по стандартизированным протоколам распределенных, слабо связанных, легко заменяемых узлов (сервисов) со стандартизированными интерфейсами (рисунок 4) [18, 19].



Рисунок 4 – Сервис-ориентированная архитектура

Преимущества подхода: автономность сервисов: каждый сервис является независимым и может функционировать без вмешательства других сервисов; переиспользуемость сервисов: сервисы могут быть использованы повторно в различных приложениях и контекстах; стандартизированные контракты: взаимодействие между сервисами осуществляется через стандартизированные интерфейсы и протоколы на общей платформе взаимодействия; композиция сервисов: объединение

нескольких сервисов в единое более приложение в интересах гибкой настройки и изменения функциональности системы без необходимости изменения отдельных сервисов; интероперабельность: сервисы могут работать на различных платформах и быть написаны на разных языках программирования; масштабируемость: архитектура позволяет легко добавлять новые экземпляры сервисов для обработки возросшей нагрузки, а также оптимизировать или улучшать отдельные сервисы без влияния на всю систему; повышение надежности: проблемы в одном сервисе не обязательно влияют на работу всей системы, а каждый сервис можно протестировать независимо.

Недостатки подхода: сложность управления: с увеличением количества сервисов усложняется управление системой; большая нагрузка: служба проверяет сообщение на соответствие заявленному заранее контракту, и, если применяется несколько сервисов, это может привести к увеличению времени отклика и снижению общей производительности.

Помимо рассмотренных подходов к разработке распределенных систем на практике также применяются следующие архитектурные модели: модель «мастер-слейв» (Master-Slave); кластерные системы.

Модель «мастер-слейв» (главный-подчиненный) – это централизованная модель организации вычислительных систем, в которой система состоит из нескольких узлов, каждый из которых имеет свою роль [16]: мастер (Master) – основной узел системы, который обрабатывает все операции записи и координирует данные, отвечает за согласование всех изменений и поддержание согласованности данных среди всех слейвов; слейвы (Slave) – вторичные узлы системы, которые реплицируют данные с мастера и отвечают за обработку операций чтения.

Ключевые особенности модели: репликация данных: мастер отправляет изменения данных всем слейвам, чтобы поддерживать их актуальность; высокая доступность: если мастер выходит из строя или недоступен, один из слейвов может быть повышен до роли мастера для обеспечения непрерывности обработки операций записи; масштабируемость на чтение: добавление дополнительных слейвов позволяет распределить нагрузку на операции чтения, увеличивая производительность системы; относительная низкая стоимость: использование слейвов для операций чтения позволяет использовать более дешевое оборудование для них.

Кластерные системы (кластеры) – это группы взаимосвязанных узлов, которые работают как единая система и решают общие задачи. Основная цель кластеризации – повышение производительности, доступности и отказоустойчивости системы [20].

Преимущества модели: высокая производительность: задачи разбиваются на подзадачи, которые выполняются одновременно на нескольких узлах; отказоустойчивость: в случае отказа одного узла система автоматически переключается на резервный, что минимизирует риски потери данных; масштабируемость: при увеличении нагрузки можно добавить новые узлы в кластер.

Недостатки модели: сложность настройки и управления: для создания и настройки кластера нужны глубокие знания операционных систем, сетей и аппаратного обеспечения; высокая стоимость создания и масштабирования: создание физического кластера может потребовать значительных затрат на приобретение дополнительного оборудования, кабелей, коммутаторов и программного обеспечения.

Таким образом, рассмотренные подходы к построению распределенных вычислительных систем и существующие архитектурные модели при их раздельном

применении или в комбинации друг с другом позволяют учесть специфику применения систем различного назначения и разработать соответствующую архитектуру конкретной распределенной системы с учетом особенностей и условий ее применения.

Решение задачи

Исходя из постановки и условий решаемой задачи, с учетом коммуникативных функций ПУ и бортовых программных систем БпЛА на различных уровнях рассматриваемой иерархической системы (см. таблицу 1), а также результатов исследования существующих концепций проектирования распределенных систем, определим требования к организации взаимодействия бортовых программных систем БпЛА в группе и с ПУ в условиях нестабильного информационного обмена.

1. Требования к организации взаимодействия между низшим и средним уровнями системы в направлении обмена «низший уровень – средний уровень» должны определяться свойствами концепции BASE: Basically Available (в основном доступный) – вне зависимости от наличия связи разведывательного БпЛА-не лидера (или транспортного БпЛА) с разведывательным БпЛА-лидером в текущий момент времени первый должен выдать информацию на борт последнего; при нарушении связи это реализуется посредством применения алгоритма маршрутизации обмена информацией, использующего доступные для взаимодействия в текущий момент времени БпЛА в группе; Soft state (мягкое состояние) – вследствие возникновения сбоев в связи изменяется количество БпЛА, участвующих в информационном обмене в различные моменты времени, что обуславливает несогласованность данных, поэтому информация должна обновляться с установленным для системы темпом;

Eventually consistent (в конечном итоге согласованный) – на борту разведывательного БпЛА-лидера должна производиться обработка (для достижения согласованности полученных данных) и обобщение информации за группу БпЛА.

Кроме того, при проведении поисково-спасательной операции в зависимости от ряда условий (например, площади района поиска, выделенного на операцию ресурса и т.п.) может применяться различное количество N разведывательных и K транспортных БпЛА. То есть должно выполняться требование масштабируемости, которое наряду с требованием доступности обеспечивает концепция BASE.

2. Требования к организации взаимодействия между низшим и средним уровнями системы в направлении обмена «средний уровень – низший уровень» должны определяться свойствами концепции ACID: Atomicity (атомарность) – команды управления и служебная информация с борта разведывательного БпЛА-лидера должны выдаваться разведывательным БпЛА-не лидерам и транспортным БпЛА без искажений или не выдаваться совсем; Consistency (согласованность) – средний и низший уровни системы должны находиться в согласованном состоянии как до выдачи команд управления и служебной информации разведывательным БпЛА-лидером, так и после. То есть все разведывательные и транспортные БпЛА должны однозначно интерпретировать одну и ту же команду или служебную информацию от БпЛА-лидера в один и тот же момент времени; Isolation (изолированность) – выдача одной команды управления или служебной информации разведывательным БпЛА-лидером не должна влиять на выдачу другой команды управления или служебной информации; Durability (долговечность) – после успешной выдачи команды управления или служебной информации разведывательным БпЛА-

лидером изменения данных должны быть сохранены даже в случае сбоя в информационном обмене.

Выдача команд и служебной информации разведывательным БПЛА-лидером не подразумевает работу с большими объемами данных, что не снижает производительности системы. А так как основное внимание уделяется надежности и целостности данных, то применение концепции ACID в данном случае вполне оправдано.

3. Требования к организации взаимодействия между средним и высшим уровнями системы в обоих направлениях обмена должны определяться свойствами концепции ACID: Atomicity (атомарность) – данные, участвующие в информационном обмене между ПУ и разведывательным БПЛА-лидером, должны быть не искажены или не участвовать в обмене совсем; Consistency (согласованность) – средний и высший уровни системы должны находиться в согласованном состоянии в процессе взаимного обмена данными, то есть ПУ и разведывательный БПЛА-лидер должны однозначно интерпретировать одну и ту же информацию в один и тот же момент времени; Isolation (изолированность) – данные, участвующие в информационном обмене между ПУ и разведывательным БПЛА-лидером, не должны влиять друг на друга; Durability (долговечность) – изменения данных, участвующих в информационном обмене между ПУ и разведывательным БПЛА-лидером, должны быть сохранены даже в случае его сбоя.

В рассматриваемом случае обработка данных производится на высшем уровне рассматриваемой системы – ПУ, имеющем достаточно мощные вычислительные ресурсы. Следовательно, фактор работы с большими объемами данных не является критичным и не влияет на производительность системы, в целом. При этом основное

внимание уделяется надежности и целостности данных, то есть применение концепции ACID, как и в предыдущем случае, оправдано.

Таким образом, требования к организации взаимодействия бортовых программных систем БПЛА в группе и с ПУ в условиях нестабильного информационного обмена определяются свойствами концепции BASE – между низшим и средним уровнями системы в направлении обмена «низший уровень – средний уровень»; и свойствами концепции ACID – между низшим и средним уровнями системы в направлении обмена «средний уровень – низший уровень», а также между средним и высшим уровнями системы в обоих направлениях обмена.

Для удовлетворения обоснованным требованиям к организации взаимодействия бортовых программных систем БПЛА в группе и с ПУ в условиях нестабильного информационного обмена соответствующую ей архитектуру необходимо рассматривать как сервис-ориентированную, с кластерным разбиением групп разведывательных и транспортных БПЛА согласно реализуемому функционалу и одноранговым представлением взаимодействия узлов внутри кластера разведывательных БПЛА-не лидеров.

Предлагаемая архитектура организации взаимодействия бортовых программных систем БПЛА в группе и с ПУ в условиях нестабильного информационного обмена проиллюстрирована на рисунке 5.

Кластер управления высшего уровня системы состоит из одного узла – ПУ. Однако, данный кластер может состоять из нескольких узлов – объединенных ПУ. Это определяется масштабом общей задачи, решаемой системой.

Посредством *Стандартизированного интерфейса 1*, реализованного с учетом требований, обусловленных свойствами концепции ACID, осуществляется взаимный

обмен информацией (команды управления, служебная информация, обобщенная видовая информация) с *Кластером управления среднего уровня системы* – разведывательным БпЛА-лидером группы.

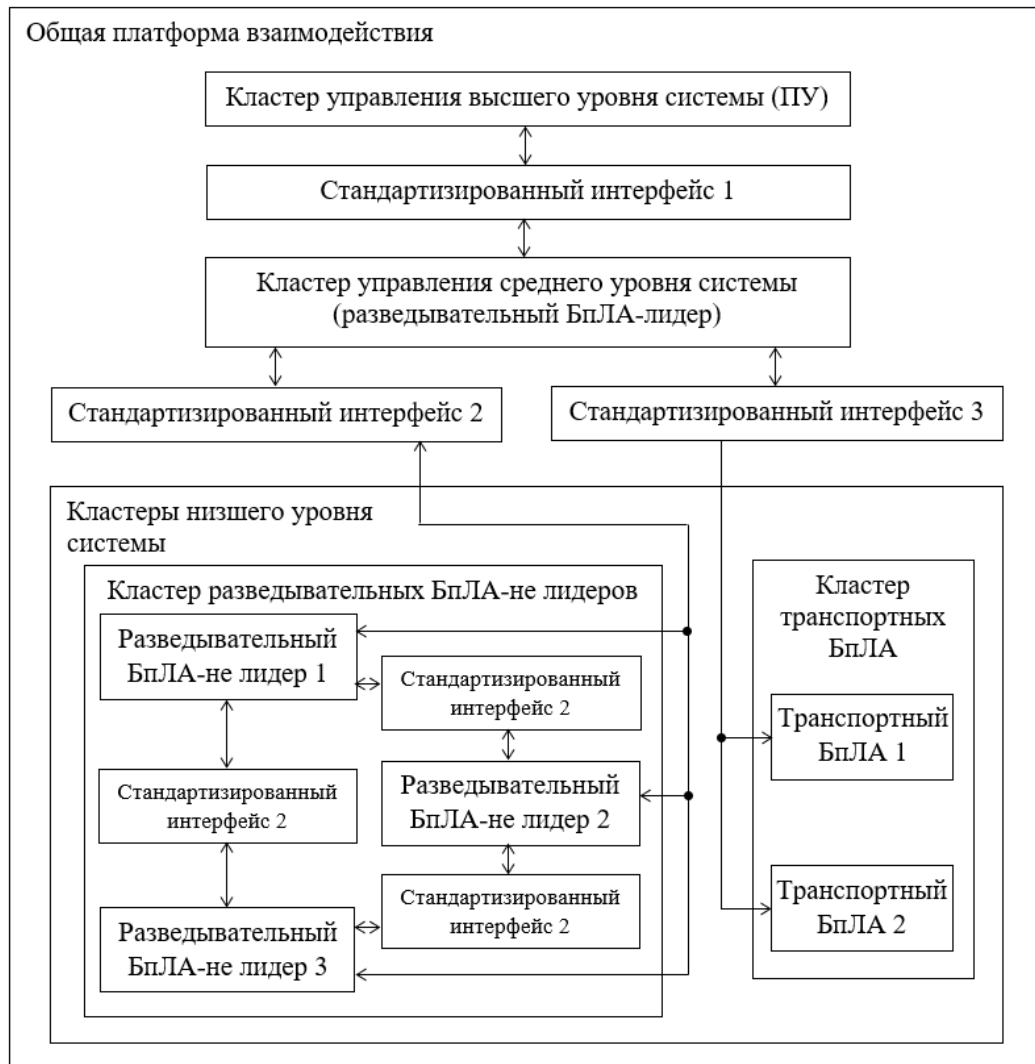


Рисунок 5 – Архитектура организации взаимодействия бортовых программных систем БпЛА в группе и с ПУ в условиях нестабильного информационного обмена

Кластер управления среднего уровня системы может состоять из нескольких БпЛА-лидеров групп. Это определяется количеством групп, которое обусловлено как масштабом решаемой задачи, так и числом, а также удаленностью друг от друга

участков поиска. Роль лидера является выборной, выборы запускаются с использованием соответствующего алгоритма выбора лидера в тот момент времени, когда установленный лидер не способен выполнять требуемый функционал в полном объеме [21].

При назначении нового БпЛА-лидера группы осуществляется переход узла – прежнего БпЛА-не лидера группы из *Кластера разведывательных БпЛА-не лидеров* в *Кластер управления среднего уровня системы*. В свою очередь, прежний лидер переходит из *Кластера управления среднего уровня системы* в *Кластер разведывательных БпЛА-не лидеров*.

Так как потенциально любой разведывательный БпЛА-не лидер может стать лидером группы, то взаимодействие между разведывательными БпЛА вне зависимости от их роли в группе осуществляется посредством *Стандартизированного интерфейса 2*. Данный интерфейс реализован с учетом требований, обусловленных свойствами концепции BASE и концепции ACID. Кроме того, *Кластер управления среднего уровня системы* взаимодействует с *Кластером транспортных БпЛА* посредством *Стандартизированного интерфейса 3*, реализованного с учетом требований, обусловленных свойствами концепции BASE и концепции ACID. Отличием *Стандартизированного интерфейса 3* от *Стандартизированного интерфейса 2* является отсутствие необходимости реализации обмена видовой информацией, присущей разведывательным БпЛА.

Таким образом, с учетом условий решаемой задачи, функций ПУ и бортовых программных систем БпЛА на различных уровнях рассматриваемой иерархической системы (см. таблицу 1), а также обоснованных требований к организации взаимодействия разработана архитектура организации взаимодействия бортовых программных систем БпЛА в группе и с ПУ в условиях нестабильного информационного обмена.

Выводы

В результате проведенных в статье исследований разработана архитектура организации взаимодействия бортовых программных систем БпЛА в группе и с ПУ, основанная на их представлении в виде элементов (узлов) распределенной иерархической системы доминантного типа, учитывающая целевые и коммуникативные функции этих элементов (узлов) на различных уровнях рассматриваемой иерархической системы, условия и особенности применения группы БпЛА под управлением ПУ в поисково-спасательной операции в горной местности, а также обоснованные требования к организации взаимодействия (между низшим и средним уровнями системы в направлении обмена «низший уровень – средний уровень» требования определяются свойствами концепции BASE; между низшим и средним уровнями системы в направлении обмена «средний уровень – низший уровень», а также между средним и высшим уровнями системы в обоих направлениях обмена – свойствами концепции ACID), отличающаяся тем, что представляет собой сервис-ориентированную архитектуру с кластерным разбиением групп разведывательных и транспортных БпЛА согласно реализуемому функционалу и одноранговым представлением взаимодействия узлов внутри кластера разведывательных БпЛА-лидеров, что позволяет обеспечить организацию устойчивого взаимодействия бортовых программных систем БпЛА в группе и с ПУ в условиях нестабильного информационного обмена при отсутствии БпЛА-ретрансляторов.

Практическая значимость заключается в возможности реализации разработанной архитектуры с использованием бортовых программных систем современных БпЛА без необходимости модернизации их аппаратной части.

Список источников

1. Меркулов В.И., Миляков Д.А., Самодов И.О. Оптимизация алгоритма группового управления БЛА в составе локальной сети // Известия ЮФУ. Технические науки. 2013. № 3. С. 157–166.
2. Меркулов В.И., Харьков В.П., Шамаров Н.Н. Оптимизация коллективного управления группой БЛА // Информационно-измерительные и управляющие системы. 2012. № 7. С. 3–8.
3. Меркулов В.И., Харьков В.П. Оптимизация иерархического управления группой БЛА // Информационно-измерительные и управляющие системы. 2012. № 8. С. 61–67.
4. Ким Н.В., Крылов И.Г. Групповое применение беспилотного летательного аппарата в задачах наблюдения // Труды МАИ. 2012. № 62. URL: <https://trudymai.ru/published.php?ID=35507>
5. Моисеев Д.В., Чинь Ван Минь, Мозолев Л.А., Моисеева С.Г., Фам Суан Куен. Маршрутизация полета легкого беспилотного летательного аппарата в поле постоянного ветра на основе решения разновидностей задачи коммивояжера // Труды МАИ. 2015. № 79. URL: <https://trudymai.ru/published.php?ID=55782>
6. Моисеев Г.В., Моисеев В.С. Основы теории создания и применения имитационных беспилотных авиационных комплексов: монография. – Казань: РЦ МКО, 2013. – 207 с.
7. Моисеев В.С. Прикладная теория управления беспилотными летательными аппаратами: монография. – Казань: РЦ МКО, 2013. – 768 с.
8. Моисеев В.С. Групповое применение беспилотных летательных аппаратов: монография. – Казань: РИЦ «Школа», 2017. – 572 с.

9. Подлипьян П.Е., Максимов Н.А. Многофазный алгоритм решения задачи планирования полета группы беспилотных летательных аппаратов // Труды МАИ. 2011. № 43. URL: <https://trudymai.ru/published.php?ID=24769>
10. Ананьев А.В., Стафеев М.А., Макеев Е.В. Разработка способа организации связи с использованием беспилотных летательных аппаратов малой дальности // Труды МАИ. 2019. № 105. URL: <https://trudymai.ru/published.php?ID=104223>
11. Ананьев А.В., Иванников К.С., Филатов С.В. Основные принципы построения систем связи на базе беспилотных летательных аппаратов // Труды МАИ. 2022. № 125. URL: <https://trudymai.ru/published.php?ID=168188>. DOI: [10.34759/trd-2022-125-16](https://doi.org/10.34759/trd-2022-125-16)
12. Godwin M.F., Spry S.C., Hedrick J.K. A Distributed System for Collaboration and Control of UAV Groups: Experiments and Analysis. Center for the Collaborative Control of Unmanned Vehicles University of California, Berkeley, 2007. 224 p. DOI: [10.1007/978-3-540-48271-0_9](https://doi.org/10.1007/978-3-540-48271-0_9)
13. Fahlstrom P.G., Gleason T.J. Introduction to UAV Systems. John Wiley & Sons, Ltd. 2012. 287 p.
14. Müller W., Reinert F., Pallmer D. Open architecture of a counter UAV system // Preceding SPIE 10651, Open Architecture/Open Business Model Net-Centric Systems and Defense Transformation. 2018. DOI: [10.1117/12.2305606](https://doi.org/10.1117/12.2305606)
15. Bernstein P.A., Goodman N., Hadzilacos V. Concurrency Control and Recovery in Database Systems. Addison Wesley Publishing Company, 1987. 370 p.
16. Цимбал А.А., Аншина М.Л. Технологии создания распределенных систем. Для профессионалов. – СПб: Питер, 2003. – 576 с.

17. Brewer E.A. A Certain Freedom: Thoughts on the CAP Theorem // Proceeding of the XXIX ACM SIGACT-SIGOPS symposium on Principles of distributed computing. New York: ACM. 2010. Iss. 29, No. 1. P. 335–336. DOI: [10.1145/1835698.1835701](https://doi.org/10.1145/1835698.1835701)
18. Таненбаум Э.С., Ван Стеен М. Распределенные системы. Принципы и парадигмы. – СПб.: Питер, 2003. – 877 с.
19. Радченко Г.И. Распределенные вычислительные системы. – Челябинск: Фотохудожник, 2012. – 184 с.
20. Комаров Д.А., Кислицын Е.В. Анализ больших данных: подходы, методы, алгоритмы // IV Международная научно-практическая конференция «Инфокоммуникационные технологии: актуальные вопросы цифровой экономики» (Екатеринбург, 30–31 января 2024): сборник трудов. – Екатеринбург: Сибирский государственный университет телекоммуникаций и информатики, 2024. С. 162–168.
21. Хорсик И.А., Митрофанов Д.В., Шатовкин Р.Р. Алгоритм выбора лидера в группе беспилотных летательных аппаратов при распределенном поиске наземных объектов в условиях нарушения связи // Воздушно-космические силы. Теория и практика. 2025. № 33. С. 139–152.

Reference

1. Merkulov V.I., Milyakov D.A., Samodov I.O. Optimization of the algorithm for group control of UAVs in a local network. *Izvestiya YuFU. Tekhnicheskie nauki*. 2013. No. 3. P. 157–166. (In Russ.)
2. Merkulov V.I., Kharkov V.P., Shamarov N.N. Optimization of collective control of a group of UAVs. *Informatsionno-izmeritel'nye i upravlyayushchie sistemy*. 2012. No. 7. P. 3–8. (In Russ.)

3. Merkulov V.I., Kharkov V.P. Optimization of hierarchical control of a group of UAVs. *Informatsionno-izmeritel'nye i upravlyayushchie sistemy*. 2012. No. 8. P. 61–67. (In Russ.)
4. Kim N.V., Krylov I.G. Using a group of unmanned aerial vehicle in the task of monitoring. *Trudy MAI*. 2012. No. 62. (In Russ.). URL: <https://trudymai.ru/eng/published.php?ID=35507>
5. Moiseev D.V., Chin' Van Min', Mozolev L.A., Moiseeva S.G., Fam Suan Kuen. Light unmanned aerial vehicle flight routing in the fixed wind field on the basis of a solution of travelling salesman problem variants. *Trudy MAI*. 2015. No. 79. (In Russ.). URL: <https://trudymai.ru/eng/published.php?ID=55782>
6. Moiseev G.V., Moiseev V.S. *Osnovy teorii sozdaniya i primeneniya imitatsionnykh bespilotnykh aviatsionnykh kompleksov: monografiya* (Fundamentals of the Theory of Creation and Application of Simulation Unmanned Aircraft Systems: monograph). Kazan: RC MKO Publ., 2013. 207 p.
7. Moiseev V.S. *Prikladnaya teoriya upravleniya bespilotnymi letatel'nyimi apparatami: monografiya* (Applied Theory of Control of Unmanned Aerial Vehicles: monograph). Kazan: RC MKO Publ., 2013. 768 p.
8. Moiseev V.S. *Gruppovoe primeneniye bespilotnykh letatel'nykh apparatov: monografiya* (Group use of unmanned aerial vehicles: monograph). Kazan: RIC "School" Publ., 2017. 572 p.
9. Podlip'yan P.E., Maksimov N.A. Multi-phase algorithm for solving the problem of planning the flight of unmanned aerial vehicles. *Trudy MAI*. 2011. No. 43. (In Russ.). URL: <https://trudymai.ru/eng/published.php?ID=24769>

10. Anan'ev A.V., Stafeev M.A., Makeev E.V. Developing communication organization method employing short-range unmanned flying vehicles. *Trudy MAI*. 2019. No. 105. (In Russ.). URL: <https://trudymai.ru/eng/published.php?ID=104223>
11. Anan'ev A.V., Ivannikov K.S., Filatov S.V. Construction basic principles of communication systems based on unmanned aerial vehicles. *Trudy MAI*. 2022. No. 125. (In Russ.). URL: <https://trudymai.ru/eng/published.php?ID=168188>. DOI: [10.34759/trd-2022-125-16](https://doi.org/10.34759/trd-2022-125-16)
12. Godwin M.F., Spry S.C., Hedrick J.K. *A Distributed System for Collaboration and Control of UAV Groups: Experiments and Analysis*. Center for the Collaborative Control of Unmanned Vehicles University of California, Berkeley, 2007. 224 p. DOI: [10.1007/978-3-540-48271-0_9](https://doi.org/10.1007/978-3-540-48271-0_9)
13. Fahlstrom P.G., Gleason T.J. *Introduction to UAV Systems*. John Wiley & Sons, Ltd. 2012. 287 p.
14. Müller W., Reinert F., Pallmer D. Open architecture of a counter UAV system. *Preceding SPIE 10651, Open Architecture/Open Business Model Net-Centric Systems and Defense Transformation*. 2018. DOI: [10.1117/12.2305606](https://doi.org/10.1117/12.2305606)
15. Bernstein P.A., Goodman N., Hadzilacos V. *Concurrency Control and Recovery in Database Systems*. Addison Wesley Publishing Company, 1987. 370 p.
16. Tsimbal A.A., Anshina M.L. *Tekhnologii sozdaniya raspredelennykh sistem. Dlya professionalov* (Technologies for creating distributed systems. For professionals). Saint Petersburg: Piter Publ., 2003. 576 p.
17. Brewer E.A. *A Certain Freedom: Thoughts on the CAP Theorem // Proceeding of the XXIX ACM SIGACT-SIGOPS symposium on Principles of distributed computing*. New York: ACM. 2010. Iss. 29, No. 1. P. 335–336. DOI: [10.1145/1835698.1835701](https://doi.org/10.1145/1835698.1835701)

18. Tanenbaum E.C., Van Steen M. *Raspredeleennye sistemy. Printsipy i paradigmy* (Distributed Systems. Principles and Paradigms). Saint Petersburg: Piter Publ., 2003. 877 p.
19. Radchenko G.I. *Raspredeleennye vychislitel'nye sistemy* (Distributed computing systems). Chelyabinsk: Fotokhudozhnik Publ., 2012. 184 p.
20. Komarov D.A., Kislitsyn E.V. Big data analysis: approaches, methods, algorithms. *IV Mezhdunarodnaya nauchno-prakticheskaya konferentsiya «Infokommunikatsionnye tekhnologii: aktual'nye voprosy tsifrovoi ekonomiki»*: sbornik trudov. Ekaterinburg: Sibirskii gosudarstvennyi universitet telekommunikatsii i informatiki Publ., 2024. P. 162–168.
21. Khorsik I.A., Mitrofanov D.V., Shatovkin R.R. Leader selection algorithm in a group of unmanned aerial vehicles at the distributed search of the ground objects in the conditions of communication disturbance. *Vozdushno-kosmicheskie sily. Teoriya i praktika*. 2025. No. 33. P. 139–152. (In Russ.)

Статья поступила в редакцию 27.06.2025

Одобрена после рецензирования 02.07.2025

Принята к публикации 25.08.2025

The article was submitted on 27.06.2025; approved after reviewing on 02.07.2025; accepted for publication on 25.08.2025