

## Системы автоматической генерации программ мониторинга

А.И. Водяхо<sup>1</sup>, Е.Л. Евневич<sup>2</sup>, Н.А. Жукова<sup>2</sup>, Н.В. Климов,<sup>3</sup> М.А. Червонцев<sup>1</sup>

<sup>1</sup>Санкт-Петербургский государственный электротехнический университет, г. Санкт-Петербург, Российская Федерация

<sup>2</sup>Санкт-Петербургский институт информатики и автоматизации Российской академии наук, г. Санкт-Петербург, Российская Федерация

<sup>3</sup>Санкт-Петербургский национальный исследовательский университет информационных технологий механики и оптики, г. Санкт-Петербург, Российская Федерация

<sup>1</sup>Национальный исследовательский Мордовский государственный университет  
<sup>2</sup>Вятский государственный университет

**Аннотация:** Предлагаются модели систем генерации программ мониторинга, позволяющих существенно сократить затраты временных и иных ресурсов на мониторинг сложных технических объектов. Системы рассматриваются на нескольких уровнях: подсистем и модулей, коммуникационных связей, сценариев использования, физического представления. Предлагается методика адаптации моделей систем к предметным областям.

**Ключевые слова:** сложный технический объект, мониторинг, модели программных систем, генерация программ.

### Введение

В процессе развития систем мониторинга существенно увеличилось число реализуемых ими функций, они стали применяться во всё большем количестве предметных областей. На сегодняшний день системы мониторинга используются практически повсеместно: в сфере медицины [1], экологии и природопользования [2], социальной сфере [3], сфере энергопотребления [4], производства [5], городского хозяйства [6], сельского хозяйства [7], рыболовства [8], умного дома [9, 10, 11] и т. д.

Объекты мониторинга, как правило, представляют собой сложные технические объекты. Наблюдение за ними ведется с применением установленных на объектах датчиков, возможно использование внешних систем наблюдения. Получаемые об объектах данные представляют собой многомерные временные ряды результатов измерений их параметров.

Мониторинг проводится исходя из целей, поставленных конечными потребителями, и требований, которые предъявляются к результатам мониторинга. Мониторинг предусматривает сбор данных о наблюдаемых объектах и построение на основе этих данных моделей объектов. Возможно построение структурно-логических, функциональных, поведенческих и других моделей. Модели описывают объекты через их элементы, взаимосвязи между элементами, изменения элементов и взаимосвязей в пространстве и во времени. При построении моделей могут применяться статистические методы обработки данных, алгоритмы интеллектуального анализа и машинного обучения [9].

В настоящее время проведение мониторинга предусматривает активное участие экспертов. Необходимость привлечения экспертов приводит к существенным временным затратам и затратам иных ресурсов. Необходимо наличие программных систем, способных автоматически генерировать программы мониторинга.

В настоящее время разработаны формальные модели, методы и алгоритмы, позволяющие автоматизировать мониторинг за счет автоматического построения процессов сбора данных, а также моделей объектов мониторинга по поступающим от них данным [12, 13]. В статье рассматриваются модели систем автоматической генерации программ мониторинга, позволяющие реализовать предложенные формальные модели и методы на практике.

### **Обобщённая модель систем автоматической генерации программ мониторинга**

На вход системы поступают наборы данных об объекте наблюдения. Примерами данных являются:

- дискретные гетерогенные показания низкоуровневых датчиков, характеризующие состояния объектов (температура, влажность, напряжение и пр.);
- данные об объектах, предоставляемые другими источниками; данные могут быть представлены в текстовом формате, в виде изображений, звуковых сигналов и в других формах;
- данные о возможных зависимостях в поведении параметров объектов, состоянии их отдельных систем и подсистем, а также других взаимосвязях, свойственных объектам;
- информационные сообщения, оповещающие об отдельных происходящих на объектах событиях, которые могут привести к существенным изменениям в их состоянии и поведении.

Для передачи наборов данных с объектов или от других источников, они представляются в виде информационных потоков. Содержание информационных потоков носит название контента. Контент характеризуется такими свойствами, как полнота, релевантность, качество, степень формализма и пр. Обработка контента информационных потоков осуществляется в контексте, который определяют факторы, оказывающие влияние на проводимый мониторинг и его результаты [14].

В прикладных предметных областях могут использоваться различные подходы к сбору и передаче данных, возможны различные варианты реализации программных систем. Для того чтобы абстрагироваться от особенностей предметных областей в системе генерации программ мониторинга (СГПМ) присутствует интеграционный компонент – **подсистема интеграции**, включающая драйвера-адаптеры. Подсистема интеграции обеспечивает возможность взаимодействия с наблюдаемыми объектами. К основным задачам, решаемым подсистемой, относятся:

---

- приведение гетерогенных информационных потоков, поступающих от наблюдаемых объектов, к стандартизованному формату, используемому в СГПМ;
- передача наблюдаемым объектам управляющих команд, позволяющих осуществлять сбор данных при мониторинге.

Для работы с данными, передаваемыми с объектов, предусмотрена **подсистема получения и обработки данных (шина данных)**. Подсистема выступает в роли единого брокера циркулирующих в системе данных. После обработки данных осуществляется их перенаправление в подсистему накопления и хранения данных, а также в подсистему генерации моделей наблюдаемых объектов и в подсистему генерации процессов сбора данных об объектах мониторинга.

**Подсистема генерации моделей объектов** позволяет строить модели на основе получаемых от объектов данных. Согласно методам, предложенным в [13], генерация моделей осуществляется с применением индуктивно-дедуктивного подхода. На первом шаге для построения моделей применяются индуктивные методы. Индуктивный синтез моделей объектов выполняется по фрагментам исходного потока данных. Из каждого фрагмента выделяются отдельные элементы, которые затем связываются между собой, образуя в результате многоуровневые структуры. На верхних уровнях этих структур размещаются элементы, характеризующие состояния объектов в целом, на нижнем – отдельных их элементов (датчиков, узлов, агрегатов и пр.). Такие модели носят название локальных моделей объектов [15]. Состав и количество уровней локальных моделей может быть различным и определяется имеющимися об объекте данными. Выбор алгоритмов обработки осуществляется с учетом свойств данных. Возможно применение корреляционных алгоритмов, алгоритмов кластерного анализа и других.

---

Второй шаг генерации моделей объектов предусматривает построение моделей, характеризующих изменения, происходящие в объектах с течением времени. В соответствии с методом, рассмотренном в [13], по фрагментам потока данных строится последовательность локальных моделей. Локальные модели связываются в единую глобальную модель с применением дедуктивных методов.

**Подсистема генерации процессов сбора данных** реализует метод, приведенный в [14, 15]. Методом предусматривается, что имеется модель, построенная по полученным от объекта данным, а также задана модель, которую необходимо построить. Первая модель носит название текущей модели, вторая – требуемой. Она задается на основе требований конечных пользователей. В качестве требуемых моделей могут рассматриваться модели, соответствующие объектам, находящимся в исправном состоянии, или, наоборот, модели, описывающие неисправности, возникающие на объектах. При генерации процессов сбора данных определяется последовательность переходов, которая может позволить построить требуемую модель на основе текущей за счет получения дополнительных данных и формирование на их основе недостающих узлов и связей. Например, при представлении моделей в виде графов для построения последовательностей переходов оцениваются расстояния между графовыми представлениями текущих и требуемых моделей, определяются способы редактирования текущих моделей для их преобразования в требуемые.

**Подсистема взаимодействия с оператором** представляет собой пользовательский интерфейс. Типовым вариантом реализации интерфейса является информационная панель, в которой отображается информация о выявленных неисправностях, о возможных способах их устранения и пр.

## Модели систем генерации программ мониторинга

Системы генерации программ мониторинга можно представить в виде совокупности моделей, каждая из которых является представлением некоторого определенного набора аспектов, интересующих субъектов разработки: пользователей, проектировщиков, менеджеров, технических специалистов, обслуживающий персонал и т. д.

Модели охватывают основные решения, принятые при выборе структур систем, и описывают их компоненты. Каждая из моделей описывает системы с различных точек зрения. Можно выделить следующие уровни описания систем:

1. Уровень подсистем и модулей - определяет общую логическую схему систем, состав имеющихся компонент и логических связей между ними.
2. Уровень коммуникационных связей - содержит описание задач (процессов и нитей), их взаимодействия и конфигурации, а также взаимосвязи с системными объектами и задачами.
3. Уровень сценариев использования, в состав которых входят варианты использования, описывающие поведение систем.
4. Уровень физического представления или представления развертывания - содержит информацию о схеме развёртывания систем.

Представление системы автоматической генерации программ мониторинга в виде подсистем и их логических связей представлено на рис. 1.

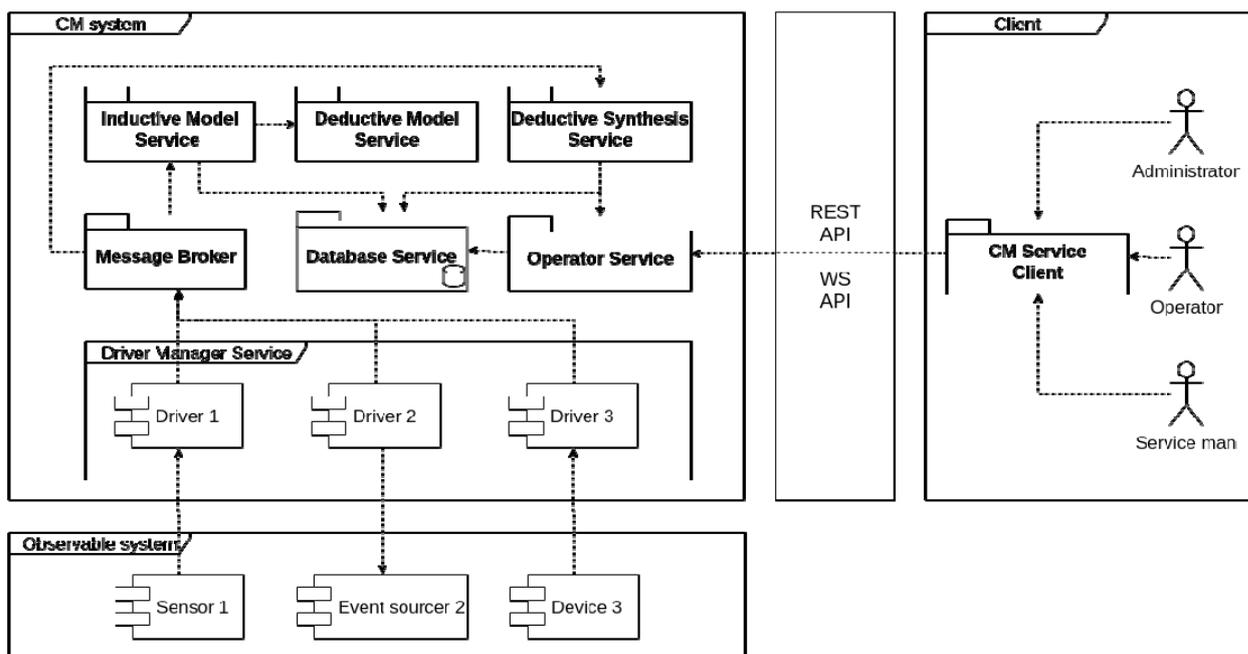


Рис 1. – Структурная схема системы генерации программ мониторинга  
Основными участниками информационного взаимодействия являются три системы:

1. Наблюдаемый объект и входящие в него элементы.
2. Система генерации программ мониторинга.
3. Система взаимодействия с оператором.

## 2 Уровень коммуникационных связей

Представление систем в виде подсистем и их прикладных интерфейсов программирования приведено на рисунке 2.

Основным связывающим элементом между подсистемами является шина данных (**Message Broker Service**). Она обеспечивает своевременную доставку входных данных и последующую коммуникацию между подсистемами посредством использования шаблона “Издатель-Подписчик”.

Драйверы-адаптеры внешних систем (**Driver Service**) выступают в качестве источника данных для шины данных и всех потребителей информации о наблюдаемом объекте.

Для некоторых задач системы необходимо сохранять данные в подсистеме базы данных, а также своевременно получать доступ к ним из других

подсистем. Для этих задач подсистема базы данных (Database Service) предоставляет сервис работы с данными, позволяющий обрабатывать внешние запросы на сохранение и получение данных по унифицированному протоколу.

Сервис для обработки запросов входит в состав подсистемы взаимодействия с оператором (Operator Service). посредством данного сервиса клиентская часть подсистемы, с которой оператор работает непосредственно, имеет возможность получения уведомлений об изменении текущего состояния наблюдаемого объекта. Также сервис выступает в роли посредника на этапе формирования управляющего воздействия на наблюдаемый объект.

Подсистемы генерации моделей объектов и процессов сбора данных являются лишь источниками данных, инициативно сохраняя необходимые данные в подсистеме базы данных или передавая их дальше посредством шины данных.

### **3 Уровень сценариев использования**

Для систем генерации программ мониторинга можно выделить три основных сценария:

1. Получение потоков данных от наблюдаемого объекта и генерация локальных моделей объекта мониторинга.
2. Генерация и хранение глобальной модели объекта мониторинга на основе локальных моделей.
3. Генерация процессов сбора данных об объекте с целью определения возникающих на нем неисправностей.

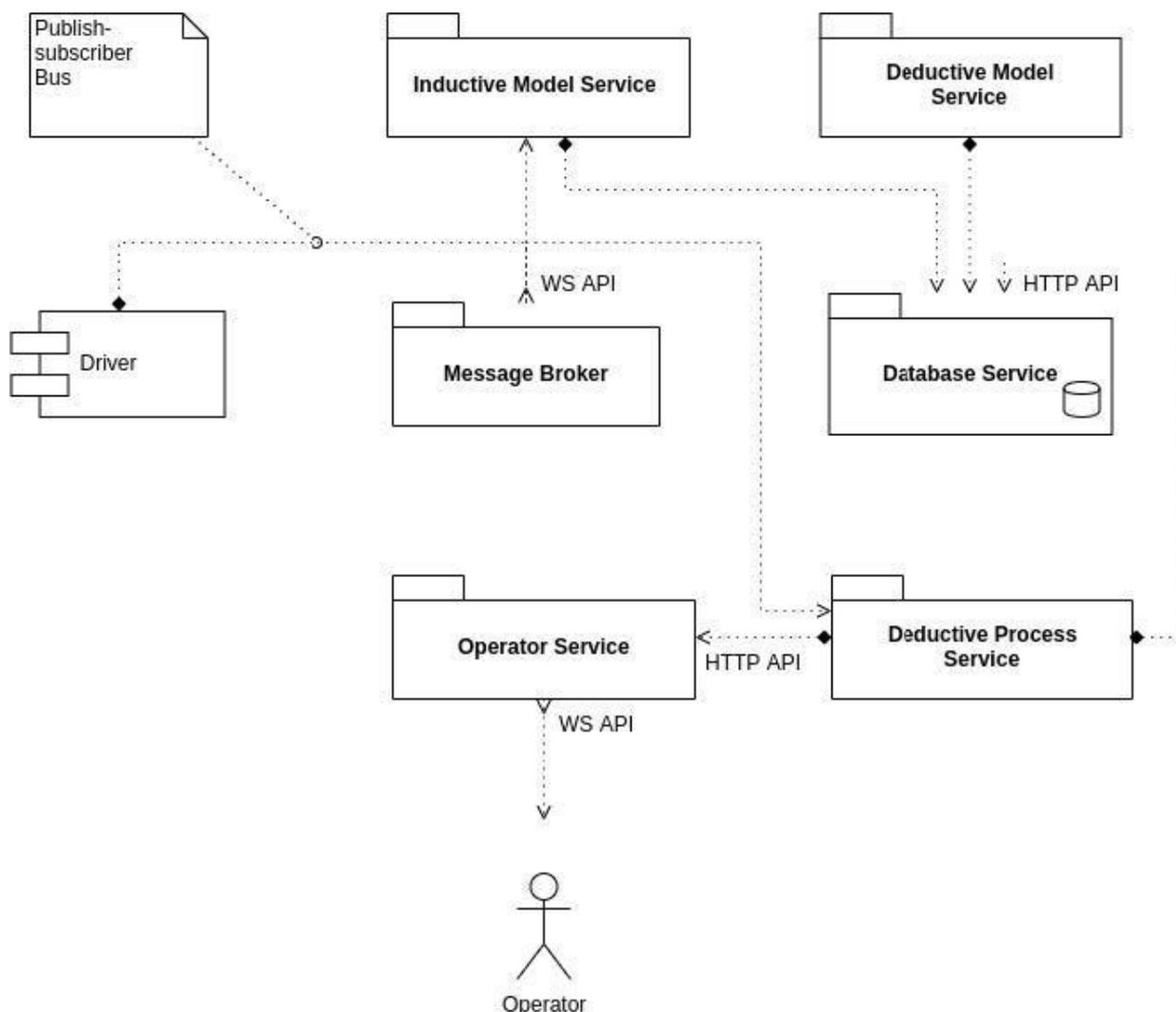


Рис 2. – Прикладные интерфейсы программирования в системе генерации программ мониторинга

### 3.1 Генерация локальных моделей наблюдаемого объекта

Диаграмма последовательности генерации локальных моделей наблюдаемого объекта изображена на рисунке 3. С наблюдаемого объекта поступает гетерогенный поток данных, содержащий информацию о состоянии объекта и его поведении.

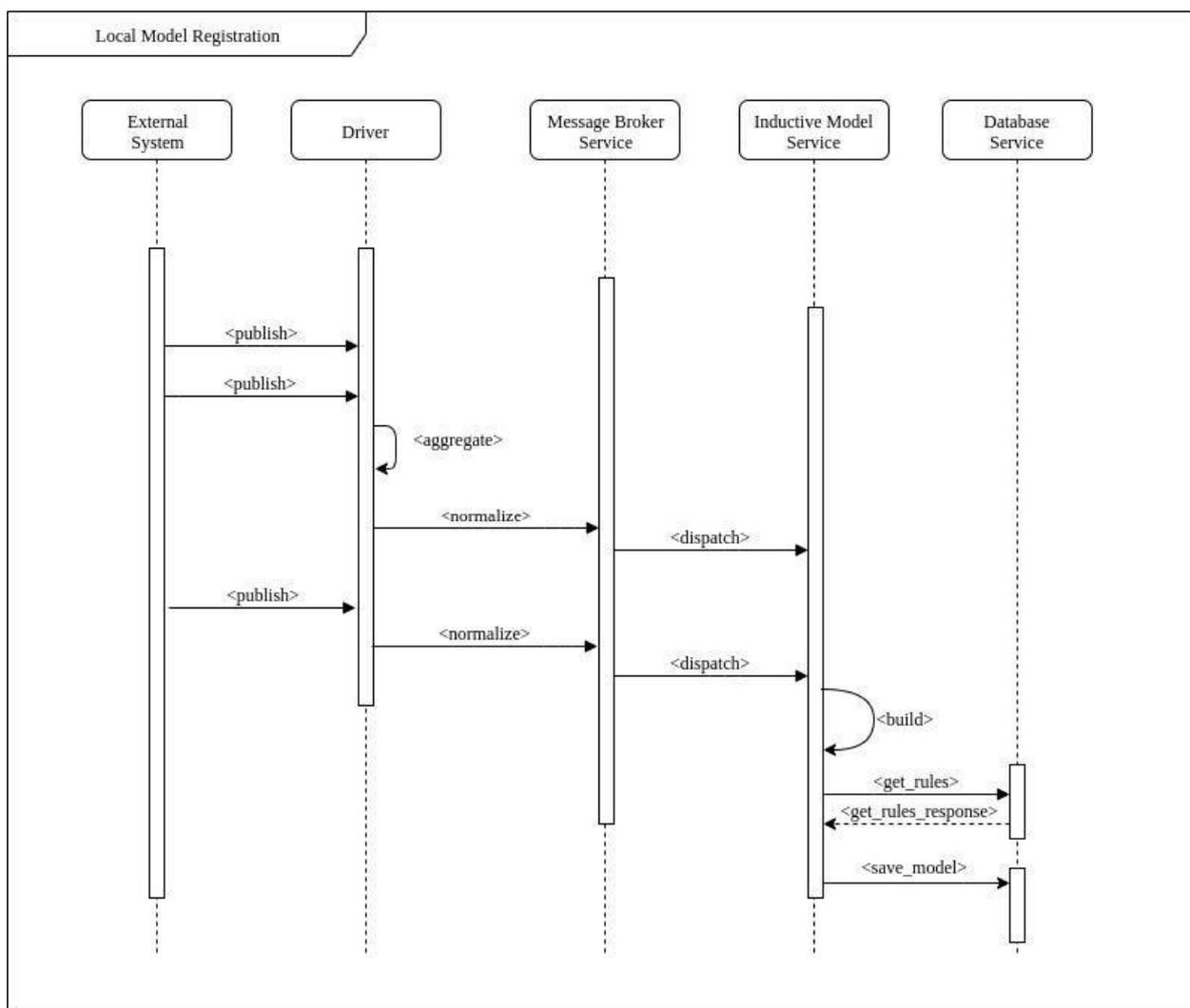


Рис 3. – Сценарий генерации локальных моделей наблюдаемого объекта  
Сбор данных осуществляется при помощи специализированных драйверов-адаптеров: проактивно (посредством запроса драйвером данных у наблюдаемого объекта) либо пассивно (в случае инициативного предоставления данных наблюдаемым объектом - например, по “подписке”, оформленной драйверами заранее, либо сконфигурированной на уровне элементов объекта).

Драйвер осуществляет предварительную агрегацию и нормализацию полученных данных (в частности, приведение их к общему для систем генерации программ мониторинга формату и семантике). Нормализованные данные подаются на общую шину данных посредством которой они

попадают в подсистему генерации локальных моделей, а также в подсистему базы данных. Данная подсистема использует метаданные и правила, хранящиеся в подсистеме базы данных, для построения иерархической многоуровневой модели на основе одноуровневого потока входных данных. Сформированная локальная многоуровневая модель сохраняется для дальнейшего использования в подсистеме базы данных.

### **3.2 Генерация глобальных моделей наблюдаемого объекта**

Диаграмма последовательности генерации глобальных моделей объекта мониторинга приведена на рисунке 4.

В процессе работы систем генерации программ мониторинга периодически выполняется построение или перестройка глобальных моделей наблюдаемых объектов. При этом выявляются связи между элементами построенных локальных моделей. На основе взаимосвязей определяется набор правил, описывающих возможности и условия перехода элементов наблюдаемого объекта из одного состояния в другое. Построение глобальной модели осуществляется исходя из локальных моделей и правил, их связывающих [15]. Построенная глобальная модель сохраняется в подсистеме базы данных для последующего использования на этапе генерации процессов сбора данных.

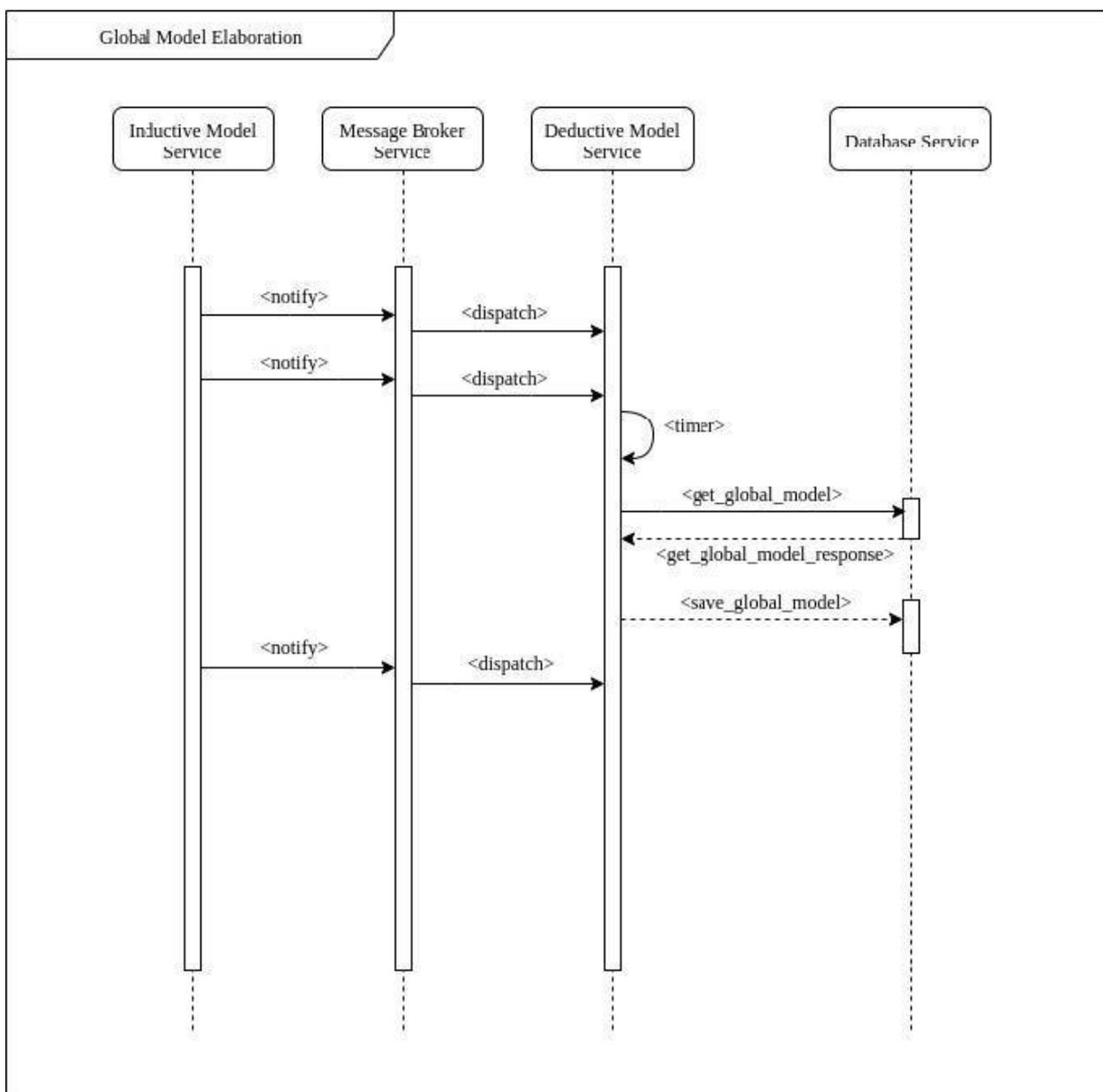


Рис 4. – Сценарий генерации глобальных моделей наблюдаемого объекта

### 3.3 Генерация процессов сбора данных об объекте с целью определения неисправностей, возникающих на объектах

Сценарий генерации процессов сбора данных об объектах приведен на рис. 5. Поток нормализованных данных через шину данных поступает в подсистему генерации процессов сбора данных. Из подсистемы базы данных извлекается текущая актуальная глобальная модель наблюдаемого объекта. Формируется гипотеза о нахождении объекта в исправном состоянии или, наоборот, о наличии неисправности на объекте. Доказательство гипотезы выполняется

исходя из полученных входных данных с использованием глобальной модели объекта. В случае успешного доказательства подсистема генерации процессов завершает свою работу. В противном случае, на основе полученного частичного пути доказательства определяются элементы, не позволившие его выполнить. Формируются уведомления о выявленных элементах и их состояниях. Уведомления вместе с сопутствующей информацией (время возникновения и пр.) посредством шины данных передаются в подсистему взаимодействия с оператором. Для получения дополнительной информации, необходимой для оценки состояния элементов генерируются соответствующие процессы сбора данных.

#### **4 Уровень физического представления**

В общем случае система генерации программ мониторинга является модульной и слабосвязанной, что позволяет реализовывать схемы физического развёртывания подсистем на отдельных вычислительных узлах в рамках некоторой общедоступной сети. Типовым решением является использование клиент-серверной архитектуры на базе стека сетевых протоколов TCP/IP, а также верхнеуровневых прикладных протоколов: HTTP, STOMP, WAMP и пр.

Диаграмма развёртывания системы генерации программ мониторинга представлена на рисунке 6.

Каждая из подсистем предлагаемой модели может быть построена на своём, независимом от остальных, стеке технологий и наборе используемых фреймворков; допускается также использование различных версий фреймворков и сред исполнения в разных подсистемах.

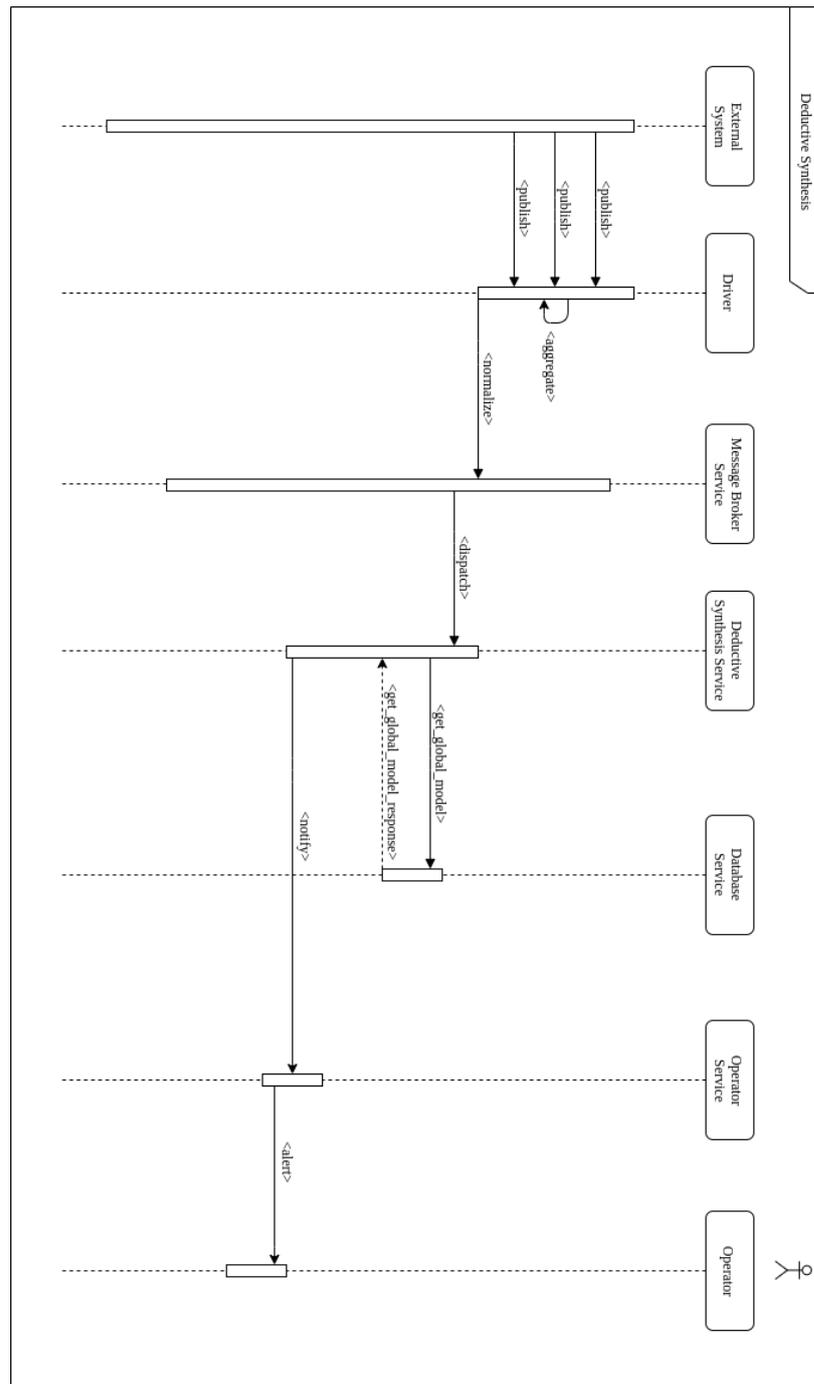


Рис 5. – Сценарий генерации процессов сбора данных о наблюдаемом объекте

Чтобы избежать потенциальных конфликтов при развёртывании в рамках одного сервера, предпочтительным является использование одной из популярных технологий виртуализации или контейнеризации. На текущий момент к таким технологиям относится технология Docker, которая

позволяет создавать изолированные среды выполнения на разделяемых вычислительных ресурсах и облегчает процедуру развёртывания дистрибутивов подсистем на вычислительных узлах. Функционирование системы в виде совокупностей описанных подсистем возможно как в закрытой сети вида Intranet, так и в публичной сети вида Internet.

Предложенные модели систем генерации программ мониторинга являются верхнеуровневыми; конкретная реализация каждой из подсистем может варьироваться в зависимости от используемых протоколов, масштаба систем и пр.

#### **4. Методика адаптации систем генерации программ мониторинга к предметным областям**

Для большинства сложных технических объектов характерен высокий уровень диверсификации и разнообразия конкретных технических решений - используемого стека технологий, набора протоколов, форматов хранения и передачи данных. В таких условиях системы генерации программ мониторинга должны обладать возможностью адаптации под совокупности технических решений, используемых на объектах. Предложенные модели систем генерации программ мониторинга предполагают высокую степень вариативности конкретных программных реализаций. Ниже рассматриваются типовые точки расширения СГПМ, позволяющие адаптировать их к предметным областям.

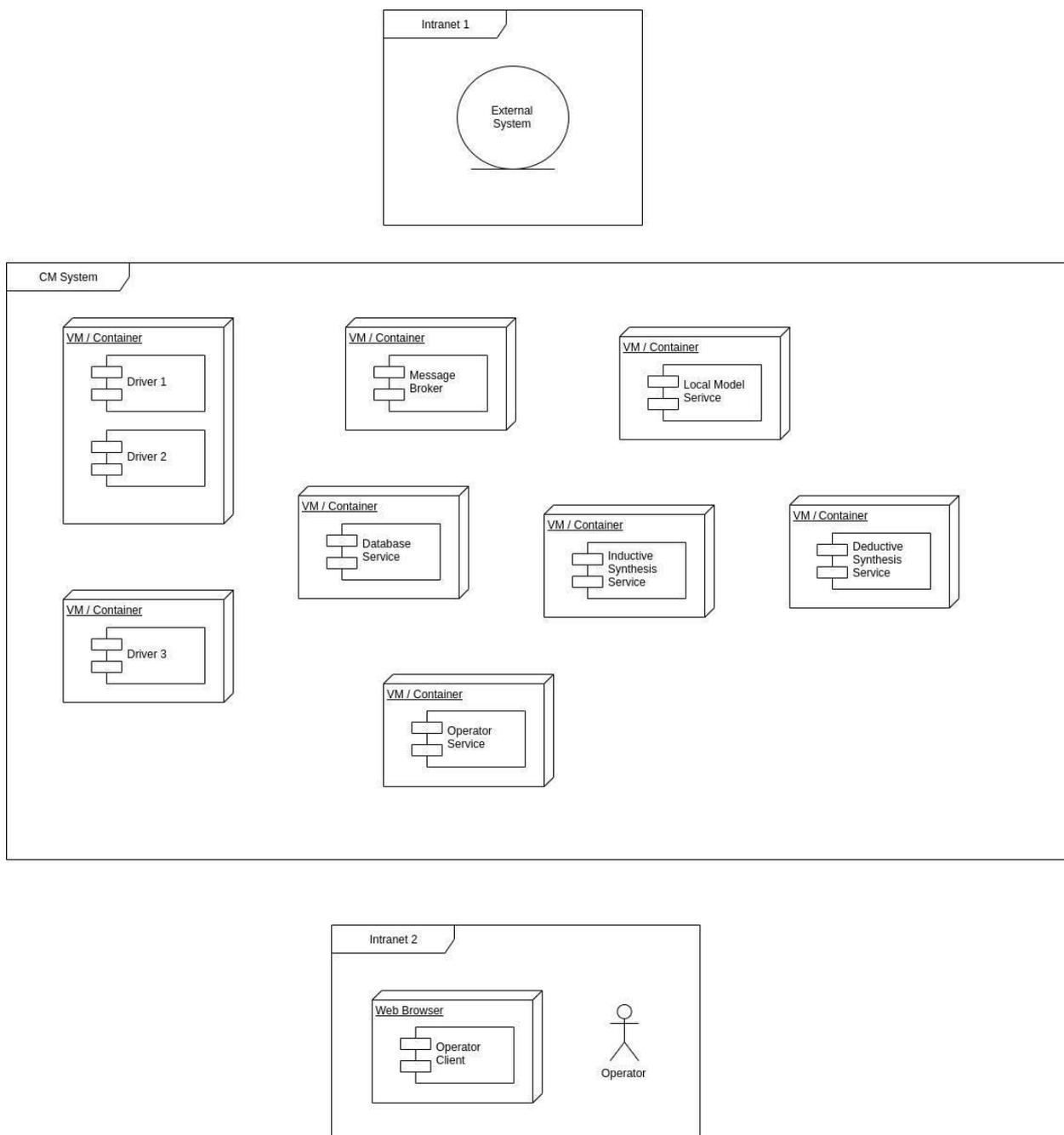


Рис 6. – Среда выполнения системы генерации программ мониторинга

### **Подключение источников информации о техническом объекте посредством драйвера**

Исходными данными, на основе которых формируются локальные и глобальные модели наблюдаемых объектов, являются получаемые от объектов наборы данных, характеризующие состояния их элементов в конкретные моменты времени. Данные наборы, как правило, представляют

собой N-мерный вектор числовых значений. Задача драйверов состоит в предварительной агрегации и нормализации получаемого потока данных и последующей передачи через шину данных в остальные подсистемы генерации программ мониторинга.

В типовых сетевых клиент-серверных приложениях передача данных обычно осуществляется на одном из уровней коммуникационной модели OSI; наибольшее распространение получили протоколы прикладного уровня (HTTP и выше - REST, SOAP, JSON-RPC и пр.), которые адаптированы под большинство современных клиент-серверных фреймворков, реже встречаются более низкоуровневые протоколы на базе TCP или UDP. Во всех случаях предполагается наличие контракта прикладного интерфейса программирования (Application Programming Interface Contract) - некоторой схемы, описывающей передаваемые и получаемые сообщения между участниками информационного взаимодействия.

Драйвера содержит две коммуникационные составляющие. Первая составляющая обеспечивает получение данных от наблюдаемых объектов. Она является уникальной для каждого конкретного драйвера. Другая составляющая позволяет передавать нормализованные данные в другие подсистемы генерации программ мониторинга. При передаче данных осуществляется подключение к шине данных по унифицированному протоколу WS-STOMP. Эта составляющая является единой для всех подключаемых к конкретной системе генерации программ мониторинга, она строго определена на этапе проектирования внутренних протоколов и форматов данных.

К типовым сценариям получения данных от элементов наблюдаемых объектов на уровне сетевых протоколов прикладного уровня относятся:

– инициативный опрос информационных элементов драйвером по протоколу HTTP. В этом случае необходимо предварительное конфигурирование

---

драйвера в результате которого должны быть определены один или несколько сетевых адресов, соответствующих одному или нескольким элементам. С заданной периодичностью  $T$  программный код драйвера выполняет HTTP-запрос на заданные адреса. Рассматриваемый сценарий предполагает наличие потенциальной задержки вплоть до периода опроса  $T$  на получение информации о произошедших изменениях в наблюдаемой системе. Однако, в ряде случаев, такой сценарий является предпочтительным, т. к. не требует адаптировать наблюдаемые объекты, которые, в большинстве случаев, уже имеют в своём составе необходимые для взаимодействия с внешними системами программные компоненты сервера и обработчики запросов;

– пассивное получение потоков данных от элементов наблюдаемых объектов. В этом случае при старте драйвера осуществляется “подписка” на получение данных об элементах объектов, после чего драйвер переходит в режим ожидания и активизируется при получении от объекта очередного фрагмента данных. В такой коммуникационной модели, как правило, используется протокол WebSocket, устанавливающий долговременное подключение к серверам, входящим в состав объектов, а также верхнеуровневые прикладные протоколы. Среди верхнеуровневых протоколов наибольшее распространение при промышленной разработке программного обеспечения на сегодняшний день получили протоколы WAMP и STOMP.

Также, возможна реализация “пассивной” модели опроса с использованием протокола HTTP. При этом используются следующие механизмы:

– Long Polling - механизм открытия долгого HTTP-соединения с периодическим “продлением”. Он позволяет, оставаясь на уровне HTTP, практически исключить задержку на периодический опрос, однако требует затрат дополнительных ресурсов на поддержание связи;

---

– Notification Consumer / Producer – механизм, который при старте работы драйвера оформляет “подписку” на получение данных об объекте на основе поддерживаемого объектом семантического протокола; “подписка” сохраняется объектом и используется им при передаче данных внешним информационным системам. При возникновении изменений в состоянии объекта, им осуществляется публикация нового фрагмента данных в формате HTTP-посылки, которая отправляется на адреса подписавшихся драйверов. Данный способ предполагает поддержку объектом верхнего уровня протокола, что возможно в случаях, если на объекте подобный протокол используется для внутренней передачи информации.

#### **Уточнение правил формирования локальной модели наблюдаемого объекта**

В ряде случаев при генерации программ мониторинга необходимо наличие специализированных для конкретной предметной области знаний или метаинформации о наблюдаемых объектах. В частности, при генерации моделей объектов на основе поступающих данных может потребоваться уточнение правил связывания и группировки отдельных элементов данных из информационных потоков, поступающих от объектов.

Применение специализированных знаний в подсистема формирования локальных моделей обеспечивается за счет ее предварительного конфигурирования и настройки. В процессе конфигурирования подсистемы возможно предварительное задание / уточнение структурных уровней объектов, а описание возможных взаимосвязей между различными состояниями элементов наблюдаемых объектов.

#### **Формулирование понятия целевого и аномального состояний системы**

При генерации процессов сбора данных, предполагающей доказательство гипотезы о состояниях наблюдаемого объекта, необходимо наличия описаний допустимых состояний объектов и их элементов.

---

В простейшем случае рассматриваются две группы состояний:

- исправное состояние, при котором наблюдаемый объект функционирует в штатном режиме, характеризующееся тем, что основные показатели его состояния не выходят за рамки некоторых допустимых значений;
- состояние объекта с неисправностями, при котором основные показатели состояния объекта находятся в опасных (предтревожных) или критических (тревожных) границах.

Как правило, на этапе пусконаладки системы генерации программ мониторинга и её первичной конфигурации осуществляется серия тестовых “прогонов” при различных состояниях наблюдаемых объектов. При таких прогонах, увеличивается нагрузка на объект, временно отключаются некоторые из подсистем, осуществляется воздействие при помощи внешних и внутренних возмущающих факторов и т. д. По результатам каждого из прогонов подсистемой синтеза моделей объектов генерируется несколько локальных моделей объекта - “слепков” состояния объекта. Выполняется классификация сгенерированных моделей, в результате которой каждая из них относится к одной из групп состояний. Возможны два варианта классификации: “ручной” или полуавтоматический и автоматический. При ручной классификации привлекаются эксперты предметной области, которые для моделей определяют состояния объектов. Для автоматической классификации необходимо задание соответствующих правил.

### **Выводы**

Предложенные системы генерации программ мониторинга позволяют проводить наблюдение за сложными техническими объектами за счет сбора данных об объектах и построения на их основе моделей объектов мониторинга. В основу систем положены модели, методы и алгоритмы

---

автоматического построения моделей объектов по поступающим от них данным.

В состав систем входят подсистемы интеграции, получения и обработки данных, генерации моделей наблюдаемых объектов и процессов сбора данных, взаимодействия с оператором. Системы представляются в виде совокупности моделей, которые описывают их на различных уровнях: подсистем и модулей, коммуникационных связей, сценариев использования, а также уровне физического представления.

Предложенная методика адаптации систем генерации программ мониторинга позволяет использовать системы в различных предметных областях за счет использования драйверов для подключения специализированных источников информации о наблюдаемых технических объектах, определения правил формирования моделей объектов и описания их возможных состояний.

### Литература

1. Sohn, H., Farrar, C. R., Hemez, F. M., Shunk, D. D., Stinemates, D. W., Nadler, B. R., & Czarnecki, J. J. (2004). A review of structural health monitoring literature: 1996–2001, Los Alamos National Laboratory (p. 311). LA-13976-MS.
2. Gaikwad, N., Mistry Y. Review on Environment Monitoring System and Energy Efficiency. // International Journal of Engineering Research and Applications. — Т. 5. — pp. 90—92.
3. Chakravarthi, M. K., Tiwari R. K., Handa S. Accelerometer based static gesture recognition and mobile monitoring system using neural networks // Procedia Computer Science. — 2015. — Т. 70. — pp. 683—687.
4. Eissa, M. M., Elmesalawy M. M., Hadhoud M. M. Wide area monitoring system based on the third generation universal mobile telecommunication system (UMTS) for event identification // International Journal of Electrical Power & Energy Systems. — 2015. — Т. 69. — pp. 34—47.

5. Singh, R., Gernaey K. V., Gani R. An ontological knowledge-based system for the selection of process monitoring and analysis tools // Computers & chemical engineering. — 2010. — Т. 34, No 7. — pp. 1137—1154.
  6. Bello, J.P., Silva, C., Nov, O., DuBois, R.L., Arora, A., Salamon, J., Mydlarz, C. and Doraiswamy, H., 2018. SONYC: A system for the monitoring, analysis and mitigation of urban noise pollution. arXiv preprint arXiv:1805.00889.
  7. Suma, N., Samson, S.R., Saranya, S., Shanmugapriya, G. and Subhashri, R., 2017. IOT based smart agriculture monitoring system. International Journal on Recent and Innovation Trends in computing and communication, 5(2), pp.177-181.
  8. Raju, K. R. S. R., Varma G. H. K. Knowledge based real time monitoring system for aquaculture using IoT // IEEE 7th International Advance Computing Conference (IACC). — 2017. — Jan. — pp. 318—321.
  9. Ge, Z., Song Z., Gao F. Review of recent research on data-based process monitoring // Industrial & Engineering Chemistry Research. — 2013. — Т. 52, No 10. — pp. 3543—3562.
  10. Петров К.С., Лебедь К.Г., Тарасенко Д.М., Скориченко В.А. Использование интеллектуальных технологий в современном доме // Инженерный вестник Дона, 2019, №7. URL: [ivdon.ru/ru/magazine/archive/N7y2019/6087](http://ivdon.ru/ru/magazine/archive/N7y2019/6087).
  11. Соколова Э.С., Багиров М.Б, Женарстанов А.М., Бородина Т.Л., Дмитриев Д.В. Система управления сетью “Интернета Вещей” для удалённой идентификации радиометок // Инженерный вестник Дона, 2019, №6. URL: [ivdon.ru/ru/magazine/archive/n6y2019/6014](http://ivdon.ru/ru/magazine/archive/n6y2019/6014).
  12. Osipov, V. Y. Automatic synthesis of action programs for intelligent robots // Programming and Computer Software. — 2016. — Т. 42, № 3. — pp. 155—160.
  13. Osipov, V., Stankova, E., Vodyaho, A., Lushnov, M., Shichkina, Y. and Zhukova, N., 2019, July. Automatic Synthesis of Multilevel Automata Models of
-

Biological Objects. In International Conference on Computational Science and Its Applications. Springer, Cham. pp. 441-456.

14. Zhukova N., Andrianova N., Klimov N. Program System for Object Models Deductive Synthesis // Proceedings of the 24th Conference of Open Innovations Association FRUCT. — FRUCT Oy. 2019. — p. 116.

15. Osipov V., Vodyaho A., Zhukova N. About one approach to multilevel behavioral program synthesis for television devices // International journal of computers and communications. — 2017. — T. 11. — pp. 17—25.

### References

1. Sohn, H., Farrar, C. R., Hemez, F. M., Shunk, D. D., Stinemates, D. W., Nadler, B. R., & Czarnecki, J. J. (2004). A review of structural health monitoring literature: 1996–2001, Los Alamos National Laboratory (p. 311). LA-13976-MS.
2. Gaikwad, N., Mistry Y. International Journal of Engineering Research and Applications. T. 5. pp. 90-92.
3. Chakravarthi, M. K., Tiwari R. K., Handa S. Procedia Computer Science. 2015. T. 70. pp. 683—687.
4. Eissa, M. M., Elmesalawy M. M., Hadhoud M. M. International Journal of Electrical Power & Energy Systems. 2015. T. 69. pp. 34-47.
5. Singh, R., Gernaey K. V., Gani R. Computers & chemical engineering. 2010. T. 34, № 7. pp. 1137-1154.
6. Bello, J.P., Silva, C., Nov, O., DuBois, R.L., Arora, A., Salamon, J., Mydlarz, C. and Doraiswamy, H., 2018. SONYC: A system for the monitoring, analysis and mitigation of urban noise pollution. arXiv preprint arXiv:1805.00889.



7. Suma, N., Samson, S.R., Saranya, S., Shanmugapriya, G. and Subhashri, R., 2017. IOT based smart agriculture monitoring system. International Journal on Recent and Innovation Trends in computing and communication, 5(2), pp.177-181.
8. Raju, K. R. S. R., Varma G. H. K. IEEE 7th International Advance Computing Conference (IACC). 2017. Jan. pp. 318-321.
9. Ge, Z., Song Z., Gao F. Industrial & Engineering Chemistry Research. 2013. T. 52, № 10. pp. 3543—3562.
10. Petrov K.S., Lebed` K.G., Tarasenko D.M., Skorichenko V.A. Inzhenernyj vestnik Dona, 2019, №7. URL: [ivdon.ru/ru/magazine/archive/N7y2019/6087](http://ivdon.ru/ru/magazine/archive/N7y2019/6087).
11. Sokolova E.S., Bagirov M.B, Zhenarstanov A.M., Borodina T.L., Dmitriev D.V. Inzhenernyj vestnik Dona, 2019, №6. URL: [ivdon.ru/ru/magazine/archive/n6y2019/6014](http://ivdon.ru/ru/magazine/archive/n6y2019/6014).
12. Osipov, V. Y. Programming and Computer Software. 2016. T. 42, № 3. pp. 155-160.
13. Osipov, V., Stankova, E., Vodyaho, A., Lushnov, M., Shichkina, Y. and Zhukova, N., 2019, July. In International Conference on Computational Science and Its Applications. Springer, Cham. pp. 441-456.
14. Zhukova N., Andrianova N., Klimov N. Proceedings of the 24th Conference of Open Innovations Association FRUCT. FRUCT Oy. 2019. p. 116.
15. Osipov V., Vodyaho A., Zhukova N. International journal of computers and communications. 2017. T. 11. pp. 17-25.