

УДК 004.067

Н. А. Жукова, А. И. Водяхо, М. А. Червонцев  
Санкт-Петербургский государственный электротехнический  
университет «ЛЭТИ» им. В. И. Ульянова (Ленина)

## Архитектурное проектирование систем мониторинга состояния сложных технических и природных систем

*Современный этап развития информационных технологий характеризуется многократным увеличением объемов обрабатываемой информации и, как следствие, увеличением сложности создаваемых информационных систем, в частности систем мониторинга и управления. В данной статье представлено описание архитектурного подхода к проектированию систем мониторинга и управления состоянием сложных технических и природных систем. Рассматривается обобщенная структура системы мониторинга, определяются основные альтернативные подходы к построению систем мониторинга и управления и типовые решаемые задачи. Предлагается архитектурный подход к построению систем мониторинга, в рамках которого определяется девять архитектурных точек зрения: обобщенная, системная, функциональная, модельная, информационная, объектно-ориентированная, сервисно-процессная, структурная, проектная и точка зрения, учитывающая стандарты. Рассматриваются стеки архитектурных моделей. Предлагаемый подход был использован при построении ряда систем мониторинга и управления, относящихся к различным предметным доменам, таким как космическая отрасль, телекоммуникации, океанография и медицина.*

### Архитектура программных систем, архитектурные модели, системы мониторинга

Постоянное расширение сферы применения информационных технологий (ИТ) приводит к проблеме больших данных [1], т. е. к многократному увеличению объемов обрабатываемой информации и, как следствие, увеличению сложности создаваемых информационных систем (ИС). Расширение сферы применения ИТ происходит как за счет систем низкой ценовой категории, например Интернета вещей, где требуется получить предельно низкую совокупную стоимость владения, так и за счет систем, отвечающих за функционирование жизненно важных систем, где требуется обеспечить предельные показатели по надежности функционирования, достоверности формируемых результатов. Большинство систем являются распределенными, имеют большой срок службы. В ходе эксплуатации системы многократно модернизируются. Вносимые изменения могут относиться к их структуре, логике работы, составу используемого оборудования.

Объекты предметных областей и используемые в этих областях ИС образуют связанные структуры, являющиеся распределенными гетерогенными структурами с множественными связями. При этом появляется задача отслеживания и управле-

ния их состоянием. Задача управления сводится к решению двух основных задач: определению текущего состояния структур посредством мониторинга и формированию управляющих воздействий с целью перехода в требуемое состояние [2].

**Обобщенная структура системы мониторинга (СМ).** Подсистему мониторинга и целевой объект можно рассматривать либо как единую систему, либо целевой объект можно рассматривать как контекст, в котором происходит функционирование СМ. Выбор варианта зависит от специфики решаемой задачи. Если подсистема мониторинга не может выдавать управляющие воздействия, т. е. оказывать влияние на целевой объект, то последний можно рассматривать как контекст. СМ и целевой объект образуют единую систему в случае, когда СМ может оказывать управляющие воздействия на целевой объект, т. е. является системой мониторинга и управления (СМУ).

Целевые объекты, как правило, имеют связи с другими объектами. Совокупность целевого объекта и связанных с ним объектов образует целевую систему (ЦС). Как ЦС, так и СМ включают программный (бизнес-логику) и инфраструктурный (ИК) компоненты. ИК – это датчики, исполнитель-

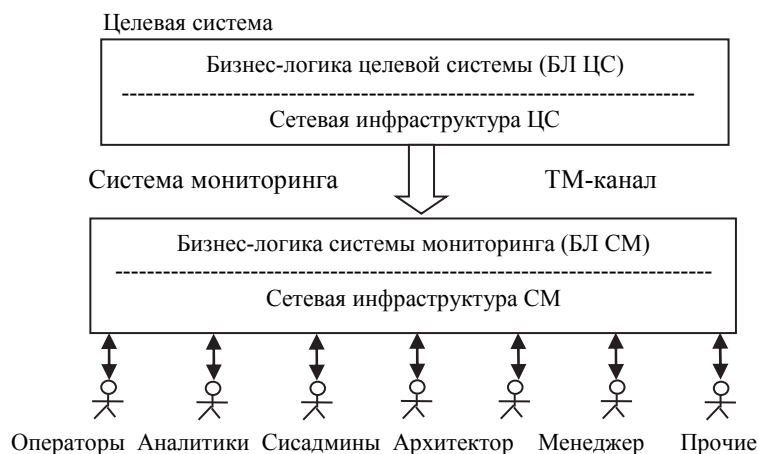


Рис. 1

ные механизмы, серверы и сетевое оборудование. Обобщенная структура СМ показана на рис. 1.

ЦС представляется в виде  $\{Д, ИМ, ЭО, БП, С, Л\}$ , где Д – множество датчиков; ИМ – множество исполнительных механизмов, ЭО – множество элементов оборудования; БП – множество реализуемых бизнес-процессов, С – множество сигналов (сообщений), которыми обмениваются перечисленные ранее элементы, Л – записи в логфайлы о событиях типа начало и конец бизнес-процесса, Д – источники сигналов, которые генерируются в контексте конкретного БП. Множество С имеет формат  $\{БП, ВМ\}$ , где БП – идентификатор бизнес-процесса, ВМ – временные метки, каждая из которых имеет формат  $\{И, З, T_{\text{выд}}\}$ , где И – имя параметра; З – значение параметра;  $T_{\text{выд}}$  – временная метка выдачи сигнала.

ИМ являются приемниками сигналов. Принимаемые сигналы имеют формат  $\{И, З, T_{\text{пр}}\}$ , где И – имя параметра; З – значение параметра;  $T_{\text{пр}}$  – временная метка приема сигнала. ЭО – элемент, которому можно поставить в соответствие состояние (исправен, неисправен, частично исправен). В качестве ЭО могут выступать Д, ИМ, серверы, элементы сетевой инфраструктуры. БП – процесс обработки информации, который имеет множество входов и выходов, по которому поступают и выдаются С – в общем случае это сообщения, которыми обмениваются БП. Л – записи в логфайлы о системных событиях обычно с привязкой ко времени. Информация из логфайлов также доступна для анализа.

СМ представляется в виде  $\{МЦС, ЭМЦС, ПР\}$ , где МЦС – модель целевой системы; ЭМЦС – эталонная модель целевой системы; ПР – процессор.

МЦС – совокупность данных, пересылаемых от ЦС в СМ одной или несколькими порциями. ЭМЦС – эталонная модель ЦС, с которой сравнивается МЦС, определяет функционирует ЦС в нормальном режиме или наблюдаются отклонения. ПР – процессор, который реализует процесс сравнения МЦС и ЭМЦС.

Каждый из параметров ЭМЦС описывается в терминах 4 цветов: зеленый – нормальное функционирование; желтый – значения параметра показывают, что при определенных условиях могут возникнуть ошибки в функционировании; красный – параметры имеют значения, однозначно свидетельствующие об ошибочном функционировании; черный – значения параметра не получены или устройство неисправно.

Для описания ЭМЦС в общем случае может использоваться темпоральная логика. Математический аппарат, применяемый для работы с временными последовательностями, является достаточно сложным. Задача упрощается тем, что все процессы детерминированы, т. е. известны моменты начала и окончания БП.

**Основные типы систем мониторинга.** СМУ на верхнем уровне можно классифицировать по трем параметрам: структуре ЦС (централизованная, распределенная), структуре СМУ (централизованная, распределенная) и способу обработки (отложенная, online). Таким образом, имеем 8 базовых вариантов (табл. 1).

В общем виде СМУ представлена на рис. 2, что соответствует варианту 7. Данный вариант соответствует случаю, когда ЦС является системой распределенных целевых объектов (РЦО). Это могут быть, в частности, подвижные объек-

Таблица 1

№	Структура ЦС	Структура СМУ	Способ обработки
0	Централизованная	Централизованная	Отложенная
1			Online
2		Распределенная	Отложенная
3	Online		
4	Распределенная	Централизованная	Отложенная
5			Online
6		Распределенная	Отложенная
7			Online

ты. СМУ также представляет собой распределенную систему (РСМУ). На рис. 2 – это дерево, но могут использоваться и другие топологии. РЦО и РСМУ связаны между собой несколькими каналами телеметрии.

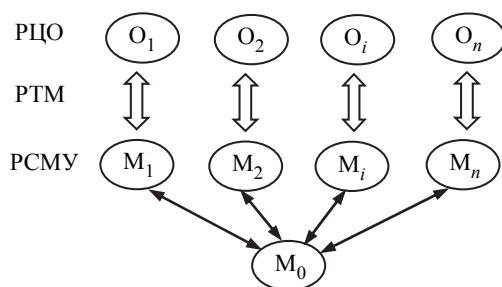


Рис. 2

**Типовые задачи, решаемые СМУ.** Можно выделить следующие основные задачи, решаемые СМУ: анализ телеметрии в отложенном режиме, обработка online телеметрии и управление целевой системой, управление сетевой инфраструктурой СМУ, отладка процесса работы с телеметрией, моделирование работы СМУ на этапе проектирования.

*Анализ телеметрии в отложенном режиме.* На вход СМ поступает поток событий в формате {время–имя–значение}. Требуется найти момент времени или событие, начиная с которого наблюдаются отклонения, отследить историю развития ситуации.

*Обработка online телеметрии и управление целевой системой.* На вход СМУ поступает поток событий в формате {время–имя–значение}. Требуется при поступлении очередной тройки проверить, что события развиваются ожидаемо, если события развиваются с отклонением от нормы, то выполнить процедуру коррекции, которая может заключаться либо в изменении настроек, либо в реконфигурации, для чего в ЦС выдается одно или несколько пар {имя–значение}.

*Управление сетевой инфраструктурой СМУ.* В случае если СМУ реализуется как распределенная система, то появляется задача мониторинга и управления собственной сетевой инфраструктурой. Задача похожа на предыдущую зада-

чу – обработка online телеметрии и управление целевой системой.

*Отладка процесса работы с телеметрией.* Проведение ряда различных экспериментов на этапе отладки процедур мониторинга конкретной СМУ.

*Моделирование работы СМУ на этапе проектирования.* Проведение ряда различных экспериментов на этапе первичной отладки СМУ.

**Архитектурный подход к построению систем мониторинга.** СМУ являются очень большим и разнообразным, с точки зрения назначения и используемых подходов к технической реализации, подклассом информационно-управляющих систем.

Архитектурный подход (стиль) проектирования наряду со стилями, основанными на календарном планировании (Calendar Driven), стилем, основанным на управлении требованиями (Requirements Driven), стилем, в основу которого положена разработка документации (Docunent Driven), и стилем, в основу которого положено управление качеством (Quality Driven), является одним из возможных подходов к проектированию ИС [3].

Архитектурный подход предполагает разработку метамодели в форме архитектурного описания [4], которое предназначено для следующих целей:

1. Архитектурное описание может использоваться в качестве языка общения разработчиков и заказчиков системы.
2. На основе архитектурного описания может быть сгенерирован исполняемый код.
3. Повторное использование архитектурного знания [5].

Архитектурное описание включает в себя заинтересованные стороны, их интересы, архитектурные виды и точки зрения, модели и фреймворки [4].

Архитектурный фреймворк обычно определяют как эталонное решение, на базе которого могут быть созданы требуемые системы. Архитектурный фреймворк может быть поддержан инструментальными средствами разработки.

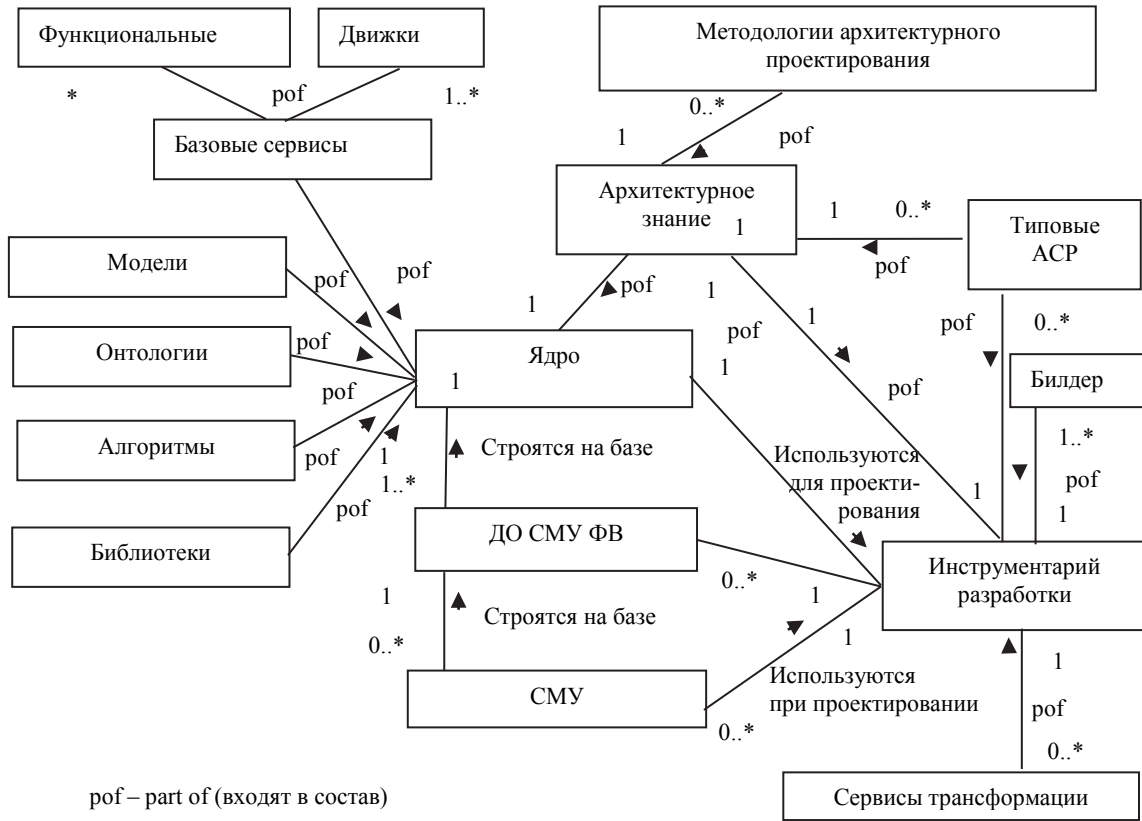


Рис. 3

Сущность предлагаемого подхода иллюстрирует семантическая диаграмма (рис. 3). Основными сущностями, в терминах которых описывается предлагаемый подход, являются: СМУ Ядро, доменно-ориентированные (ДО) СМУ фреймворки (ФВ) и инструментарий разработки целевых СМУ, типовые архитектурно-структурные решения (АСР). Ядро включает набор базовых сервисов, в число которых входят функциональные сервисы разного уровня и сервисы-движки, модели, библиотеки, алгоритмы и онтологии. Частью ядра также являются архитектурные знания. К основным элементам архитектурного знания относятся типовые архитектурно-структурные решения и процессы проектирования, в частности архитектурного проектирования.

На базе ядра с использованием инструментария могут быть построены ДО СМУ ФВ, которые, в свою очередь, могут использоваться для построения целевых СМУ. Целевые СМУ могут строиться непосредственно на основе ядра.

Инструментальная среда разработки основывается на использовании архитектурного знания. Основу среды разработки составляет среда разработки (Билдер), а также сервисы трансформации и библиотека типовых АСР, которая является составной частью архитектурного знания. В целом

интерпретация термина «архитектурное знание» совпадает с предложенным в [5], но в рамках рассматриваемого подхода используется в смысле «знание, представленное в виде онтологии».

**Архитектурные модели СМУ.** Архитектурная модель СМУ представляет собой архитектурное описание в рамках стандарта [4] и включает в себя 10 точек зрения (Viewpoint, ТчЗ) (рис. 4): обобщенную; системную; функциональную; модельную; информационную; объектно-ориентированную; сервисно-процессную; структурную, проектную; ТчЗ, учитывающую стандарты. Совокупность этих точек зрения позволяет реализовать архитектурную модель СМУ. Основное назначение модели – обеспечение перехода от программно-независимых моделей к исполняемым моделям.

ТчЗ, учитывающая стандарты	Обобщенная ТчЗ	Системная ТчЗ	Проектная ТчЗ
		Функциональная ТчЗ	
		Модельная ТчЗ	
		Информационная ТчЗ	
		Объектно-ориентированная ТчЗ	
		Сервисно-процессная ТчЗ	
		Структурная ТчЗ	

Рис. 4

1. Обобщенная ТчЗ интегрирует все точки зрения и образует архитектурный контекст для других точек зрения.

2. Системная ТчЗ рассматривает систему как совокупность взаимодействующих подсистем, определяет способы взаимодействия подсистем. Системная ТчЗ используется в качестве структурного описания системы. Эта точка зрения соответствует структуре, представленной на рис. 2.

3. Функциональная ТчЗ определяет реализуемый функционал. Он соответствует выделенным ранее типовым задачам.

4. Модельная ТчЗ описывает систему в терминах используемых моделей и алгоритмов. В качестве базового может использоваться подход, известный как проектирование на моделях (Model Driving Engineering – MDE). Идея этого подхода состоит в построении стека моделей [6].

5. Информационная ТчЗ описывает информационные структуры для представления исходных данных и знаний и результатов их многоуровневых трансформаций. Информационная ТчЗ является ключевой, поскольку СМУ ориентированы на обработку данных и знаний.

6. Объектная ТчЗ рассматривает систему в терминах объектно-ориентированного (ОО) проектирования. На сегодняшний день основное проектирование ИС ведется средствами ОО-проектирования, которые хорошо поддержаны инструментальными средствами. Следует отметить, что в настоящее время достаточно хорошо проработаны вопросы прямого и обратного перехода между ОО и онтологическими представлениями [7].

7. Сервисно-процессная ТчЗ описывает динамику функционирования СМУ, при этом бизнес-процесс определяется как последовательность обращений к сервисам. Каждый бизнес-процесс при обращении к нему рассматривается как сервис. Поэтому разделять бизнес-процессы и сервисы представляется нецелесообразным.

8. Структурная ТчЗ описывает СМУ в терминах программных модулей, из которых она состоит.

9. Проектная ТчЗ рассматривает СМУ как совокупность требуемых характеристик и возможностей, а также с точки зрения процесса проектирования. В основу точки зрения представляется целесообразным положить модельно-ориентированный подход (Model Driven Architecture – MDA). В качестве архитектурного фреймворка целесообразно использовать фреймворк Захмана [8], настроенный на данную предметную область. СМУ могут быть реализованы с использованием

различных технологий и парадигм построения программных систем.

10. ТчЗ, учитывающая стандарты, рассматривает, в частности, действующие технические стандарты, методики, руководства, ограничения и т. п. Сюда входят стандарты, определяющие процесс проектирования, аппаратные и программные платформы, стандарты безопасности, разного рода отраслевые стандарты и т. п.

Далее более детально раскрываются модели систем СМУ и рассматриваются технологические вопросы их построения.

**Стек моделей СМУ** определяется в рамках концепции MDE. Ключевыми для СМУ MDE являются следующие понятия: модель (model – M), метамодель (metamodel – MM), связывание моделей (model weaving – MW), преобразование моделей (model transformation – MT). Рассмотрим модели, образующие стек моделей СМУ. Стек моделей содержит пять уровней M0–M4 (табл. 2). Уровень M0 представляет собой описание конечной СМУ, уровень M1 соответствует модели таковой системы. На уровне M2 размещаются метамодели, которые определяются для предметных доменов. Эти модели содержат описания особенностей предметных областей, для которых строятся СМУ. Доменные модели можно считать ключевыми моделями для рассматриваемого метода проектирования и гибкого подхода в целом. Уровень M3 – это уровень метамodelей, определяющих весь класс СМУ. Эти модели наследуются от метамodelей информационно-управляющих систем (ИУС). Каждая модель конечной системы должна соответствовать некоторой метамodelи, которая определяет ее структуру и семантику в более общем виде. Можно говорить, что метамodelь определяет ключевые сущности, отношения и ограничения языка моделирования. Уровень M4 – это уровень метамodelей для метамodelей, которые называют метаметамodelями. Поскольку уровень метаметамodelей является самым высоким уровнем, то метаметамodelи соответствуют только сами себе.

Таблица 2

Уровень	Описание
Уровень M4 (верхний уровень)	Уровень метамodelей для метамodelей
Уровень M3	Метамodelи класса систем СМУ
Уровню M2	Метамodelи доменных систем СМУ
Уровню M1	Модель конечной системы СМУ
Уровень M0 (нижний уровень)	Моделируемая система

Ограничений на вложенность моделей в рамках каждого из уровней не налагается. Как правило, модели, относящиеся к одному уровню, отличаются степенью детализации.

Рассмотрев основные модели систем СМУ, рассмотрим основные операции их преобразования. К ним относятся операции трансформации и связывания. Поясним эти понятия.

Представление набора моделей в виде стека требует определения процедуры перехода от модели верхнего уровня к моделям нижнего уровня. Процесс преобразования моделей – это процесс, который определяет, каким образом можно получить целевую модель из одной или нескольких исходных моделей. Процесс трансформации моделей можно определить следующим образом (рис. 5). Пусть модель  $M_a$ , соответствующая метамодели  $MM_a$ , трансформируется в модель  $M_b$ , которая соответствует метамодели  $MM_b$ . Само преобразование можно определить как  $MT$ .  $MT$  также представляет собой модель, соответствующую метамодели  $MMT$ , которая, в свою очередь, является моделью и должна соответствовать метаметамодели, например MOF-модели.

Модели могут трансформироваться либо «по горизонтали», либо «по вертикали». В первом случае трансформируемая модель преобразовы-

вается в модель, которая принадлежит одному с ней уровню. Во втором случае трансформируемая модель является моделью более высокого уровня.

С точки зрения типов трансформируемых моделей можно выделить следующие основные варианты: трансформация типа «формальная модель – формальная модель», описание, имеющее формализованную синтаксическую структуру (например, XML-описание) – формальная модель, текстовое описание – описание, имеющее формализованную синтаксическую структуру, текстовое описание – формальная модель, формальная модель – программный код.

Можно определить следующие базовые механизмы трансформации моделей: прямое преобразование (Direct manipulation), структурное преобразование (Structure Driven), операционное (Operational), преобразование с использованием шаблонов (Template-based), реляционные (Relational), преобразование графа (Graph Transformation), XSLT-преобразования (XSLT-based), гибридное (Hybrid).

Для связывания моделей разных типов предназначен специальный тип моделей – модели связывания, основное назначение которых состоит в установлении соответствия между отдельными элементами моделей (низкоуровневое связывание).

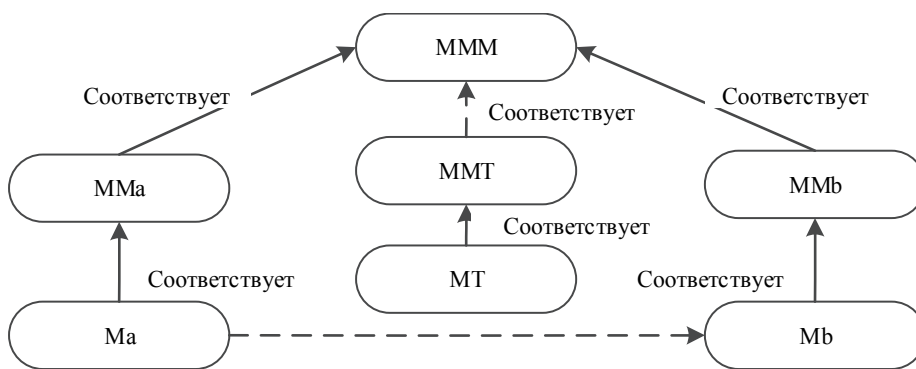


Рис. 5

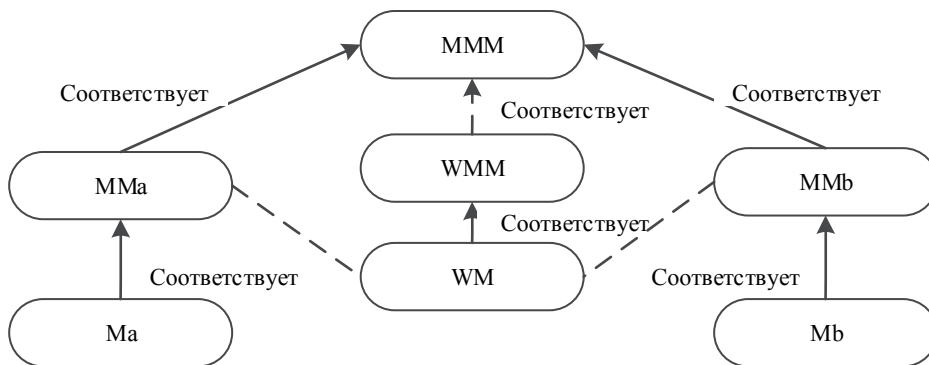


Рис. 6

Модель связывания может быть использована для связывания любых элементов разных метамоделей, таких как метаклассы, атрибуты, ссылки, типы данных. На рис. 6 модель связывания WM используется для установления соответствия между элементами метамоделей ММа и ММб. При этом сама WM должна соответствовать метамоделю связывания WMM.

При необходимости использования для описания архитектуры системы множества мета-метамоделей, их свойств и отношений между их элементами вводится понятие более высокого уровня, в качестве которого выступает понятие мегамодели [9]. Мегамодель можно рассматривать как метамоделю MDE, которая позволяет определять отношения между моделями и мета-моделями, а также их свойства. С использованием мегамоделей определяются отношения между отдельными элементами, в частности моделями, которые фактически являются мета-моделями, видами и точками зрения. Следует заметить, что такой подход в полной мере соответствует стандарту [4], который предлагает считать, что каждый вид должен соответствовать точке зрения, а каждая модель должна соответствовать типу модели.

Мегамоделирование предусматривает построение фреймворка. Модели, определяемые таким фреймворком, имеют некоторые отличия от [8]. В частности, фреймворк определяет модели следующих типов: терминальные модели (Terminal Models), которым соответствуют реальные системы и некоторые мета-модели; мета-модели (MetaModels), которые определяют доменно-ориентированные концепции; метамета-модели (MetaMetamodels), которые описывают концепции верхнего уровня и соответствуют сами себе.

Мета-модели и метамета-модели выступают в качестве эталонных моделей (Reference Models), поскольку они представляют своего рода эталон для моделей, которые должны им соответствовать. Терминальные модели подразделяются на модели трансформации (Transformation Models), связывания (Weaving Models) и мегамодели (MegaModels). Модели трансформации определяют процессы преобразования моделей, модели связывания определяют отношения между моделями, а мегамодели предназначены для поддержки процесса мегамоделирования. Следует отметить, что терминальная модель не является абстрактным метаклассом и может использоваться для порожде-

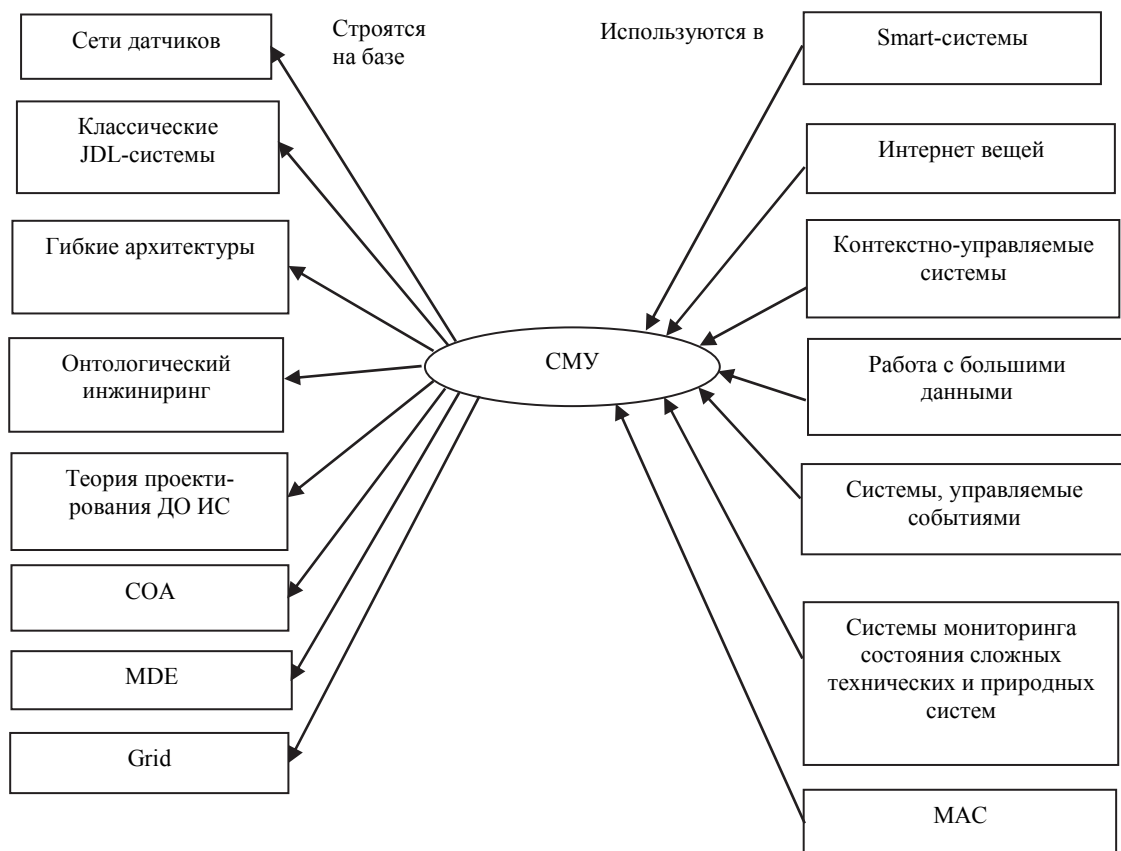


Рис. 7

ния конкретных экземпляров и поэтому должна соответствовать одной метамодели.

**Технологический стек систем мониторинга и управления.** На рис. 7 приведен расширенный список технологий, на базе которых строятся СМУ, и список технологий, которые могут использовать СМУ.

В качестве основных технологий, которые могут быть использованы при построении СМУ, можно отметить следующие: сети датчиков [10], классические JDL-системы [10], agile-архитектуры [11], онтологический инжиниринг [7], теория проектирования доменно-ориентированных ИС [12], сервисно-ориентированные архитектуры (COA) [13], [14], проектирование на моделях (Model Driven Engineering – MDE) [6], Grid-технологии [15]. В свою очередь, СМУ может входить в качестве составной части в такие технологии, как разного рода Smart-системы; Интернет вещей; контекстно-управляемые системы (Context Aware Systems); системы, ориентированные на работу с большими данными; системы, управляемые событиями; системы мониторинга состояния сложных технических и природных систем; мультиагентные системы (MAC) [16].

**Результаты использования предлагаемого подхода для решения задач практики.** Предлагаемый подход был использован при построении ряда СМУ, относящихся к различным предметным доменам, таким как космическая отрасль, океанография, медицина, телекоммуникации.

В целом полученные результаты проектирования СМУ показали следующее:

1. В области объектов космического назначения обеспечено повышение точности и достоверности результатов обработки данных, получаемых от наблюдаемых объектов, достигнуто существенное сокращение времени их обработки при решении задач мониторинга.

2. В области океанографии усовершенствованы существующие модели, методы и системы, обеспечивающие процессы мониторинга и управления за счет включения в их состав адаптивных процедур, настройки параметров, внесения изменений в конфигурацию.

3. В области практической и исследовательской медицины значительно увеличено число решаемых прикладных задач в результате модификации постановок существующих задач, а также рассмотрения принципиально новых, ранее не решавшихся задач, требующих использования новых прикладных теорий.

4. В области телекоммуникаций существенно расширена область применения систем мониторинга и управления за счет низкой стоимости их проектирования и сопровождения, а также удобства работы с ними конечных пользователей.

Предлагаемый архитектурный подход может быть использован для решения отдельных задач в других областях, включая области образования и городского хозяйства. Архитектурный подход может найти применение при построении систем мониторинга и управления самого разного назначения, включая системы, построенные по принципу Интернета вещей и Интернета событий.

## СПИСОК ЛИТЕРАТУРЫ

1. Van der Aalst W. Process Mining. Data Science in Action. 2nd ed. Berlin, Heidelberg: Springer-Verlag, 2016. 467 p.

2. Охтилев М. Ю., Соколов Б. В., Юсупов Р. М. Интеллектуальные технологии мониторинга состояния и управления структурной динамикой сложных технических объектов. М.: Наука, 2005. 291 с.

3. Bass L., Clements P., Kazman R. Software Architecture in Practice. 3rd ed. Upper Saddle River, NJ: Addison-Wesley, 2013. 661 p.

4. International Standard ISO/IEC/IEEE 42010 Systems and software engineering. Architecture description. URL: <http://www.iso-architecture.org/42010/> (дата обращения: 31.05.2018).

5. Software Architecture Knowledge Management / M. A. Babar, T. Dingsøyr, P. Lago, H. van Vliet. Dordrecht, Heidelberg, London, New York: Springer, 2009. 279 p.

6. Frankel D. S. Model Driven Architecture. Indianapolis, Indiana: Wiley Publishing, Inc., 2003. 355 p.

7. Gasevic D., Djuric D., Devedzic V. Model Driven Architecture and Ontology Development. Springer-Verlag, 2006. 315p.

8. The Zachman International e-Commerce Site: <http://www.zachmaninternational.com> (дата обращения: 31.05.2018).

9. Zivin B. E., Jouault J., Valduriez P. On the Need for Megamodels. URL: <https://scinapse.io/papers/195085068> (дата обращения: 31.05.2018).

10. Blasch E., Bosse E., Lambert D. High-Level Information Fusion Management and System Design. Norwood, MA: Artech House Publishers, 2012. 376 p.

11. Babar M. A., Brown A. W., Mistrik I. Agile Software Architecture. Waltham, MA: Elsevier Inc., 2014. 392 p.

12. Эванс Э. Предметно-ориентированное проектирование – структуризация сложных программных систем: пер. с англ. М.: Издательский Дом «Вильямс», 2010. 448 с.

13. Erl T. SOA: principles of service design. NY: Prentice Hall, 2008. 545 p.



14. Ньюмен С. Создание микросервисов. СПб.: Питер, 2016. 304 с.

15. Hwang K., Fox G. C., Dongarra J. J. Distributed and Cloud Computing Waltham. MA: Elsevier Inc., 2012. 631 p.

16. Walton C. Agency and the Semantic Web. New York: Oxford University Press Inc., 2007. 249 p.

---

N. A. Zhukova, A. I. Vodyaho, M. A. Chervontsev  
*Saint Petersburg Electrotechnical University «LETI»*

## ARCHITECTURAL APPROACH TO DEVELOPMENT OF INFORMATION SYSTEMS FOR MONITORING THE STATUS OF COMPLEX ARTIFICIAL AND NATURAL SYSTEMS

*Modern step of information technologies development can be characterized by tremendous increasing of information volumes to be processed. As a result, the complexity of new information systems is permanently increasing. Monitoring systems are one of the important subclasses of information systems, which are widely used both for monitoring and control the status of complex artificial and natural systems. In the article the description of an architectural approach to the development of the information systems for the monitoring the status of complex systems of different nature is discussed. The generalized structure of the monitoring system is suggested. Main alternative approaches to monitoring systems organization and typical functionality of monitoring systems are analyzed. An architectural approach to monitoring system development is suggested. This approach assumes usage of 9 viewpoint: All Viewpoint, Systems Viewpoint, Capability Viewpoint, Model Viewpoint, Object Viewpoint, Information Viewpoint, Service-Process Viewpoint, Project Viewpoint and Standards Viewpoint. Suggested approach was successfully used for building a number of monitoring systems for different subject domains such as telecommunications, space, oceanography and medicine.*

**Software architecture, architectural models, monitoring systems**

---

УДК 004.9

В. Н. Кафтасьев, В. А. Гладцын  
АО «НИЦ ЭТУ»

И. О. Бакаев  
Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина)

## Разработка программного обеспечения систем представления и отображения информации коллективного пользования

*Описывается разработанное программное обеспечение, предназначенное для использования на аппаратных средствах систем представления и отображения информации коллективного пользования. Данное программное обеспечение было использовано при создании центров мониторинга, контроля и управления Глобальной навигационной спутниковой системы ГЛОНАСС. Приводится состав разработанного программного обеспечения, его основные функциональные характеристики. Одной из особенностей созданного программного обеспечения является функционирование в среде отечественной операционной системы в защищенном исполнении Астра Линукс Special Edition. Также приводится состав библиотек, разработанных в процессе создания описываемого программного обеспечения, предназначенных для поддержки разработки программ сторонними пользователями, использующими в своих программах отображение в формах различных графиков, гистограмм, круговых диаграмм, часов с различными циферблатами и временными поясами. Разработка программного обеспечения проводилась на языке C++ с использованием среды разработки Qt.*

**Программное обеспечение, видеостена, операционная система Астра Линукс**

В рамках работ по созданию центров контроля управления и подтверждения характеристик Глобальной навигационной спутниковой

системы ГЛОНАСС предприятие АО «НИЦ ЭТУ» разработало программное обеспечение (ПО) систем представления и отображения информации,

---