

УДК 004.067

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

Мультимодельный подход к построению систем мониторинга

Рассматривается проблема управления процессом мониторинга состояния сложных систем различной природы. Отмечается, что управление системами мониторинга сложных систем различной природы, состоящими из большого числа элементов, – сложная задача, особенно при динамической структуре системы. Для решения указанной проблемы предлагается использовать модельный подход. В качестве модели структуры целевой системы предлагается использовать относительно конечный автомат, а в качестве модели поведения – граф потока работ. Актуальность моделей обеспечивается за счет снятия и анализа лог-файлов. Авторами ранее был предложен алгоритм синтеза многоуровневых относительно конечных автоматов. Для поддержания в актуальном состоянии моделей поведения предлагается использовать алгоритмы Process Mining. Приводится классификация задач мониторинга и рассматриваются возможные подходы к их решению.

Мультимодельный подход, системы мониторинга, программные архитектуры

Современный этап развития человеческого общества связан с все более активным использованием информационных технологий (ИТ) и постепенным переходом к цифровому обществу, что ставит перед ИТ-специалистами новые и непростые задачи. При этом бурное развитие микроэлектроники и телекоммуникаций дает им в руки принципиально новый инструментарий. Переход к цифровому обществу связан, в первую очередь, с расширением сферы применения цифровых технологий, а также с необходимостью обрабатывать очень большие объемы данных – это хорошо известная проблема больших данных. Практическая реализация концепции цифрового общества требует создания очень сложных информационных систем (ИС). В этом контексте возникает задача управления гетерогенной распределенной динамической сетевой инфраструктурой, состоящей из очень большого числа элементов. Одной из ключевых подзадач управления распределенной инфраструктурой является задача мониторинга ее состояния. В самом общем виде мониторинг можно определить как процесс получения знаний о целевой системе (ЦС) в целях решения

конкретных задач заинтересованными сторонами. В качестве исходных данных для функционирования систем мониторинга (СМ) выступают сообщения и сигналы, поступающие от ЦС, а также контекстное знание о ЦС.

В условиях высокой сложности структуры и ее динамичности задача мониторинга становится нетривиальной и не может быть эффективно решена с использованием существующих подходов. Предлагаемый авторами подход к построению систем мониторинга можно определить как мультимодельный.

Существующие подходы к построению систем мониторинга и используемые технологии. Наиболее подробно и системно вопросы, связанные с построением систем мониторинга, рассмотрены в [1]. Однако в этой монографии изложены вопросы, преимущественно связанные со структурной динамикой ИС. Классические подходы к управлению сетями, в частности корпоративными сетями в рамках ITIL/ITSM [2], начинают давать сбои при попытке применить их, например, для интернета вещей. Можно также указать ряд статей авторов, посвященных вопро-

сам мониторинга [3], [4]. Кроме того, имеется достаточно много работ по тематике, связанной с отдельными технологиями, используемыми в рамках предлагаемого подхода. Это обработка данных в сетях датчиков [5], работы, посвященные синтезу автоматов из данных [6], [7], вопросам восстановления структур бизнес-процессов по лог-файлам (Process Mining) [8], [9].

Модельно-ориентированный подход к построению СМ. В основу развиваемого подхода к построению систем мониторинга положен модельный, а точнее мультимодельный подход [1]. Идея состоит в том, что ключевой элемент СМ – система моделей. Основными моделями являются структурные модели, модели поведения и контекстные модели. Все основные операции, используемые в процессе реализации процедур мониторинга, – это операции над моделями.

Обобщенная структура СМ. В самом общем виде структура СМ может быть представлена в виде: $СМ = \{ЦС, С, ПМ, ЗС\}$, где ЦС – целевая система; ПМ – собственно подсистема мониторинга; С – множество сообщений о событиях, множество заинтересованных сторон (ЗС) (рис. 1).

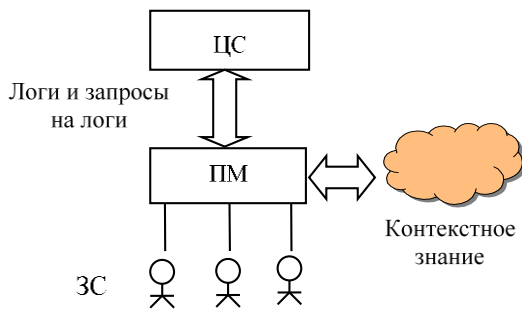


Рис. 1

Целевая система. В качестве ЦС могут выступать системы различной природы. ЦС описывается посредством моделей, которые характеризуют ее структуру и реализуемое поведение (функциональность).

Типы ЦС. С точки зрения принадлежности к доменам задач можно выделить по крайней мере следующие основные классы ЦС: социальные системы, биологические системы, природные системы, антропогенные системы, гибридные системы, относящиеся к нескольким предметным доменам.

Ограничения, накладываемые на ЦС. В рамках развиваемого подхода на структуру и поведение ЦС накладываются следующие основные ограничения:

1. С точки зрения ПМ, ЦС является пассивной, т. е. она никак не реагирует на попытки снятия логов. Она не может препятствовать снятию логов, не изменяет своего состояния и поведение, не может преднамеренно исказить их содержание.

2. Источники логов привязаны к подсистемам и никак не связаны между собой.

3. Все логи содержат структурированные данные, которые представлены в XES (eXtensible Event Stream)-формате или могут быть приведены к XES-формату [10].

4. ЦС не является системой жесткого реального времени, и все управляющие воздействия формируются на основе модели ЦС.

5. Все неструктурированные данные должны быть приведены к XES-формату в рамках предварительной обработки.

6. Все шумы должны быть вычищены на этапе предварительной обработки.

Таким образом, в рамках развиваемого подхода не решаются следующие типы задач:

1. Если на вход СМ поступают логи, представляющие собой неструктурированные данные (тексты на естественных языках, голосовые сообщения). К таким задачам можно отнести разного рода поисковые системы, получающие данные из Интернета, социальных сетей.

2. Если ЦС является активным компонентом, т. е. может препятствовать снятию логов посредством их подавления или преднамеренного искажения, или, обнаружив попытку снятия логов, ЦС изменяет свою структуру и (или) поведение.

3. Если семантика сообщений (логов) неизвестна. Например, исследования языков, но которых общаются подсистемы ЦС (люди, животные, роботы etc).

Структура (статика) ЦС определяется в терминах подсистем, связей между ними, связей с внешним миром и связей с подсистемой мониторинга. ЦС может иметь структуру произвольной сложности.

В этом случае можно говорить о структурной динамике.

Динамика (поведение) ЦС в зависимости от ее типа может описываться разными способами. Чаще всего встречаются следующие варианты:

1. Динамика ЦС описывается в терминах изменения состояний ее отдельных подсистем, а также ее входов и выходов. Структура ЦС может быть как статической, так и динамической.

2. Динамика ЦС описывается в терминах реализуемой функциональности. В свою очередь, термин «функциональность» может интерпретироваться по-разному применительно к конкретным классам ЦС. При этом под функциональностью часто понимается возможность реализации некоторого процесса (применительно к антропогенным системам используется термин «бизнес-процесс» (БП) [11]).

3. Для описания динамики ЦС используется некоторая комбинация перечисленных ранее подходов.

Первый вариант можно позиционировать как классический. Второй вариант связан с получающимися все большее распространение процессно-ориентированными ИС (Process Aware Systems) [9], в основу описания функционирования которых положено понятие БП.

Третий вариант – это комбинация первого и второго вариантов. Примером такого варианта может служить современная сложная сетевая инфраструктура, работающая по принципу интернета вещей. В этом смысле данный вариант может служить основой для построения обобщенной модели ЦС, которую можно определить как $ЦС = \{СТ, БП\}$, где СТ – многоуровневая динамическая структура ЦС, БП – множество динамических БП, протекающих в ЦС. Ограничений на структуру БП не накладывается. Лог-файлы, или просто логи (Л) – это основной, а как правило, и единственный источник информации о событиях, происходящих в ЦС.

Типовые задачи, решаемые СМ. Общая постановка задачи мониторинга в рамках мульти-модельного подхода. В общем случае требуется построить искомую модель по известным моделям и контекстному знанию: $M_t \leftarrow \{M_s, Л, K_c\}$, где M_t – целевая модель; M_s – известные модели; Л – снимаемые с ЦС логи; K_c – контекстное знание. В качестве моделей выступают структурная модель ЦС и модель поведения ЦС. В рассмотрении участвуют эталонная и реальная модели.

Частные постановки задачи мониторинга. В зависимости от того, что известно и что требуется узнать о ЦС, можно выделить 4 базовых варианта постановки задачи мониторинга (табл. 1).

Таблица 1

Задача	Статическая модель	Динамическая модель
А	Известна	Известна
В	Известна	Неизвестна
С	Неизвестна	Известна
Д	Неизвестна	Неизвестна

Задача А. Проверка актуальности модели. Соответствует случаю, когда известна как статическая модель (структура ЦС), так и динамическая, т. е. известны модели, относящиеся ко всем уровням, которые представляют интерес. В общем виде Задача А может быть сформулирована следующим образом. Используя известные статическую и динамическую модели ЦС и логи, при минимальных затратах проверить актуальность моделей и, если модели не актуальны, построить актуальные статическую и динамическую модели с заданным уровнем качества.

На практике могут появляться несколько отличные варианты постановки Задачи А, которые можно рассматривать как частные случаи. Можно выделить по крайней мере 6 таких частных задач:

А1) требуется подтвердить сам факт актуальности моделей;

А2) требуется откорректировать модель;

А3) требуется определить точку, время и причину расхождения эталонной и реальной моделей;

А4) требуется построить картину развития аварийной ситуации;

А5) требуется определить угрозы появления аварийных ситуаций и представить варианты развития;

А6) требуется обнаружить недокументированные активности с ЦС.

Задача В. Построение модели поведения по структурной модели и логам. В данном случае структура ЦС известна, но отсутствуют знания о БП, протекающих в ЦС. К этому классу, в частности, относится классическая задача Process Mining [8], когда требуется восстановить структуру БП либо с нуля, либо по известной динамической модели (БП) более высокого уровня. В качестве частных случаев можно выделить следующие задачи:

В1) известны статические модели, относящиеся ко всем уровням, и динамические модели, относящиеся к верхним уровням, требуется построить недостающие модели БП;

В2) требуется по логам отследить БП, относящиеся к определенным группам, например БП отдельных пользователей или БП определенного типа;

В3) требуется быстро определить подозрительную активность БП.

Задача С. Построение структурной модели по модели поведения и логам. БП, протекающие в системе, известны или могут быть восстановлены по логам. Требуется определить реальную структуру ЦС. Здесь можно выделить следующие частные задачи:

С1) требуется получать актуальную информацию о постоянно изменяющейся структуре системы;

С2) по известному поведению требуется определить структуру «черного ящика» за минимальное число шагов;

С3) требуется определить неисправный элемент системы за минимальное время;

С4) требуется выяснить, какой элемент системы является «узким местом».

Например, применительно к биологическим системам по внешним проявлениям требуется определить больной орган, в технических системах это может быть вариант, когда необходимо по аномалии в поведении определить причину неисправности.

Задача D. Построение структурной и поведенческой моделей методом спуска. Отсутствует полная информация как о структуре системы, так и о БП. Имеется только набор логов. Предполагается, что структура и семантика логов известны.

Обобщенная структура ПМ. ПМ можно определить как систему обработки логов. На основе логов и контекстной информации строятся модели ЦС, позволяющие выдавать информацию ЗС в соответствии с их ролью и их запросами. Обобщенная структура ПМ показана на рис. 2.

ПМ включает в себя 4 основных элемента: $ПМ = \{P, АУСД, АУД БП, АУМ\}$, где P – репозиторий; АУСД – автомат управления структурной динамикой; АУД БП – автомат управления динамикой БП; АУМ – автомат управления мониторингом. Репозиторий предназначен для хранения моделей, логов, скриптов, контекстной и другой информации.

Используемые модели. *Требования, предъявляемые к моделям.* Основные требования, предъявляемые к моделям, используемым в рамках развиваемого подхода, определяются, в первую очередь, спецификой решаемых задач и могут быть сформулированы следующим образом: возможность описывать постоянно меняющуюся многоуровневую структуру произвольной сложности, возможность автоматически строить структурную модель на основе ограниченной информации, поступающей в форме логов.

Основные требования к поведенческим моделям – возможность отслеживать ход и корректность выполнения БП по поступающим логам; возможность корректировать структуру БП при изменении структуры БП.

Модели, используемые в рамках модельного подхода к построению СМ, – это, по сути, архитектурные модели [12].

Классификация моделей – это, по существу, ответ на следующие вопросы: моделируется ли структура или поведение ЦС; чьими глазами исследователь смотрит на ЦС; на каком уровне исследуется ЦС; моделируется ли некоторый класс ЦС или конкретная ЦС; какие формализмы используются для построения моделей ЦС.

Для построения СМ могут использоваться разные формализмы, например: UML-диаграммы, сети Петри, графы, в частности графы потока работ, онтологии, конечные автоматы.

С учетом требований, предъявляемых к статическим моделям, наиболее перспективным представляется использование относительно ко-

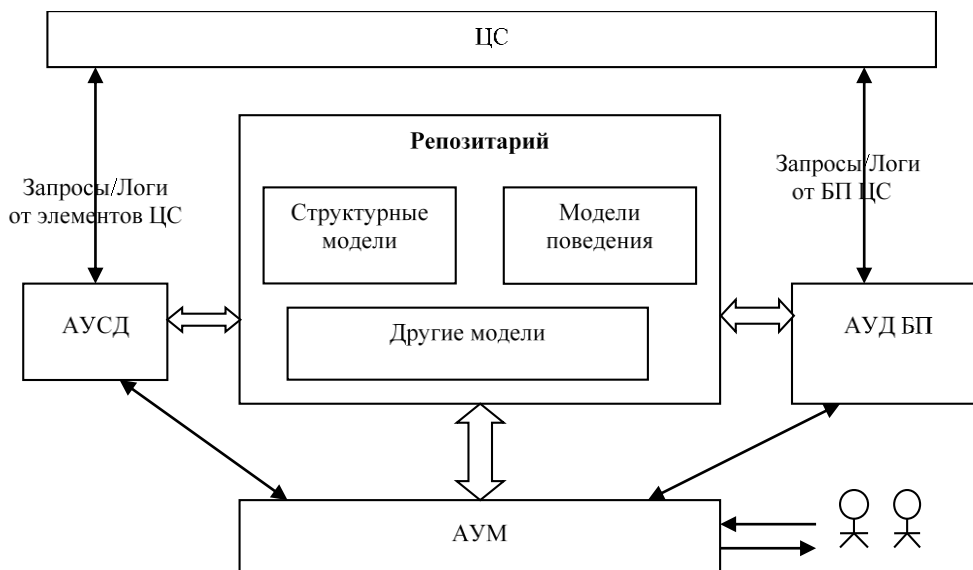


Рис. 2

нечных многоуровневых автоматов. Основным доводом в пользу этого подхода является возможность автоматического синтеза подобных автоматов по логам [6], [7]. Статические модели ЦС описывают ЦС в терминах подсистем, связей между ними, их входов и выходов и свойств (атрибутов).

Каждый элемент статической модели может иметь произвольное число атрибутов. Значения атрибутов могут быть считаны в произвольный момент времени в формате лога либо определены косвенно или посредством логического вывода. В атрибутах присутствуют параметры, характеризующие техническое состояние в терминах «зеленый-желтый-красный» (ЗЖК) [1], где «зеленый» означает, что элемент полностью исправен, «желтый» – элемент частично исправен, «красный» – элемент неисправен. Кроме того, параметр может находиться в состоянии «недоступен». В этом случае его значение может быть определено через логический вывод. С каждым элементом может быть связано произвольное число точек снятия логов. Различают эталонную и рабочую модели. Эти модели могут иметь разные атрибуты. В частности, рабочая модель привязана к конкретному моменту времени. Эталонная модель может быть привязана к интервалу, а параметры могут быть окрашены в ЗЖК-цвета. Элементы (подсистемы) могут появляться и исчезать в произвольный момент времени (включение-выключение, поломка-исправление, вход-выход пользователей в систему). Модели новых элементов могут быть известны и могут храниться в репозитории, но могут быть и неизвестны. Связи между элементами прописываются в атрибутах. Связи могут меняться, т. е. исчезать и появляться.

Более формально модель, описывающая структурную динамику, представленную в форме многоуровневого относительно конечного автомата, может быть определена следующим образом.

Автомат = {Множество входных сообщений, множество выходных сообщений и логов, Состояния автомата, Множество таблиц переходов}.

Состояние автомата = {Общее состояние, Текущая структура},

Общее состояние = {Исправен, Неисправен, Частично исправен},

Текущая структура = {Элементы, Связи},

Элемент структуры = {Атрибуты, Связи},

Атрибут = {Множество пар Имя-Значение},

Связи = {Сиблинги, Родители, Дети}.

На вход автомата могут приходиться:

1) запросы о состоянии (запросы о текущем состоянии, запросы о прошлых состояниях, запросы о будущих состояниях);

2) запросы на реконфигурацию (запросы на реконфигурацию с целью повышения производительности, запросы на реконфигурацию при появлении (исчезновении) элемента, логи от ЦС).

На вход автомата могут поступать логи (логи, формируемые встроенными системами контроля и диагностики, входящими в состав подсистем ЦС; логи, формируемые по запросу от АУСД; логи, формируемые диагностическими скриптами, запускаемыми АУСД).

Автомат управления структурной динамикой (АУСД). Это процессор (движок), обеспечивающий доступ к модели посредством выполнения некоторого набора операций. АУСД предназначен для отслеживания состояния ЦС и управления структурной динамикой ЦС.

На вход АУСД поступают логи, которые могут быть сообщениями (ответами на запросы) либо сигналами о событиях. АУСД может выдавать запросы о состоянии элементов ЦС и сигналы управления о реконфигурации или настройке параметров. Все запросы конечных пользователей о состоянии элементов ЦС проходят через АУМ).

Функционирование автомата можно описать следующим образом. Автомат переходит в очередное состояние под действием сигнала или лога, который содержит информацию об изменении состояния элемента ЦС. Автомат имеет память трасс переходов. При появлении запроса о состоянии ЦС или элементов ЦС от пользователя автомат формирует требуемое представление на основе вектора текущего состояния и (или) исторических векторов. Автомат может запрашивать состояние элементов ЦС. При этом состояние автомата не изменяется.

Обобщенный алгоритм функционирования АУСД выглядит следующим образом:

1. Запуск процесса поллинга.
2. Ожидание информации с входов.
3. При появления информации об изменении состояния элемента выполняются действия:
 - 3.1. Скорректировать модель ЦС.
 - 3.2. Если необходимо, реконфигурировать ЦС.
 - 3.3. Отправить информацию о реструктуризации АУД БП и заинтересованным пользователям.
 - 3.4. Скорректировать и перезапустить процесс поллинга.

4. При появлении запроса от пользователя о состоянии ЦС:

4.1. Если возможно сформировать ответ на основе вектора состояния, то сформировать и отправить.

4.2. Если невозможно, то сгенерировать и запустить скрипт. По результатам работы скрипта сформировать и отправить ответ.

5. Если на вход поступает запрос о причинах неправильного хода бизнес-процесса, то запускается процедура диагностики, по результатам которой формируется ответ.

Поведенческие (динамические) модели ЦС. Динамическая модель – это совокупность бизнес-процессов, реализуемых в ЦС.

Для описания БП может быть использован широкий спектр моделей и языков, таких, как сети Петри, автоматы, Work Flow-графы, BPEL, YAWL и др. [8]. Информацию о ходе реализации БП можно получить посредством логов в XES-формате.

Исходя из перечисленных ранее требований, предъявляемых к модели поведения, в качестве основной модели поведения была выбрана модель, основанная на графе потока работ [8]. При этом интерес представляют все 3 компонента графа потока работ: граф потока управления, граф потока данных (объектов) и ресурсный граф [8]. Основным доводом в пользу такого выбора послужило наличие в свободном доступе инструментария для синтеза БП по логам.

Более формально модель поведения может быть описана следующим образом:

$\langle \text{Модель поведения} \rangle ::= \langle \text{Эталонная модель} \rangle | \langle \text{Текущая модель} \rangle,$

$\langle \text{Текущая модель} \rangle ::= \langle \text{Множество логов} \rangle,$

$\langle \text{Лог} \rangle ::= \langle \text{Идентификаторы} \rangle \langle \text{Временная метка} \rangle \langle \text{Информация об артефактах} \rangle \langle \text{Информация о ресурсах} \rangle,$

$\langle \text{Идентификаторы} \rangle ::= \langle \text{Идентификатор события} \rangle \langle \text{Идентификатор приложения} \rangle \langle \text{Идентификатор экземпляра приложения} \rangle.$

Эталонная модель представляет собой граф потока работ, вершины которого нагружены информацией, идентичной информации, присутствующей в логах в форме Имя – Значения. При этом, если возможно, могут быть заданы интервалы значений по принципу ЗЖК. Более формально модель поведения может быть описана следующим образом:

Эталонная модель = {Множество вершин, Множество дуг, Множество разметок}.

$\langle \text{Вершина} \rangle ::= \langle \text{Имя вершины} \rangle \langle \text{Параметр-Значения} \rangle.$

Содержимое поля Параметр-Значения может задаваться двумя способами: один параметр – одно значение или один параметр – интервал значений, один параметр 3 интервала (зеленый, желтый, красный).

Автомат управления БП (АУБП). На вход АУБП поступают логи. На выходе имеем сигналы управления БП, которые осуществляют настройки БП. Предполагается, что сама система реконфигурации находится внутри ЦС, а сигналы управления выбирают один из вариантов реконфигурации из списка возможных. АУБП может запрашивать информацию о текущем состоянии у АУСД и получать такую информацию. АУБП информирует АУСД о нарушениях в процессе реализации БП и получает информацию об изменении структуры ЦС. Кроме того, на вход АУБП поступают запросы о текущем состоянии БП, на которые АУБП выдает ответы.

Обобщенный алгоритм функционирования АУБП:

1. На вход АУБП поступают логи и сигналы, формируемые БП.

2. Логи сравниваются с эталонными логами:

2.1. Если лог попадает в «зеленую зону», то продолжение.

2.2. Если лог попадает в красную или желтую зону, то посылается сообщение в АУСД.

2.3. Если поступает сигнал о реструктуризации ЦС, то рассматривается возможность реструктуризации БП. Если это возможно, то выполняется реструктуризация и выдается сообщение в АУМ.

Разметки реализуются стандартными для графов потока работ методами [8], [9].

Следует заметить, что, к сожалению, представлять эталонную модель в форме следов нельзя, поскольку для корректно работающего процесса иногда невозможно точно определить порядок поступления логов.

Возможные подходы к решению задач мониторинга. Задачи типа А. Проверка актуальности моделей. Сам факт неактуальности модели может обнаруживаться либо через появление сообщений (логов), либо при срабатывании внутренних схем контроля. Предполагается, что процедуры поллинга запущены.

Задача А1. Подтвердить корректность структурной модели и модели поведения. Признаком некорректности моделей могут служить:

сообщение от АУСД о срабатывании схем внутреннего контроля ЦС или реакция на запрос о перестройке БП, сообщение от АУБП о появлении желто-красного лога, реакция – запрос к АУСД о мониторинге структуры ЦС с целью выявления причины.

Задача А2. Коррекция моделей. Корректировка структурной модели производится после установления факта изменения состояния элемента. При этом возможны варианты: корректировка непосредственно по информации из лога, запуск некоторой процедуры реструктуризации модели по результатам дополнительного тестирования, запуск процесса построения новой структурной модели.

Коррекция модели поведения выполняется в следующих случаях: поступление сигнала о реструктуризации ЦС, появление желтого или красного лога, появление запроса от пользователя о реструктуризации, например в целях повышения эффективности функционирования ЦС.

Задача А3. Определить точки расхождения эталонной и реальной моделей. Эта задача обычно возникает в системах, работающих в *post mortem*-режиме (режиме отложенной обработки). Типовой подход состоит в просмотре логов, поступающих от БП, т. е. от АУБП. Просмотр можно начинать как с начала, так и с конца. Можно отслеживать граф потока управления, ресурсный граф и граф потока объектов на предмет появления желто-красных логов. Можно отслеживать логи, поступающие от АУСД, на предмет появления желто-красных логов. Обычно такой подход позволяет обнаружить время и причину появления ошибки.

Задача А4. Построить картину развития аварийной ситуации. Эта задача обычно возникает в системах, работающих в *post mortem*-режиме. Модель развития ситуации можно представить как пару упорядоченных и выровненных по времени цепочек желто-красных логов от АУСД и АУБП, которые требуется связать причинно-следственными связями.

Задача А5. Определить угрозы появления аварийных ситуаций и представить варианты развития. Угрозы возникновения аварийных ситуаций определяются в терминах появления желтых логов – как от АУСД, так и от АУБП. Прогноз строится на основе предположения, что желтые логи превращаются в красные, и определения структурного элемента, ответственного за появление желтого лога, если аварийная ситуация выявлена через лог от АУБП.

Задачи типа В. Построение модели поведения по структурной модели и логам. Это достаточно хорошо исследованная классическая задача Process Mining [8], [9].

Задача В1. Построить модель поведения по логам. Восстановление графа потока управления – это задача Process Mining в классической постановке. Восстановление ресурсного графа и графа потока объектов – это отдельные задачи.

Задача В2. Восстановить по логам структуру отдельных БП. Это частный случай задачи В1. Требуется использовать логи, ассоциированные с конкретным ресурсом.

Задача В3. Найти определенные паттерны нежелательного поведения. Решение требует наличия библиотеки паттернов.

Задачи типа С. Построение структурной модели по модели поведения и логам. Поскольку в качестве структурной модели выступает многоуровневый относительно конечный автомат, то эта задача – суть задача синтеза автомата по логам.

Задача С1. Получить актуальную информацию о постоянно изменяющейся структуре ЦС. Основной подход к решению этой задачи – постоянное отслеживание логов и сообщений от систем встроенного контроля ЦС, запросов на реконфигурацию от пользователей. В определенных ситуациях может возникнуть необходимость перестроить структурную модель.

Задача С2. Построить структурную модель по модели поведения. Это задача синтеза относительно конечного автомата по логам. Ее можно также определить как задачу определения структуры «черного ящика» в терминах относительно конечного автомата по известному поведению за минимальное число шагов. Алгоритмы синтеза таких автоматов достаточно подробно исследованы в [6], [7], [13].

Задача С3. Определить неисправный элемент ЦС за минимальное время. В данной задаче известно, что некоторая подсистема ЦС работает некорректно, но не известен конкретный элемент, являющийся причиной неисправности. Требуется выбрать тест, обеспечивающий нахождение неисправного элемента.

Задача С4. Определить элемент системы, являющийся «узким местом». Понятие «узкое место» можно определить по-разному. Если речь идет о ненадежном элементе, то это задача А5. Если речь идет об «узком месте» с точки зрения производительности или пропускной способности

сти, то эта задача решается посредством снятия соответствующих логов. Это могут быть как логи, снимаемые напрямую со структурных элементов, так и логи, получаемые от БП.

Задачи типа D. Построение структурной и поведенческой моделей методом спуска. Эту достаточно часто встречающуюся задачу можно решать разными способами в зависимости от того, какие логи доступны. Если доступны логи, относящиеся ко всем уровням, то на их основе может быть синтезирован относительно конечный автомат (задача С2), а затем можно построить БП (задача В1). Если доступны логи, относящиеся только к определенным уровням, то можно использовать метод «последовательного спуска», т. е. построены модели, которые ограничиваются уровнем детализации доступных логов.

Обобщенный алгоритм выглядит следующим образом:

1. Определить модель поведения на верхнем уровне i_0 .

2. Применив процедуру определения структуры «черного ящика» определить структуру верхнего уровня i_0 .

3. Для каждого элемента модели уровня i_0 с помощью тестовых последовательностей построить тем же способом модели более низкого уровня. Таким образом, алгоритм сводится к последовательному применению процедуры определения структуры «черного ящика» и построению БП по логам.

Случай, когда сами логи доступны, но их структура неизвестна, является отдельной задачей, которая в данной статье не рассматривается, хотя описание решения этой задачи для частных случаев можно найти, например, в [14], где решается задача определения структуры логов. Этот тип задач может быть определен как отдельный тип (тип Е). Задача типа Е может быть сформулирована как задача определения структуры лог по структурной модели и модели поведения.

Сервисная модель управления процессом мониторинга. При наличии большого числа структурных решений СМ на первый план выступает задача использования инвариантных к реализации архитектурных решений. Достаточно очевидным в этой ситуации является использование сервисно-ориентированных архитектур. Обобщенное представление СМ в рамках предлагаемого подхода представлено на рис. 3.

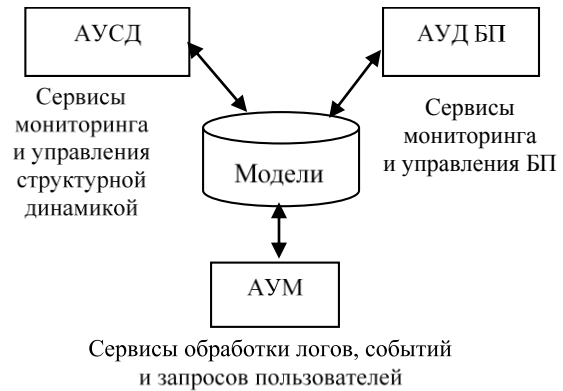


Рис. 3

В соответствии с предлагаемым подходом сервисы разделяются на 3 группы и 2 уровня. Выделяются 3 группы сервисов: сервисы мониторинга и управления структурной динамикой, сервисы мониторинга и управления БП и сервисы обработки логов, событий и запросов пользователей. Сервисы мониторинга и управления структурной динамикой реализуются АУСД, а сервисы мониторинга и управления БП – АУД БП. Это внутренние сервисы, с которыми работает АУМ. Вся входная информация (логи, сигналы и запросы пользователей), которая может инициировать транзакции, связанные с мониторингом, поступает на АУМ. При этом АУСД и АУД БП могут взаимодействовать с ЦС на предмет запроса и получения логов и т. п.

1. Внешние сервисы (обработка запросов пользователей).

2. Сервисы управления структурной динамикой:

2.1. Подтвердить актуальность статической модели ЦС.

2.2. Восстановить структуру ЦС по логам.

2.3. Отработать сигнал о неисправности элемента.

2.4. Отработать сообщение о реконфигурации ЦС.

2.5. Определить время и элемент, который сообщил о неисправности.

2.6. Определить причины плохого лог.

2.7. Построить процесс развития аварийной ситуации.

3. Сервисы мониторинга и управления БП:

3.1. Построить БП по логам.

3.2. Реконфигурировать БП при изменении структуры.

Обобщенный алгоритм управления процессом мониторинга в терминах сервисной модели. В самом общем виде алгоритм функционирования АУПМ выглядит следующим образом:

1. АУПМ запускает фоновые процессы мониторинга, которые выполняются автоматами АУСД и АУБП.

2. АУПМ ожидает появления событий.

2.1. При поступлении от АУСД сигнала о реструктуризации ЦС АУПМ выдает сигнал АУБП о необходимости реконфигурирования БП.

2.2. При поступлении сигнала от АУБП о том, что БП выполняется некорректно, АУПМ посылает в АУСД запрос об обнаружении причины отклонения.

Развиваемый мультимодельный подход к построению систем мониторинга, в основу которого положена работа с динамическими структурными

моделями, представленными в виде относительно конечных автоматов и динамических моделей бизнес-процессов, можно рассматривать как архитектурный подход. Ключевыми элементами развиваемого подхода являются разработанные авторами механизмы синтеза относительно конечных автоматов и их совместное использование с механизмами Process Mining, что позволило создать ряд достаточно эффективных систем мониторинга для разных предметных доменов, таких, как системы мониторинга распределенных объектов, медицинские системы, системы мониторинга состояния мирового океана и др. [3], [4], [12], [15], [16].

СПИСОК ЛИТЕРАТУРЫ

1. Охтилев М. Ю., Соколов Б. В., Юсупов Р. М. Интеллектуальные технологии мониторинга и управления структурной динамикой сложных технических объектов. М.: Наука, 2006. 410 с.

2. ITIL – IT Service Management. URL: <https://www.axelos.com/best-practice-solutions/itil>. (дата обращения 23.03.19).

3. Водяхо А. И., Жукова Н. А., Мустафин Н. Г. Онтологический подход к построению систем мониторинга ресурсов в сетях кабельного телевидения // Изв. СПбГЭТУ «ЛЭТИ». 2017. № 6. С. 16–21.

4. Жукова Н. А., Водяхо А. И., Червонцев М. А. Архитектурное проектирование систем мониторинга состояния сложных технических и природных систем // Изв. СПбГЭТУ «ЛЭТИ». 2018. № 7. С. 38–46.

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

6. Осипов В. Ю. Синтез результативных программ управления информационно-вычислительными ресурсами // Приборы и системы управления. 1998. № 12. С. 24–27.

7. Osipov V., Vodyaho A., Zhukova N. About one approach to multilevel behavioral program synthesis for television devices // Intern. j. of computers and communications. 2017. Vol. 11. P. 17–25.

8. Van der Aalst W. Process Mining Data Science in Action. Second Edition. Heidelberg: Springer, 2016. 468 p.

9. Process-aware information systems: bridging people and software through process technology / M. Dumas, W.

van der Aalst, A. ter Hofstede (editors). Hoboken, New Jersey: John Wiley & Sons, Inc. 2005. 409 p.

10. Günther C. W. XES Standard Definition. 2009. URL: www.xes-standard.org (дата обращения 24.03.19).

11. Fundamentals of Business Process Management / M. Dumas, M. La Rosa, J. Mendling, H. A. Reijers // Second Edition. Heidelberg: Springer, 2018. 527 p.

12. Rozanski N., Woods E. Software Systems Architecture. Working with Stakeholders Using Viewpoints and Perspectives. Upper Saddle River, NJ: Addison-Wesley, 2012. 715 p.

13. Inductive Synthesis of the Models of Biological Systems According to Clinical Trials / V. Osipov, M. Lushnov, E. Stankova, A. Vodyaho // Intern. Conf. on Computational Science and Its Appl. (ICCSA 2017). Lecture Notes in Computer Science. Vol. 10404. Cham: Springer, 2017. P. 103–115.

14. Балтрашевич В. Э., Жукова Н. А., Соколов И. С. Графовая модель группового телеметрического сигнала со сменой кадра // Изв. СПбГЭТУ «ЛЭТИ». 2010. № 10. С. 12–17.

15. Беляев С. А., Васильев А. В., Кудряков С. А. Архитектура системы мониторинга информационных трендов на основе свободного программного обеспечения // Программные продукты и системы. 2016. Т. 29, № 4. С. 85–88.

16. Беляев С. А., Черепкова Ю. С. Архитектура среды моделирования для проведения экспериментов с интеллектуальными агентами // Программные продукты, системы и алгоритмы. 2017. № 3. С. 4.

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

MULTIMODEL APPROACH TO BUILDING OF MONITORING SYSTEMS

In the paper the problem of management of process of monitoring of a condition of complex systems of various nature is considered. It is noted that management of the systems of monitoring of complex systems of various nature consisting of a large number of elements is a difficult task, especially in case the structure of a system is dynamic. For solving this problem it is suggested to use model driven approach. As model of structure of a target system it is offered to use multilevel relatively finite operational automata and as behavior model it is offered to use the work flow graph of works. The actuality of models is provided by means of analysis of log files. Authors have suggested an algorithm of multilevel relatively finite operational automata synthesis earlier. For maintenance in actual state of behavior models one can use Process Mining algorithms. Classification of problems of monitoring is given in article and possible approaches to their decision are considered.

Multimodal approach, monitoring systems, software architecture