

УДК 004.067

С. А. Аббас, А. И. Водяхо

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

Н. А. Жукова

Санкт-Петербургский институт информатики и автоматизации Российской академии наук

М. А. Червонцев

АО «НИЦ СПб ЭТУ»

Аунг Мьо То

Университет ИТМО

Об одном подходе к построению систем сбора данных в киберфизических системах, построенных на платформах туманных вычислений

Современный этап развития техники и технологий характеризуется повышением сложности создаваемых антропогенных систем, постоянным расширением сферы применения информационных технологий и повышением интеллектуального уровня создаваемых антропогенных систем и появлением новых парадигм построения информационно-ориентированных систем – таких, как киберфизические системы, интернет вещей, облачные и туманные системы. Современные информационно-ориентированные системы часто имеют динамическую структуру и реализуют сложное поведение. Сбор данных в таких системах является нетривиальной задачей. Предлагается модельный подход к построению систем сбора информации в многоуровневых киберфизических системах, построенных на платформах туманных вычислений. Модели предлагается строить в терминах знаний, для описания структуры и поведения использовать графовые модели, для управления моделями – многоуровневые относительно конечные автоматы. Отличительной особенностью предлагаемого подхода является возможность автоматического построения моделей.

Киберфизические системы, туманные платформы, системы сбора данных, модельное знание

Успехи в области микроэлектроники, телекоммуникаций и программной инженерии позволили выйти на новый уровень сложности создаваемых антропогенных систем различного назначения. Это может выражаться в увеличении числа уровней иерархии и числа подсистем, усложнении структуры связей между подсистемами и поведения системы и ее элементов, а также в повышении уровня интеллекта создаваемых систем. При этом четко проявляются тенденции к расширению сферы применения информационных технологий при построении самых разных классов систем.

Постоянно появляются новые концепции построения систем, новые классы систем и платформы, которые реализуют эти концепции. Эти новые концепции часто базируются на уже известных принципах и технологиях и в значительной степени пересекаются и дополняют друг друга. Примером могут служить киберфизические системы (Cyber-Physical Systems, CPS), в которые включаются не только программные компоненты, но и элементы самой разной природы (технологическое оборудование, транспортные системы, природные явления, биологические системы и т. п.), а также люди, кото-

рые отвечают за извлечение, накопление и использование знаний. В этом случае акцент делается на интеграцию разных технологий [1].

Другим характерным примером может служить парадигма окружающего интеллекта (Ambient Intelligence, AmI), в соответствии с которой создается среда, чувствительная и реагирующая на присутствие людей [2]. Основная идея AmI состоит в том, чтобы помочь людям в их повседневной деятельности, интуитивно используя информацию и интеллект, скрытые в сети, которая соединяет многочисленные устройства разного назначения. По мере того как эти устройства становятся все более компактными, дешевыми и взаимосвязанными, их технологическая структура становится невидимой для пользователя, которому доступен только интуитивно понятный интерфейс. Парадигма AmI и ее элементы уже сегодня активно используются при построении многих CPS и ориентированы на использование в больницах, на общественном транспорте, фабриках и др. При создании такого рода систем на первый план очень часто выступают требования минимальной стоимости.

Эту парадигму следует рассматривать прежде всего как интеграционную. Для ее практической реализации должно быть создано достаточное количество сервисов. AmI-подход интегрирует несколько хорошо известных подходов: контекстно-зависимые системы (Context Aware Systems, CAS), повсеместные вычисления (pervasive computing), системы, ориентированные на работу с профилями (profiling systems), системы с гибкой архитектурой (agile computing), когнитивные системы (cognitive computing). По замыслу разработчиков этой парадигмы AmI-системы (AmIS) должны строиться на базе связанных между собой встроенных CPS, совокупность которых должна функционировать как CAS. Эта система должна быть адаптивной, реализовывать проактивное поведение (anticipatory) и быть когнитивной, т. е. способной реализовывать элементы разумного поведения и обладать способностью к обучению. Если рассматривать перечисленные свойства как нефункциональные требования, то становится достаточно очевидным, что AmIS следует определить как контекстно-ориентированные, адаптивные (когнитивные) CPS [2].

При явной привлекательности идеи AmIS возникает естественный вопрос, каким образом можно реализовать перечисленную функцио-

нальность, реализация которой требует достаточно больших вычислительных ресурсов, при жестких ограничениях на стоимость создаваемых на базе этой концепции реальных систем. Достаточно эффективным решением данной проблемы является использование иерархических систем обработки информации.

Платформы для построения CPS. Современный этап развития информационных технологий характеризуется появлением новых технологий и платформ, которые создаются на их основе. К таким технологиям можно отнести, в частности, интернет вещей (Internet of Things, IoT), промышленный интернет вещей (Industrial Internet of Things, IIoT), интернет всего (Internet of Everything, IoE) [3], облачные и туманные вычисления, которые получают все большее распространение. CPS различного назначения все чаще строятся именно на этих платформах.

Наиболее популярной является платформа IoT [3], [4].

IoT – многоаспектное понятие. IoT можно рассматривать и как платформу, и как многоуровневое приложение. В данной статье IoT рассматривается как платформа.

Туманные структуры (Fog Systems, FS) [5] можно рассматривать как некоторую эталонную платформу [6], на которой могут быть реализованы AmI-системы.

FS – это сложные, многоуровневые системы с динамической структурой и адаптивным поведением, построенные из элементов разной физической природы, в которых кроме вертикальных связей всех уровней присутствуют многочисленные горизонтальные связи [5], [6]. Типовая структура FS показана на рис. 1.

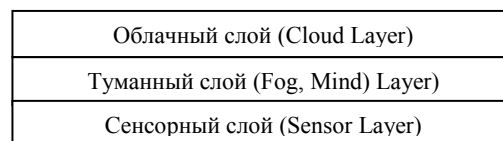


Рис. 1

На верхнем уровне располагается облачный слой (Cloud Layer), который строится на базе публичных (public), частных (private) или гибридных (hybrid) облаков [7]. Обычно это чисто виртуальный, автоматически масштабируемый ресурс, представленный пользователям в форме набора сервисов, достаточно широкого, но часто фиксированного. Отдельные приложения строятся как экосистемы, на базе одного поля сервисов. Вре-

мя доступа к сервисам определяется пропускной способностью каналов связи. Возможности управления облачным слоем со стороны пользователя обычно крайне ограничены.

Туманный уровень включает множество связанных между собой контроллеров, которые в полной мере доступны разработчику. На этом уровне в ограниченном объеме могут использоваться механизмы виртуализации, например облачка (cloudlets) [5], но чаще пользователь работает с физическими ресурсами. В настоящее время применительно к данному уровню наблюдаются две противоречивые тенденции. С одной стороны, уменьшение стоимости и повышение производительности позволяет использовать все более мощное контроллерное и телекоммуникационное оборудование, а с другой, расширение сферы применения FS идет по большей части за счет систем низкой ценовой категории. В этом контексте просматривается тенденция к разделению этого уровня на два – Fog Layer и Mind Layer. Fog Layer в полной мере доступен пользователю и может использовать механизмы виртуализации ресурсов, а Mind Layer работает с физическими ресурсами.

На самом нижнем (сенсорном) уровне располагаются датчики, исполнительные механизмы и средства связи, причем используемые на этом уровне встроенные контроллеры и средства связи становятся все более мощными и позволяют реализовать достаточно эффективное горизонтальное внутриуровневое взаимодействие.

Таким образом, задача разработки приложений, ориентированных на реализацию в туманной среде, представляет собой задачу оптимального, в некотором смысле, размещения модулей на элементах, принадлежащих трем перечисленным уровням.

Системы окружающего интеллекта как множество FS-приложений. FS – это платфор-

мы, на которых реализуются некоторые приложения или информационный компонент какой-либо CPS. В зависимости от сложности приложений можно выделить по крайней мере 3 уровня: отдельные приложения, доменно-ориентированные системы окружающего интеллекта (Ambient Intelligence Systems, AmIS) и системы, состоящие из многих систем, включая AmIS. На рис. 2 показана иерархия AmIS.



Рис. 2

Типовые задачи сбора данных в AmIS-системах. Для того чтобы реализовать процедуры, связанные с поддержкой контекстной ориентации, всеобъемлющих вычислений и когнитивности, необходимо иметь возможность реализовать эффективные процедуры сбора данных в условиях работы с динамической структурой и адаптивного поведения наблюдаемой CPS.

На рис. 3 показана укрупненная классификация задач сбора данных.

В первом случае мы имеем дело с CAS, во втором случае речь идет об адаптивной системе и в перспективе – о системе, способной к обучению (когнитивной системе). В третьем случае мы имеем дело со встроенной (обычно распределенной) системой управления, т. е. с CPS, когда в качестве потребителя собранной информации выступает исполнительная подсистема. При этом данные могут собираться либо об элементах системы (состояние, значения атрибутов), либо о протекающих бизнес-процессах.

Таким образом, можно говорить о трех основных парадигмах, используемых для построения систем сбора данных (ССД) в туманной сре-

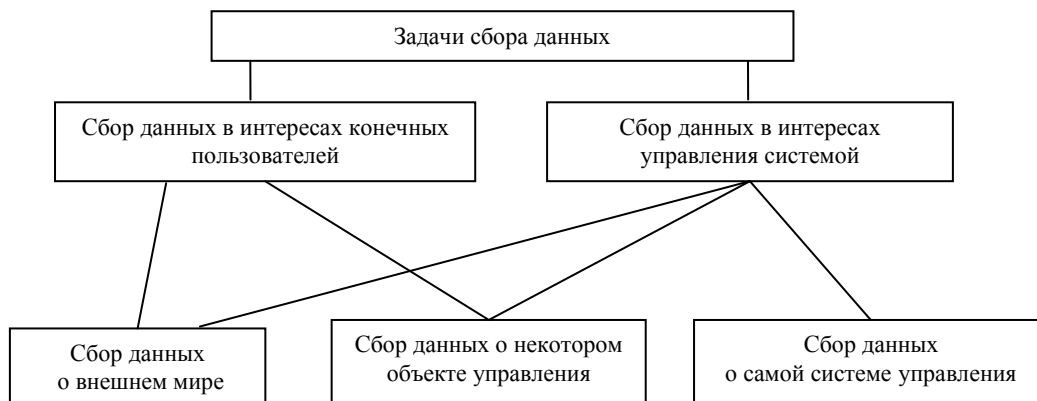


Рис. 3

де: 1) система сбора и отображения данных в требуемом виде, т. е. система трансформации представлений; 2) подсистема системы управления; 3) CAS.

Модели, основанные на трансформации представлений. Системы, основанные на трансформации представлений – это, по существу, системы, построенные по типу традиционных информационно-управляющих систем (ИУС), которые собирают данные о наблюдаемой системе и протекающих в ней бизнес-процессах (БП) и представление их заинтересованным сторонам в соответствии с их ролью. ИУС может быть представлена как $M_{ims} = \langle RQ, PRS, D, M_c, L, TR \rangle$, где RQ – множество запросов на выполнение действий по сбору информации; PRS – множество

представлений результатов запросов; D – множество точек съема данных; L – информация о событиях, поступающая в форме логов; TR – множество трансформаций.

В основе своей решаются 2 задачи: формирование политики сбора данных, формирование представлений.

Политика сбора данных определяет процедуру сбора данных. Для CPS с динамической структурой это часто приходится делать в процессе функционирования. Формирование представлений – это набор процедур (сервисов) для представления ответов на запросы в понятной для пользователя форме.

Формально модель формирования представления (МФП) можно определить как $M_{pf} = \langle TRM,$

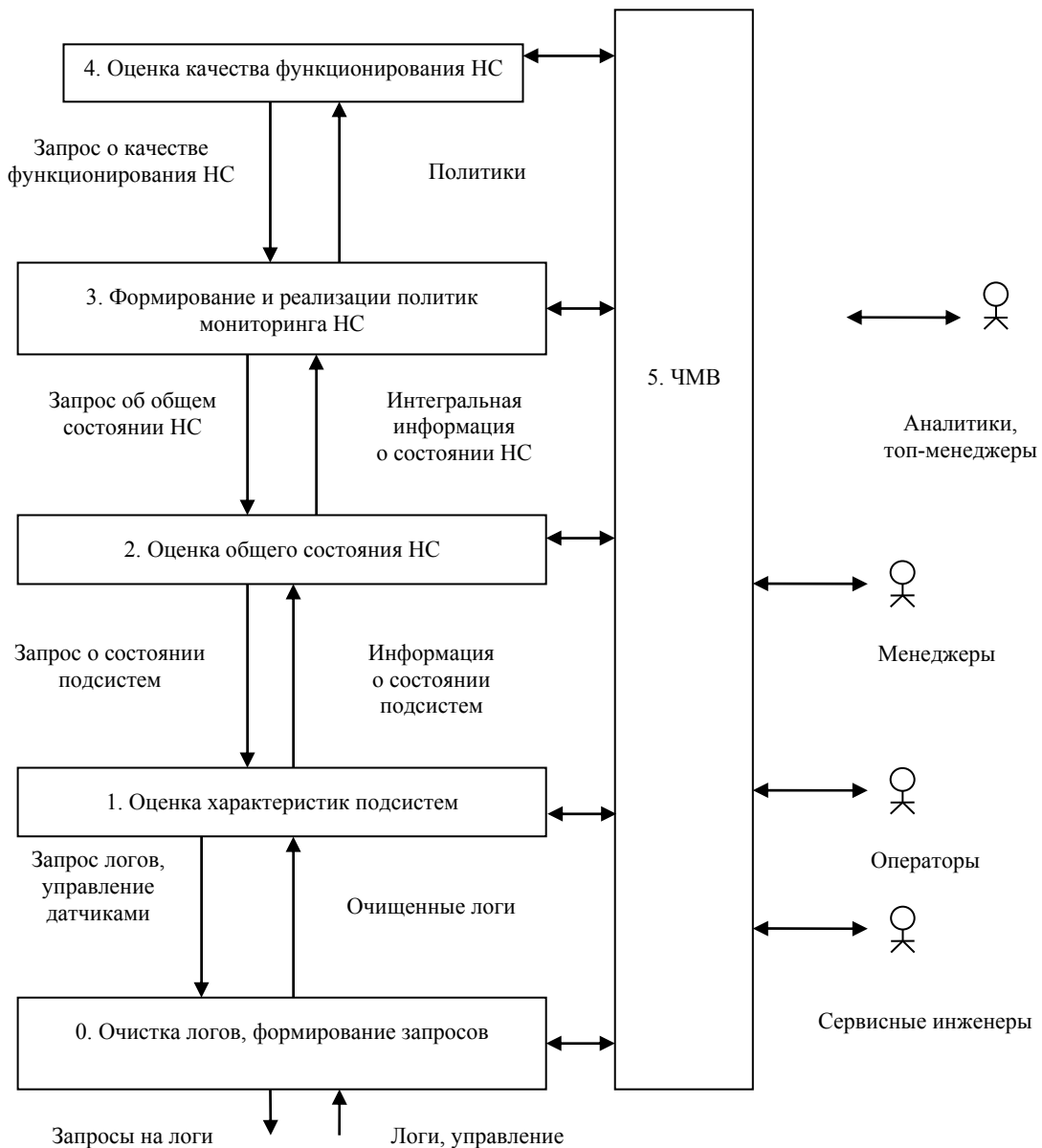


Рис. 4

DSL>, где TRM – множество моделей, построенных на базе модели M_{pf} , а DSL – множество языков общения с CM, ориентированных на разные категории заинтересованных сторон. Реализация МФП связана с реализацией набора трансформаций запросов на языке заинтересованной стороны в язык запроса модели и $DSL_i \xrightarrow{RQ} M$ и $M \xrightarrow{RS} DSL_i$. В качестве формальной модели, описывающей функционирование можно использовать, например, некоторую модификацию модели Joint Directors of Laboratories (JDL) [8], структура которой показана на рис. 4.

Фактически, эта модель определяет то, «что делается», а не то «как делается». JDL-модель – это функциональная модель. Она никогда не рассматривалась как процессная модель или как модель технической архитектуры.

Классическая JDL модель – это достаточно общая модель, которая напрямую не связана ни с одним из предметных доменов.

Предлагаемая модифицированная JDL-модель (MJDL), как и классическая, имеет 6 уровней. В число основных заинтересованных сторон входят 4 группы пользователей: сервисные инженеры (отвечают за техническое состояние инфраструктуры наблюдаемой системы (НС)); операторы (наблюдают за состоянием отдельных подсистем НС, в случае необходимости могут выдавать управляющие воздействия); менеджеры (отвечают за техническое состояние крупных подсистем или всей системы НС в целом), бизнес-аналитики и топ менеджеры, которых интересует, в первую очередь, общее состояние НС. В принципе, эти группы пользователей работают с разными уровнями модели. Пользователи взаимодействуют с системой через подсистему человеко-машинного взаимодействия (ЧМВ).

Уровень 0. Основные решаемые задачи – это формирование запросов на выдачу логов и обработка «сырых» логов; обработка, оценка и предсказание значений отдельных параметров. Можно говорить, что на этом уровне обрабатываются отдельные логи. Типичные проблемы уровня: исключение шумов (случайных логов), потери отдельных логов, невозможность снятия требуемого лога и т. п.

Уровень 1. Оценка характеристик отдельных подсистем НС. Основные решаемые задачи – оценка и предсказание значений отдельных параметров и состояния отдельных сущностей (объектов), входящих в состав НС. При этом реализу-

ются функции, связанные с формированием информации об отдельных объектах, на основе информации об элементах объектов. В качестве такой информации может рассматриваться, например, информация о техническом состоянии отдельных узлов сложной технической системы, месте нахождения объекта, скорости перемещения и направления движения.

Уровень 2. Оценка общего состояния НС. Основная задача – оценка и предсказание состояния НС. На этом уровне реализуются функции, связанные с формированием информации о ситуации в определенном контексте в терминах сущностей (подсистем), отношений между ними и событиями.

Уровень 3. Определение реакции. Этот уровень присутствует, если речь идет о системе мониторинга и управления. Основные решаемые задачи – оценка и предсказание будущих состояний системы и ее частей в терминах полезность/стоимость, формирование альтернативных вариантов генерации сигналов управления. На этом уровне осуществляется оценка ситуации, которая может включать оценку динамики ее развития, предположения о возможных действиях внешних по отношению к НС объектов, угрозах и собственных уязвимостях.

Уровень 4. Оценка эффективности. Основные решаемые задачи – оценка и предсказание характеристик эффективности как НС, так и самой ССД, и их сравнение с желаемыми показателями эффективности. На этом уровне реализуются функции мониторинга самой ССД, в частности с целью улучшения ее временных характеристик.

Уровень 5. Человеко-машинное взаимодействие. Реализуются функции, связанные с выполнением процедур ЧМВ. Кроме того, этот уровень отвечает за виртуализацию и коллективное принятие решений, на нем реализуются механизмы управления знаниями, в частности определяется, кто запрашивает информацию, кто имеет доступ к той или иной информации, с какой целью будет использована информация; определяется, в каком виде и в каком объеме она должна быть представлена заинтересованному лицу.

Каждый из уровней можно определить как $L_i = \langle M_i, T_{ij} \rangle$, где M_i – множество моделей, относящихся к i -му уровню, а T_{ij} – множество процедур трансформации моделей. В свою очередь, $T_{ij} = \langle Th_{ij}, Tv_{ij} \rangle$, где Th – процедуры горизон-

тальной трансформации; T_v – процедуры вертикальной трансформации. Процедуры вертикальной трансформации – это переходы между уровнями, а горизонтальные трансформации – переходы внутри уровней; обычно это переходы типа *Сырые данные* → *Информация* → *Знание*.

Обобщенная модель CAS [9]. В самом общем виде модель CAS можно представить как $M_{CAS} = \langle RQ, RS, D, L, PROC, M_c, M_r, \Delta \rangle$, где RQ – множество запросов на выполнение действий по сбору информации; RS – множество реакций HC; D – множество точек съема данных, PROC – множество процедур обработки; L – информация о событиях, поступающая в форме логов; M_c – текущая модель контекста; M_r – множество эталонных моделей контекста; Δ – процедура определения степени близости контекстов. В свою очередь, процедуру сбора данных можно определить как $PROC = \langle PROC_p, PROC_c \rangle$, где $PROC_p$ – множество процедур обработки, а $PROC_c$ – множество процедур управления (например, порядком обработки запросов).

Здесь имеется несколько вариантов:

1) адаптация алгоритма (процедур обработки) к контексту

$$PROC_{pef} = f(PROC_p, M_c),$$

где $PROC_{pef}$ – исполняемая процедура;

2) адаптация среды выполнения к контексту

$$PROC_{cef} = f(PROC_c, M_c);$$

3) реакция на изменение контекста – при изменении контекста формируется сигнал в форме лога, который запускает PROC.

Система сбора данных как элемент системы управления. Модель сбора данных в отдельной подсистеме CPS совпадает с описанной ранее моделью для CAS.

Формально поведения HC можно описать следующим образом. При работе в дискретном времени состояние HC описывается элементом x фазового пространства $X = \{x\}$, а его эволюция во времени описывается последовательностью x_1, x_i, x_n .

Динамика точки фазового пространства X в общем виде описывается соотношением $x_{t+i} = g(x^t)$, где t – время; при этом должны быть заданы начальные значения, по которым вычисляют последующие элементы последовательности. Пусть

указано пространство управлений $Y = \{y\}$. Управляемый объект (HC) характеризуется уравнением движения $x_{t+1} = g(x^t, y^t)$, $t \geq 1$. Кроме того, требуется добавить правила выбора управления, которые в общем случае зависят от предыстории и могут быть представлены в виде

$$y_{t+1} = f(x^t, y^{t-1}), t \geq 1.$$

Будем говорить, что выбор управлений реализует управляющая система, а систему правил $\{f\}$ называть стратегией или управлением. В дальнейшем эти термины используются как синонимы (уравнения движения управляемого объекта могут быть известны заранее либо неизвестны). Набор правил (стратегию) выбирают так, чтобы движение объекта в фазовом пространстве обладало тем или иным свойством. Требование, чтобы это свойство действительно существовало, называют целью управления.

Можно выделить несколько вариантов организации связи ССД и HC (рис. 5, а). В простейшем случае (рис. 5, а) ССД пассивно наблюдает за HC. Во втором случае HC может выдавать запросы на логи и, возможно, управляющие воздействия (рис. 5, б). В третьем случае ССД есть адаптивная система, которая может корректировать свое поведение (рис. 5, в); за коррекцию поведения отвечает регулятор Р. Можно представить случай, когда используется многоступенчатый регулятор (рис. 5, г) – тогда некоторый метарегулятор P2 корректирует поведение основного регулятора P1. Принципиально регулятор может иметь большее число уровней [10].

Очевидно, что реализовать функции интеллектуальной обработки данных можно, только задействуя практически неограниченные вычислительные возможности облачных структур. Таким образом, распределение функциональности между уровнями становится очевидным:

– сенсорный уровень отвечает за сбор сырых данных и, по возможности, за предварительную их обработку;

– туманный уровень управляет процессом сбора данных (в терминах политик); для решения подзадач идет обращение к сервисам облачного уровня;

– облачный уровень отвечает за хранение больших данных, моделей, метамodelей и предоставляет сервисы обработки туманному уровню.

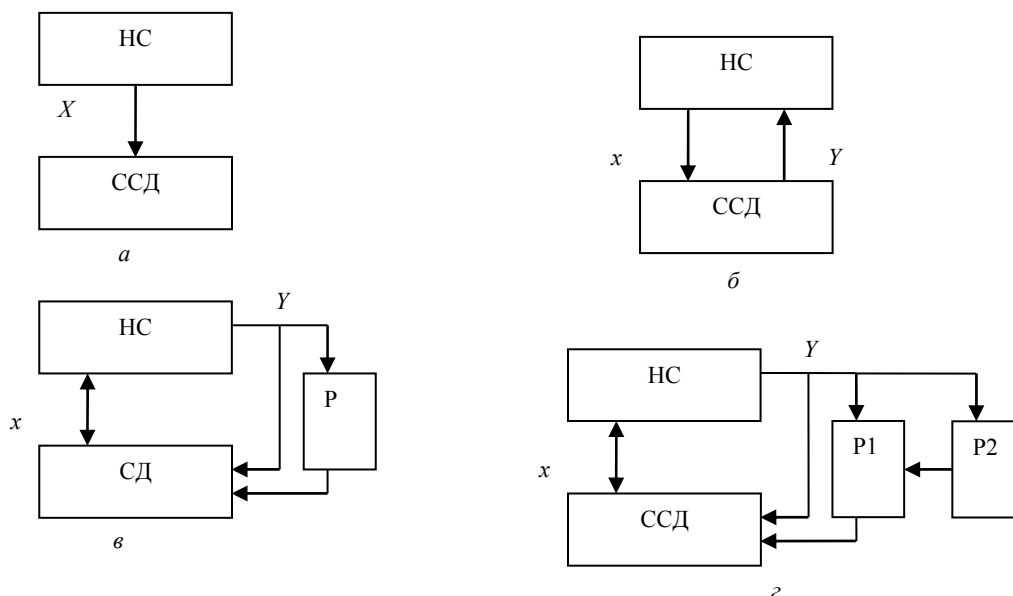


Рис. 5

Работа со знаниями и проблема больших данных. Дешевизна датчиков приводит к спонтанному росту их количества и объема поставляемых сырых данных. Хотя доступные объемы хранилищ могут содержать очень большие объемы данных, пределы существуют. При этом для нахождения и обработки данных требуются все более мощные вычислительные ресурсы. Одно из возможных решений проблем больших данных – это извлечение из них знаний и хранение только знаний (пирамида знаний). Возникает вопрос о том, какие именно знания следует извлекать и хранить.

Существует достаточно много типов знаний и подходов к их классификации. На верхнем уровне чаще всего говорят о таких типах знаний, как процедурное знание (Procedural knowledge), декларативное знание (Declarative knowledge), метазнание (Metaknowledge), структурное знание (Structural knowledge), неточное и неопределенное знание (Inexact and uncertain knowledge), здравый смысл (Commonsense knowledge), онтологическое знание (Ontological knowledge) [11].

На более низких, в частности доменных, уровнях могут появляться другие типы знаний – модельное, архитектурное и т. д. Для их представления обычно используются способы представления знаний верхнего уровня.

Для решения задач, связанных с построением AmIS, существенный практический интерес представляет *модельное знание*, которое можно определить как модель, построенную в терминах знаний.

В ОИ-системах могут применяться различные модели, в частности, это могут быть модели, используемые на этапе проектирования, и модели, используемые во время выполнения программ (*models@run.time*). Нас сейчас интересуют *run time-модели*.

При реализации процесса сбора данных, то в явном или неявном виде модель системы, о которой собираются данные, присутствует. В данной работе предлагается использовать явную модель, построенную в терминах знаний.

Предлагаемый подход. Идея состоит в использовании механизмов работы с модельным знанием:

1. Строится система моделей, описывающая структуру и поведение наблюдаемой системы (НС), в качестве которой, в частности, может выступать сама система.
2. Модель строится в терминах знаний.
3. Актуальность модели поддерживается с помощью процедур мониторинга.
4. Все запросы всех заинтересованных сторон осуществляются только к моделям.

При определенных условиях использование моделей, построенных в терминах знаний, позволяет получить ряд полезных свойств:

- 1) возможность оперативно отслеживать динамику структуры и поведения наблюдаемой CPS, состоящей из элементов разной физической природы, включая людей (социо-физические системы);

2) возможность уменьшить время отклика на запросы пользователей о состоянии наблюдаемой системы, поскольку нет необходимости каждый строить запросы к самой наблюдаемой системе (имеем своеобразный 2-ступенчатый конвейер;

3) возможность оперативно получать данные о прошлых состояниях и в определенных пределах предсказывать поведение наблюдаемой системы;

4) если речь идет о реализации механизмов *само** (самодиагностика, самовосстановление и т. д.), то модель позволяет накапливать знания о самой системе, т. е. создаются предпосылки для реализации механизмов когнитивности.

Управление в терминах моделей. Это достаточно распространенный подход в рамках информационных систем, но управление в терминах модельного знания пока находит ограниченное применение. Если речь идет о CPS, то использование модельного подхода – естественное решение. Более того, поскольку в CPS используются элементы разной физической природы, то приходится работать с моделями, представленными в терминах знаний. Этот подход используется при построении киберфизических систем, функционирование которых описывается с помощью моделей, напрямую не привязанных к конкретной реализации.

Для того чтобы указанный положительный эффект мог быть получен, модели должны обладать определенными свойствами:

1) описывать структуру и поведение наблюдаемой системы произвольной сложности с иерархической структурой;

2) отвечать на требуемые запросы пользователей при разумных затратах ресурсов;

3) возможность автоматического построения и реконфигурации при разумных затратах ресурсов.

Операции над моделями. Современные системы IoT и IoE – это распределенные многоуровневые системы. На верхнем уровне обычно

имеем более или менее жесткую структуру, а на нижнем – изменяющуюся структуру связей. Каждый элемент и каждая подсистема описываются собственными моделями.

Можно определить следующие основные типы операций над моделями: построение модели, перестройка модели, выдача требуемого представления, слияние моделей.

Модельный подход как разновидность виртуализации. Модельный подход можно рассматривать как разновидность виртуализации. Чаще всего используется виртуализация ресурсов, которая применяется повсеместно, в частности в облачных структурах (рис. 6), и предполагает, что конечный пользователь работает не с физическим ресурсом, а с его моделью (рис. 7).

Основное различие состоит в том, что используются разные типы моделей. В первом случае речь идет о программной модели некоторого внутреннего ресурса, для обращения к которому могут использоваться системные вызовы, аналогичные системным вызовам для работы с физическими ресурсами. Во втором случае реализуется модель некоторой внешней сущности, для работы с которой необходимо создать собственный набор системных вызовов, аналогичный набору вызовов для реализации процедуры сбора данных. Кроме того, необходимо иметь набор процедур для построения и поддержания в актуальном состоянии самой модели.

При построении модели НС в терминах знаний могут применяться разные платформы. В частности, могут использоваться онтологии или графы знаний [11]. Каждый из этих вариантов имеет свои достоинства и недостатки. В общем случае предпочтительным представляется онтологический подход, поскольку для этой платформы имеется доступный инструментарий.

Когнитивные возможности предлагаемого подхода. Термин «когнитивность» в широком смысле определяет то, как человек думает, при-

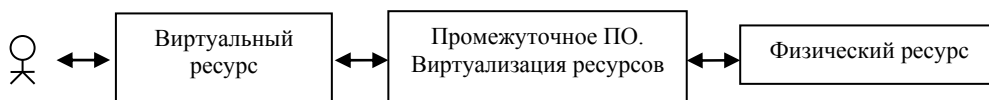


Рис. 6

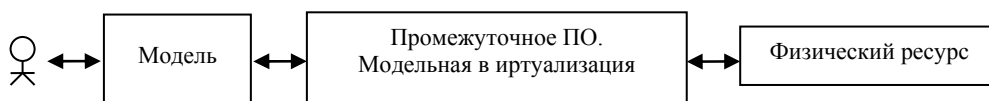


Рис. 7

нимает решения, решает задачи, учится. Под когнитивными обычно понимают системы, использующие когнитивные архитектуры, которые имитируют используемые человеком эвристические алгоритмы. Применительно к CPS можно говорить о трех аспектах когнитивности: НС является когнитивной системой, ССД является элементом когнитивной системы и ССД реализует когнитивные технологии сбора данных.

Если НС – когнитивная система, то естественно предположить, что поведение, а возможно, и структура НС постоянно меняется. Использование предлагаемого подхода позволяет оперативно отслеживать динамику поведения и структуры НС. Эта информация может быть использована как для контроля функционирования НС, так и для ее обучения. При этом сама ССД может не быть когнитивной системой,

ССД может входить в состав системы на правах подсистемы, отвечающей за получение информации о состоянии НС. В таком случае в зависимости от постановки задачи сбора данных НС может быть как когнитивной, так и не когнитивной.

Построение когнитивных ССД связано с возможностью совершенствования процедуры сбора данных посредством обучения. Для этого могут применяться различные алгоритмы машинного обучения и, в частности, нейронные сети.

Предлагаемый в данной работе подход и модели могут быть с успехом использованы для реализации процесса сбора данных в AmIS, включающих в себя в качестве элементов CPS, построенных на платформах туманных вычислений. Модельный подход к построению ССД в таких системах позволяет выйти на новый уровень сложности информационных систем, который недостижим при использовании традиционных подходов. Следует заметить, что ожидаемый переход к когнитивным CPS сделает задачу мониторинга их состояния еще более сложной, и ее можно будет решить только с использованием модельного подхода, который может найти применение в самых разных предметных доменах.

СПИСОК ЛИТЕРАТУРЫ

1. Sanfelice R. G. Analysis and design of cyber-physical systems. A hybrid control systems approach / D. Rawat, J. Rodrigues, I. Stojmenovic // *Cyber-Physical Systems: From Theory to Practice*. Boca Raton, FL: Taylor & Francis Group, 2016.
2. Grand challenges for ambient intelligence and implications for design contexts and smart societies / N. Streit, D. Charitos, M. Kaptein, M. Böhlen // *J. of Ambient Intelligence and Smart Environments*. 2019. № 11. С. 87–107.
3. Huang J., Hua K. Managing the Internet of Things: architectures, theories and applications / *The Institution of Engineering and Technology*. UK: Hertz, 2016.
4. Hallaq B., Cunningham J., Watson T. The industrial internet of things (IIoT): An analysis framework // *Computers in Industry*. 2018. Vol. 101. P. 1–12.
5. Mahmood Z. Fog computing concepts, frameworks and technologies. Cham, Switzerland: Springer International Publishing AG, 2018.
6. IEEE Standard Association. FOG – Fog Computing and Networking Architecture Framework. URL: <http://standards.ieee.org/develop/wg/FOG.html> (дата обращения 17.05.2020).
7. Hwang K., Fox G. C., Dongarra J. J. *Distributed and cloud computing waltham*. Elsevier Inc., 2012.
8. Blasch E., Bosse E., Lambert D. *High-level information fusion management and system design*. Norwood, MA: Artech House Publishers, 2012.
9. Loke S. *Context-aware pervasive systems. Architectures for a new breed of applications*. Boca Raton, FL: Taylor & Francis Group, 2007.
10. Срагович В. Г. *Адаптивное управление*. М.: Наука, 1981.
11. Рассел С., Норвиг П. *Искусственный интеллект: современный подход: пер. с англ. 2-е изд.* М.: Издательский дом «Вильямс», 2006.

S. A. Abbas, A. I. Vodyaho
Saint Petersburg Electrotechnical University

N. A. Zhukova
St. Petersburg Federal Research Centre of the Russian Academy of Sciences (SPCRAS)

M. A. Chervontsev
Saint Petersburg Research & Engineering Center

Aung Myo Thaw
ITMO University

ABOUT ONE APPROACH TO BUILDING DATA COLLECTION SYSTEMS IN CYBER-PHYSICAL SYSTEMS BUILT ON FOG COMPUTING PLATFORMS

The modern stage of development of technology is characterized by an increase in the complexity of created anthropogenic systems, a constant expansion of the scope of application of information technologies and an increase in the intellectual level of created anthropogenic systems and the emergence of new paradigms for building information-oriented systems such as cyber-physical systems, the Internet of things, cloud and fog systems. Modern information-oriented systems often have a dynamic structure and implement complex behavior. Data collection in such systems is a non-trivial task. The paper proposes a model approach to building information collection systems in multilevel cyber-physical systems built on fog computing platforms. Models are proposed to be built in terms of knowledge. To describe the structure and behavior graph models are used. It is proposed to use multi-level relatively finite state automata to control the models. A distinctive feature of the proposed approach is the ability to automatically build models.

Cyber-physical systems, fog platforms, data collection systems, model knowledge

УДК 681.5; 622.691.4.053

Д. Х. Имаев, М. Ю. Шестопалов
Санкт-Петербургский государственный электротехнический
университет «ЛЭТИ» им. В. И. Ульянова (Ленина)

С. В. Квашнин
АО «НПФ „Система-Сервис“»

Моделирование технологического процесса компримирования природного газа как объекта управления. Переходные процессы*

Предложена упрощенная модель динамики компрессорной линии, реализующая управляемый технологический процесс сжатия природного газа. Компрессорная линия образована газоперекачивающим агрегатом и сетью на его выходе. Процедуры моделирования и идентификации представляются в виде последовательного увеличения информации о причинно-следственной топологии, структурах блоков преобразования сигналов, параметрах объекта и взаимосвязи объекта с окружающей средой. Исследовано влияние инерции и упругости объема газа на переходные процессы в компрессорной линии. Проиллюстрирована процедура параметрической идентификации нелинейной компьютерной модели компрессорной линии на основе архивных данных реальной компрессорной станции. Линеаризованные модели используются для вынесения суждений о свойствах устойчивости, управляемости и наблюдаемости компрессорной линии.

Природный газ, компримирование, управление, переходный процесс, идентификация, компьютерная имитация, линеаризация, устойчивость, управляемость, наблюдаемость

Целью управления компрессорными линиями магистральных газопроводов является поддержа-

ние заданных параметров перекачиваемого природного газа в условиях нежелательного влияния среды функционирования. В настоящее время успешно эксплуатируются распределенные систе-

* Окончание, начало в № 6/2020.