

УДК 519.713

doi:10.15217/issn1684-8853.2017.5.80

# ОНТОЛОГИЧЕСКОЕ МОДЕЛИРОВАНИЕ УПРАВЛЯЮЩИХ АВТОМАТОВ СЕРВИСОВ КИБЕРФИЗИЧЕСКИХ СИСТЕМ

Н. Г. Шилов<sup>а</sup>, канд. техн. наук, старший научный сотрудник

М. С. Щекотов<sup>а</sup>, научный сотрудник

<sup>а</sup>Санкт-Петербургский институт информатики и автоматизации РАН, Санкт-Петербург, РФ

**Постановка проблемы:** сервисы киберфизических систем наделяются не только возможностью выхода в Интернет в целях поиска и извлечения информации, а также ее предоставления, но и принятия решений в зависимости от текущей ситуации. Ввиду ограниченной вычислительной мощности таких устройств использование для их описания и программирования управляющих автоматов является весьма перспективным. Однако необходимость их тесного взаимодействия вызывает проблемы, связанные с обеспечением интероперабельности. **Цель исследования:** онтологическое моделирование управляющих автоматов сервисов киберфизических систем. **Результаты:** предложено использование управляющих автоматов для разработки сервисов киберфизической системы; разработана совместимая с Resource Description Framework онтологическая модель управляющего автомата сервиса киберфизической системы, основанная на представлении сервисов, их состояний и выходных символов в виде классов, а входных символов — в виде отношений; результаты продемонстрированы на примере сценария взаимодействия роботов при формировании заданного слова из букв. **Практическая значимость:** полученные результаты могут быть использованы при разработке сервисов киберфизических систем.

**Ключевые слова** — интероперабельность, онтологическое моделирование, управляющий автомат, сервис, киберфизическая система.

## Введение

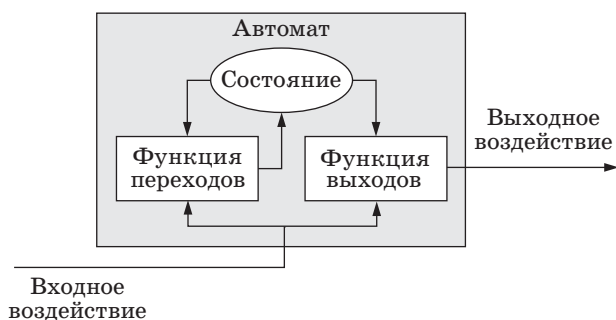
В эпоху Интернета вещей все больше устройств наделяются функциональностью, позволяющей им осуществлять доступ к Интернету. Данная функциональность была ориентирована в первую очередь на поиск и извлечение информации, однако развитие информационно-коммуникационных технологий привело к появлению новых требований и возможностей. Многие современные устройства помогают человеку в различных видах его деятельности, не только извлекая информацию, но и анализируя ее, а также принимая собственные решения в зависимости от текущей ситуации [3, 4]. Данные возможности стали одной из движущих сил развития киберфизических систем [1, 2]. Киберфизические системы представляют собой множество ресурсов, взаимодействующих между собой в информационном пространстве и управляющих устройствами в физическом пространстве в режиме реального времени. При этом они базируются на инфраструктурах, обеспечивающих связь, вычисления, управление и объединяющих сенсоры, вычислительные устройства, сервисы и средства коммуникаций. Устройства, входящие в киберфизические системы, зачастую имеют ограниченные вычислительные мощности, что существенно сужает возможности их программирования. Одним из перспективных подходов к решению данной проблемы является использование управляющих автоматов.

Теория автоматов появилась в рамках теории управляющих систем (теоретической кибернетики) в связи с бурным развитием средств электронной вычислительной техники и соответствующих областей математического знания в первую очередь для моделирования гипотетических цифровых систем. Однако последующее развитие информационных технологий вывело сферу приложения теории автоматов далеко за рамки моделирования аппаратных средств цифровой электроники, расширив ее до фундаментальных основ современной теоретической информатики. Сегодня абстракции и модели, разработанные в теории автоматов, востребованы такими научными дисциплинами, как теория формальных грамматик, математическая лингвистика, теория логических моделей, математическая логика и формальные аксиоматические системы, теория кодирования, теория вычислительной сложности и др. [8].

Под управляющим автоматом принято понимать конечный дискретный автомат, описываемый как

$$A = (X, Y, Z, \delta, \varphi, y_0),$$

где  $X$  — конечный алфавит входных символов;  $Y$  — конечное множество состояний;  $Z$  — конечный алфавит выходных символов;  $\delta: X \times Y \rightarrow Y$  — функция переходов, определяющая состояние;  $\varphi: X \times Y \rightarrow Z$  — функция переходов, определяющая выходной символ;  $y_0 \in Y$  — начальное (стартовое) состояние.



■ **Рис. 1.** Управляющий автомат  
 ■ **Fig. 1.** Finite-state machine

Схематично такой автомат изображен на рис. 1 [10].

Теория автоматов имеет широкие возможности применения [9]:

- проектирование систем логического управления;
- обработка текстов и построение компиляторов;
- спецификация и верификация систем взаимодействующих процессов;
- языки описания документов и объектно-ориентированных программ;
- оптимизация логических программ и т. д.

Одним из достаточно популярных направлений в данной области является автоматное программирование, т. е. подход к разработке программных систем со сложным поведением, основанный на модели автоматизированного объекта управления (расширении конечного автомата) [10].

Необходимость тесного взаимодействия устройств киберфизических систем также накладывает дополнительные требования, связанные с обеспечением интероперабельности. Одним из наиболее распространенных подходов к решению данной проблемы является онтологическое моделирование проблемной области [5–7]. В настоящей статье предложено решение задачи онтологического моделирования управляющих автоматов сервисов киберфизических систем.

### Сервис киберфизической системы

Согласно ранее разработанной концепции [11], сервис киберфизической системы можно представить следующим образом (рис. 2).

**Сервис** — основной действующий субъект процесса многоуровневой самоконтекстуализации. Сервис может представлять сенсор, транспортное средство, компанию — поставщика услуг и т. п. Сервис обладает структурными знаниями, параметрическими знаниями и профилем. Сервис характеризуется такими свойствами, как самокон-

текстуализация, самоадаптация, автономность и проактивность и выполняет некоторые действия.

*Структурные знания* являются концептуальным описанием проблем, которые могут быть решены соответствующим сервисом. Они представляют собой внутреннюю онтологию сервиса. Структурные знания описывают структуру параметрических знаний сервиса. В зависимости от ситуации структурные знания могут быть модифицированы (адаптированы) посредством *самоадаптации*. Они также описывают терминологию контекста и профиля сервиса.

*Параметрические знания* являются знаниями о конкретной ситуации. Их структура определяется структурными знаниями сервиса, а параметрическая составляющая зависит от контекста текущей ситуации. Они определяют поведение сервиса.

*Контекстом* называется любая информация, которая может быть использована для описания ситуации объекта, где в качестве объекта может выступать человек, место и т. п., которые считаются относящимися к задаче, решаемой пользователем, включая самого пользователя и используемые им инструментальные средства [13].

Целью контекста является представление только релевантных информации и знаний из всего объема доступных. Релевантность информации и знаний оценивается на основе того, насколько они связаны с рассматриваемой задачей. Контекст описывается в терминологии структурных знаний сервиса. Он обновляется по мере поступления информации из окружения сервиса и как результат действий сервиса. Контекст обновляет параметрические знания сервиса, которые в свою очередь определяют его поведение. Способность системы (сервиса) описывать и использовать контекст, а также адаптировать свое поведение в зависимости от контекста называется *самоконтекстуализацией* [14]. Представленный подход использует идею самоконтекстуализации для автономной адаптации поведения сервисов согласно контексту текущей ситуации для принятия ими контекстно-зависимых решений. С этой целью предусмотрена концептуальная модель предусматривает контекстную зависимость сервисов и их адаптацию к контексту. В работе используется двухэтапное построение контекста: построение контекста текущей ситуации на основании знаний проблемной области (абстрактный контекст) и конкретизация абстрактного контекста значениями переменных, получаемыми от информационных сервисов (оперативный контекст).

*Окружение* является окружающей средой киберфизической системы, частью которой является рассматриваемый сервис. Такая среда способна взаимодействовать с данной системой. Окружение влияет на контекст сервиса. Сервис может воздействовать на окружение, если он имеет соответ-



ведения сервиса в целях достижения рациональных результатов на более низких уровнях самоконтекстуализации.

*Предпочтения* — это склонность сервиса к определенным состояниям окружения или собственным состояниям, а также склонность избегать некоторых состояний. Предпочтения описаны в профиле сервиса и влияют на его поведение. Агент может изменять свои предпочтения посредством самоадаптации.

*Стратегия* — это предопределенный план действий или набор правил выбора действий для изменения состояния сервиса или его окружения с текущего на желаемое. Стратегия описывается в профиле сервиса и определяет его поведение. Сервис может изменять свои стратегии посредством самоадаптации.

*Действия* — это способность сервиса взаимодействовать с другими сервисами и вести переговоры посредством своего поведения. Они регулируются протоколом переговоров и нормами поведения.

*Протокол переговоров* — это набор базовых правил, сформулированных таким образом, чтобы при следовании сервисом этим правилам вся система вела себя так, как задумано ее создателями. Протокол переговоров определяет действия сервиса.

*Нормы поведения* — это правила, регулирующие действия сервиса. В отличие от протокола переговоров, нормы носят рекомендательный характер.

### Использование управляющих автоматов для описания поведения сервисов

Как было показано ранее, описание поведения сервиса в киберфизической системе с помощью управляющих автоматов может быть весьма целесообразным. Можно сформулировать следующие составляющие управляющего автомата сервиса:

— конечный алфавит входных символов ( $X$ ) — сообщения, которые может получать сервис от других сервисов;

— конечное множество состояний ( $Y$ ) — совокупность значений переменных, описывающих состояние сервиса и окружения, для которой может быть определено значение «желаемости» с точки зрения сервиса;

— функции переходов ( $\delta$  и  $\varphi$ ) — функции, описывающие действия сервиса в зависимости от параметров текущей ситуации;

— конечный алфавит выходных символов ( $Z$ ) — сообщения, которые сервис может посылать другим сервисам.

Рассмотрим простой, но иллюстративный пример киберфизической системы, являющейся под-

системой системы «умный дом» (рис. 3). Допустим, в наличии имеется датчик температуры ( $S_1$ ) и обогреватель ( $S_3$ ), представленные в системе с помощью сервисов, а также управляющий сервис ( $S_2$ ). Также в системе присутствуют два типа сообщений для обмена информацией: сообщения с данными о текущей температуре в помещении ( $m_1$ ); сообщение с управляющим воздействием, направленным на изменение режима работы обогревателя ( $m_2$ ).

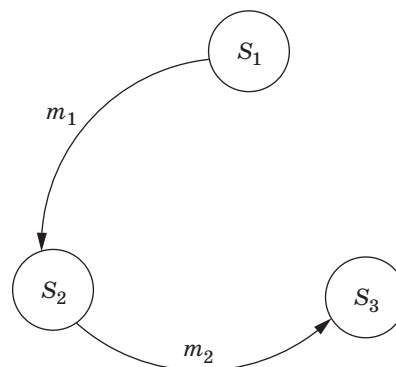
Рассмотрим автомат сервиса  $S_2$ .

У него существует всего четыре состояния:

$$Y = (y_0, y_1, y_2, y_3),$$

где  $y_0$  — температура ( $x$ ) не определена (начальное состояние сервиса);  $y_1$  — температура ниже необходимой;  $y_2$  — температура соответствует необходимой ( $x_0$ );  $y_3$  — температура выше необходимой.

Алфавит входных символов соответствует целым числам от  $-127$  до  $127$  (температура в градусах Цельсия) и передается в сообщении  $m_1$ :  $X = [-127, -126, \dots, 127]$ .



■ **Рис. 3.** Пример описания взаимодействия сервисов киберфизической системы управления температурой помещения

■ **Fig. 3.** The example of the interaction of cyber-physical system services for smart home temperature management

■ **Таблица 1.** Пример функции переходов  $\varphi$  управляющего сервиса умного дома

■ **Table 1.** The transition function  $\varphi$  of the smart home service state machine

Состояние ( $Y$ )	Управляющее воздействие ( $Z$ )
$y_1$	1
$y_2$	0
$y_3$	-1

Алфавит выходных символов соответствует целым числам  $-1$  (снизить температуру),  $0$  (не менять температуру),  $1$  (повысить температуру) и передается в сообщении  $m_2: Z = [-1, 0, 1]$ .

Функция переходов  $\delta$  выглядит следующим образом:

$$\delta(x): y = \begin{cases} y_1, & x < x_0 \\ y_2, & x = x_0 \\ y_3, & x > x_0 \end{cases}$$

Функция переходов  $\varphi$  представлена в табл. 1.

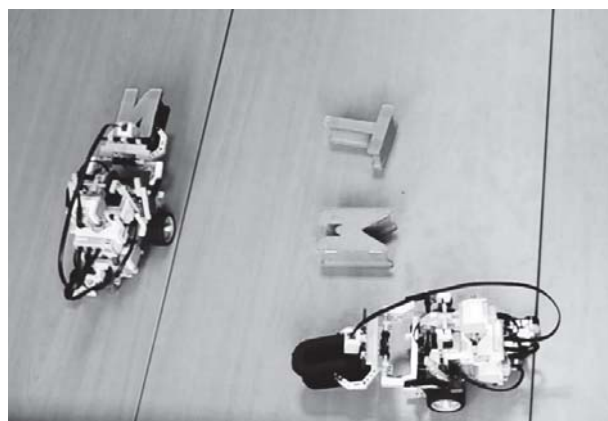
### Онтологическое описание управляющих автоматов сервисов на основе модели представления данных RDF

Поскольку в разработанной ранее концепции построения сервисов киберфизических систем для их описания используются онтологии, а для передачи сообщений — модель представления данных RDF (Resource Description Framework — «среда описания ресурса» — модель, разработанная консорциумом W3C) [11, 12], необходимо определить способ сопоставления элементов управляющих автоматов и элементов онтологии с использованием RDF. Данная модель представляет утверждения о ресурсах в пригодном для машинной обработки виде и является частью концепции семантической паутины (Semantic Web).

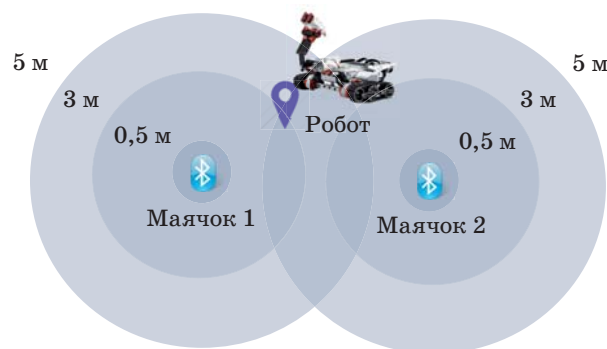
В качестве примера рассмотрим сценарий формирования роботами заданного слова из букв, произвольно размещенных на столе [12]. Для упрощения исследовательского макета роботы не распознают буквы, а определяют их на основе цветов (с использованием простых датчиков цвета), т. е. каждой букве в соответствие поставлен определенный цвет, в который она окрашена. Согласно сценарию, роботы сканируют пространство, находят буквы и складывают из них заданное слово (рис. 4).

Для определения местоположения роботов используется метод счисления координат на основе данных следующих встроенных датчиков: акселерометра, гироскопа, компаса. В начале сценария робот находится в точке с известными координатами, откуда и начинает движение. Также производится коррекция с помощью сигналов маяков BLE (Bluetooth Low Energy) [15, 16]. Использование маяков BLE позволяет определить, в какой из зон действия сигнала какого маяка находится робот (рис. 5).

Для обмена информацией используется интеллектуальное пространство, являющееся распределенным хранилищем информации. Сервисы



■ Рис. 4. Роботы складывают слово «ИТМО»  
 ■ Fig. 4. Robots are assembling the word «ИТМО»

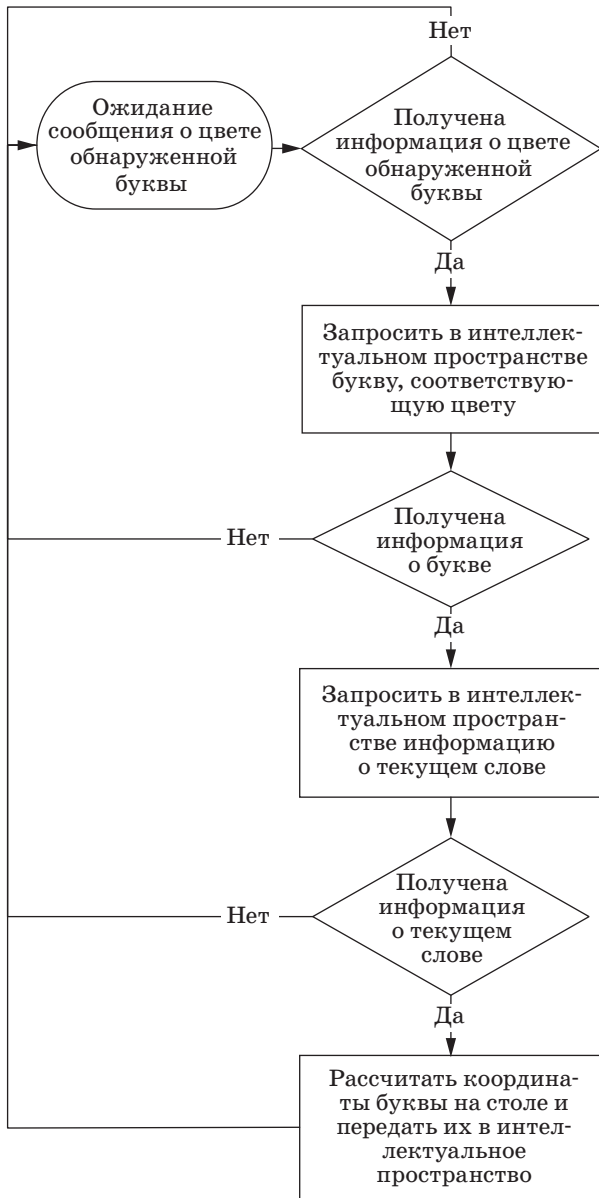


■ Рис. 5. Определение зоны местонахождения робота  
 ■ Fig. 5. The robot localization

могут передавать информацию в интеллектуальное пространство и получать ее посредством запросов.

За текущее состояние (расположение букв на столе) и управление формированием слова отвечает сервис локализации букв (letter localization service). Данный сервис собирает информацию от роботов, рассчитывает, где должны оказаться буквы, и отдает соответствующие распоряжения роботам.

Диаграмма разработанного алгоритма работы сервиса локализации букв представлена на рис. 6 и подробно описана в статье [12]. Сервис подписывается на получение сообщений об обнаружении новой буквы и переходит в режим ожидания. При появлении информации о новой букве сервис получает информацию о ее цвете и на ее основе осуществляет запрос идентификации буквы. После идентификации буквы сервис осуществляет запрос текущего слова, которое должны собрать роботы. Если текущее слово успешно получено, сервис находит в нем расположение буквы и рассчитывает на столе ее координаты, которые



■ **Рис. 6.** Диаграмма алгоритма работы сервиса локализации букв (адаптировано из работы [12])  
 ■ **Fig. 6.** The diagram of letter localization service operation algorithm (adapted from [12])

передаются роботам. Данную диаграмму в виде автомата представляет рис. 7, ему соответствует таблица состояний (табл. 2).

Дадим описание входных символов, состояний и выходных символов.

Входные символы:

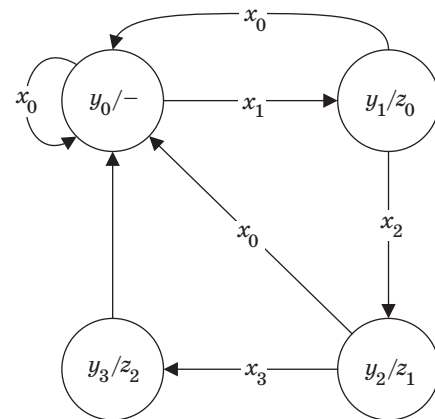
- $x_0$  — пустой или нераспознаваемый символ;
- $x_1$  — цвет обнаруженной буквы;
- $x_2$  — буква;
- $x_3$  — слово.

Состояния:

$y_0$  — ожидание сообщения о цвете обнаруженной буквы;

■ **Таблица 2.** Таблица состояний автомата сервиса локализации букв  
 ■ **Table 2.** The state table of the letter localization service state machine

$Y$	$y_0$	$y_1$	$y_2$	$y_3$
$Z$	—	$z_0$	$z_1$	$z_2$
$x_0$	$y_0$	$y_0$	$y_0$	—
$x_1$	$y_1$	—	—	—
$x_2$	—	$y_2$	—	—
$x_3$	—	—	$y_3$	—



■ **Рис. 7.** Автомат сервиса локализации букв  
 ■ **Fig. 7.** The state machine of the letter localization service

- $y_1$  — ожидание сообщения о букве;
  - $y_2$  — ожидание сообщения о слове;
  - $y_3$  — выполнение  $z_2$  и переход в состояние  $y_0$ .
- Выходные символы:

- $z_0$  — запрос в интеллектуальное пространство об определении буквы на основании заданного цвета;
- $z_1$  — запрос в интеллектуальное пространство о текущем слове;
- $z_2$  — определение координат буквы и их передача в интеллектуальное пространство.

Данный автомат можно представить в виде RDF-онтологии. Согласно RDF, любое утверждение имеет вид «субъект — предикат — объект» и называется триплетом. Для обозначения субъектов, отношений и объектов в RDF используются URI (Uniform Resource Identifier — унифицированный (единообразный) идентификатор ресурса).

Очевидным способом представления является создание классов Service для описания сервисов, State — состояний, is input for — входных символов и Output — выходных символов, а также их

экземпляров. Однако такое представление не позволяет описывать утверждения вида  $\langle y_0, x_1, y_1 \rangle$  («входной символ  $x_1$  переводит сервис из состояния  $y_0$  в состояние  $y_1$ »). В результате было предложено представить входные символы с помощью отношений, являющихся дочерними для более общего отношения is input for, связывающего состояния, т. е. его экземплярами (табл. 3, 4). Отношения RDF на уровне классов, соответствующие элементам автомата, показаны на рис. 8. В табл. 5 представлены отношения между экземплярами классов онтологии, соответствующие автомату сервиса локализации букв.

■ Таблица 3. Соответствие классов, отношений и экземпляров RDF элементам автомата

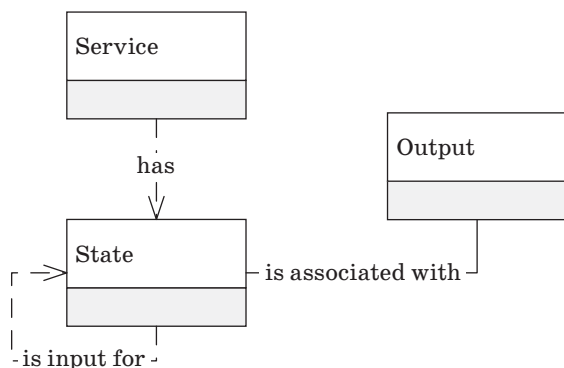
■ Table 3. Correspondence of RDF's concepts, properties and instances to state machine elements

Класс	Экземпляры
Сервис (Service)	Letter localization service
Состояние (State)	$y_0, y_1, y_2, y_3$
Выходной символ (Output)	null, $z_0, z_1, z_2$
Отношение «Входной символ» (is input for)	null, $x_0, x_1, x_2, x_3$

■ Таблица 4. Отношения RDF на уровне классов, соответствующие элементам автомата

■ Table 4. RDF properties corresponding to state machine elements

Отношение	Класс 1	Класс 2
has	Service	State
is associated with	State	Output
is input for	State	State



■ Рис. 8. Диаграмма отношения RDF на уровне классов, соответствующего элементам автомата

■ Fig. 8. RDF properties diagram corresponding to state machine elements

■ Таблица 5. Отношения между экземплярами классов RDF, соответствующие автомату сервиса локализации букв

■ Table 5. RDF triples corresponding to letter localization service state machine

Класс 1	Отношение	Класс 2
Letter localization service	has	$y_0$
То же	То же	$y_1$
- " -	- " -	$y_2$
- " -	- " -	$y_3$
$y_0$	is associated with	null
$y_1$	То же	$z_0$
$y_2$	- " -	$z_1$
$y_3$	- " -	$z_2$
$y_0$	$x_0$	$y_0$
$y_0$	$x_1$	$y_1$
$y_1$	$x_0$	$y_0$
$y_1$	$x_2$	$y_2$
$y_2$	$x_0$	$y_0$
$y_2$	$x_3$	$y_3$
$y_3$	null	$y_0$

```

<owl:Class rdf:about="#untitled-ontology-38:State">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#untitled-ontology-38:associated_with"/>
      <owl:someValuesFrom rdf:resource="#untitled-ontology-38:Output"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

<owl:NamedIndividual rdf:about="#untitled-ontology-38:Letter_localization_service">
  <rdfs:type rdf:resource="#untitled-ontology-38:Service"/>
  <has rdf:resource="#untitled-ontology-38:y0"/>
  <has rdf:resource="#untitled-ontology-38:y1"/>
  <has rdf:resource="#untitled-ontology-38:y2"/>
  <has rdf:resource="#untitled-ontology-38:y3"/>
</owl:NamedIndividual>
    
```

■ Рис. 9. Фрагменты построенной онтологии управляющего автомата

■ Fig. 9. Fragments of the letter localization service state machine ontology

Для последующей реализации предложенная онтология была построена в редакторе Protégé. На рис. 9 представлены фрагменты данной онтологии (описание класса State и описание экземпляров класса State).

Данная онтология используется сервисами, участвующими в сценарии формирования роботами заданного слова из букв, что обеспечивает однозначное «понимание» сервисами информационных сообщений, публикуемых в интеллектуальном пространстве.

## Заключение

В статье рассмотрено онтологическое моделирование управляющих автоматов сервисов киберфизических систем. Проанализирована актуальность использования управляющих автоматов в киберфизических системах. Предложены онтологическая модель управляющего автомата сервиса киберфизической системы на основе RDF, а также онтологическое представление состояний управляющего автомата сервиса киберфизической системы.

Использование управляющих автоматов в ряде случаев может существенно упростить процесс разработки сервисов киберфизических систем, а их представление с помощью онтологического моделирования позволит частично решить проблемы, связанные с необходимостью поддержки семантической интероперабельности сервисов.

Работа выполнена при финансовой поддержке РФФИ (проект № 15-07-08092 и 15-07-08391) и бюджетных тем № 0073-2014-0005 и 0073-2015-0007.

## Литература

1. Lee J., Bagheri B., Kao H. A. A Cyber-Physical Systems Architecture for Industry 4.0-Based Manufacturing Systems // *Manufacturing Letters*. 2015. Vol. 3. P. 18–23.
2. Knight J., Xiang J., Sullivan K. A Rigorous Definition of Cyber-Physical Systems // *Trustworthy Cyber-Physical Systems Engineering*. 2016. Vol. 47. P. 47–70.
3. Wortmann F., Flüchter K. Internet of Things // *Business & Information Systems Engineering*. 2015. Iss. 3. Vol. 57. P. 221–224.
4. Gubbi J., Buyya R., Marusic S., Palaniswami M. Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions // *Future Generation Computer Systems*. 2013. Iss. 7. Vol. 29. P. 1645–1660.
5. Görner M., Göschel T., Kassel S., Sander S., Klein T. An Ontology for Interoperability: Modeling of Composite Services in the Smart Home Environment // *International IFIP Working Conf. on Enterprise Interoperability*. 2015. P. 30–38.
6. Brizzi P., Bonino D., Musetti A., Krylovsky A., Patti E., Axling M. Towards an Ontology Driven Approach for Systems Interoperability and Energy Management in the Smart City // *Computer and Energy Science (SpliTech): Intern. Multidisciplinary Conf.* 2016. P. 1–7.
7. Kiljander J., D'Elia A., Morandi F., Hyttinen P., Takalo-Mattila J., Ylisaukko-Oja A., Soininen J.-P., Cinotti T. S. Semantic Interoperability Architecture for Pervasive Computing and Internet of Things // *IEEE Access*. 2014. Vol. 2. P. 856–873.
8. Гуренко В. В. Введение в теорию автоматов. — М.: МГТУ им. Н. Э. Баумана, 2013. — 62 с.
9. Hopcroft J. E., Motwani R., Ullman J. D. Automata Theory, Languages, and Computation. 3rd ed. — Cambridge: Pearson, 2006. — 750 p.
10. Поликарпова Н. И., Шалыто А. А. Автоматное программирование. — СПб.: Питер, 2009. — 176 с.
11. Смирнов А. В., Левашова Т. В., Шилов Н. Г. Архитектура и модели самоконтекстуализирующейся сети сервисов на примере виртуального логистического хаба // *Вестник Ростовского государственного университета путей сообщения*. 2013. № 4. С. 86–93.
12. Smirnov A., Kashevnik A., Mikhailov S., Mironov M., Petrov M. Ontology-Based Collaboration in Multi-Robot System: Approach and Case Study // *11th System of Systems Engineering Conf. (SoSE)*, Kongsberg, Norway, June 12–16, 2016. P. 329–334.
13. Dey A. K. Understanding and using Context // *Personal Ubiquitous Computing*. 2001. Vol. 5. N 1. P. 4–7.
14. Raz D., Juhola A. T., Serrat-Fernandez J., Galis A. Fast and Efficient Context-Aware Services. — John Wiley & Sons, 2006. — 222 p.
15. Qiu J. W., Lin C. P., Tseng Y. C. BLE-based Collaborative Indoor Localization with Adaptive Multi-Lateration and Mobile Encountering // *IEEE Wireless Communications and Networking Conf.*, Doha, Qatar, April 3–6, 2016. <http://ieeexplore.ieee.org/document/7564799/> (дата обращения: 20.05.2017).
16. Yoon P. K., Zihajezadeh S., Kang B. S., Park E. J. Adaptive Kalman Filter for Indoor Localization using Bluetooth Low Energy and Inertial Measurement Unit // *IEEE 37th Annual Intern. Conf. on Engineering in Medicine and Biology Society (EMBC)*, Milan, Italy, August 25–29, 2015. P. 825–828.



UDC 519.713

doi:10.15217/issn1684-8853.2017.5.80

**Ontological Modelling of State Machines for Cyberphysical System Services**Shilov N. G.<sup>a</sup>, PhD, Tech., Senior Researcher, nick@iias.spb.suSchekotov M. S.<sup>a</sup>, Researcher, shekotov@iias.spb.su<sup>a</sup>Saint-Petersburg Institute for Informatics and Automation of the RAS, 39, 14 Line, V. O., 199178, Saint-Petersburg, Russian Federation

**Introduction:** Services of cyberphysical systems are supposed to be able not only to access Internet for information search, extraction and representation, but also to make decisions in accordance with the environment state. Due to computational limitations of such devices, it is considered to be promising to use state machines for their programming. However, the necessity of their interaction causes problems related to their interoperability support. **Purpose:** Our goal is ontological modelling of state machines for cyberphysical system services. **Results:** Usage of state machines for the development of cyberphysical system services has been proposed. A RDF-compatible ontological model of a state machine for a cyberphysical system service has been developed, based on a representation of the services, their states and output symbols as classes, and their input symbols as properties. The results are demonstrated via a case study of robot interaction for assembling a given word out of letters. **Practical relevance:** The obtained results can be used in the development of cyberphysical system services.

**Keywords** — Interoperability, Ontological Modelling, State Machine, Service, Cyberphysical System.

**References**

1. Lee J., Bagheri B., Kao H. A. A Cyber-Physical Systems Architecture for Industry 4.0-Based Manufacturing Systems. *Manufacturing Letters*, 2015, vol. 3, pp. 18–23.
2. Knight J., Xiang J., Sullivan K. A Rigorous Definition of Cyber-Physical Systems. *Trustworthy Cyber-Physical Systems Engineering*, 2016, vol. 47, pp. 47–70.
3. Wortmann F., Flüchter K. Internet of Things. *Business & Information Systems Engineering*, 2015, iss. 3, vol. 57, pp. 221–224.
4. Gubbi J., Buyya R., Marusic S., Palaniswami M. Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions. *Future Generation Computer Systems*, 2013, iss. 7, vol. 29, pp. 1645–1660.
5. Görner M., Göschel T., Kassel S., Sander S., Klein T. An Ontology for Interoperability: Modeling of Composite Services in the Smart Home Environment. *Intern. IFIP Working Conf. on Enterprise Interoperability*, 2015, pp. 30–38.
6. Brizzi P., Bonino D., Musetti A., Krylovsky A., Patti E., Axling M. Towards an Ontology Driven Approach for Systems Interoperability and Energy Management in the Smart City. *International Multidisciplinary Conf. "Computer and Energy Science" (SpliTech)*, IEEE, 2016, pp. 1–7.
7. Kiljander J., D'Elia A., Morandi F., Hyttinen P., Takalo-Mattila J., Ylisaukko-Oja A., Soininen J.-P., Cinotti T. S. Semantic Interoperability Architecture for Pervasive Computing and Internet of Things. *IEEE Access*, 2014, vol. 2, pp. 856–873.
8. Gurenko V. V. *Vvedenie v teoriyu avtomatov* [Introduction into the State Machine Theory]. Moscow, MGTU im. N. E. Bauman Publ., 2013. 62 p. (In Russian).
9. Hopcroft J. E., Motwani R., Ullman J. D. *Automata Theory, Languages, and Computation*. 3rd ed. Cambridge: Pearson, 2006. 750 p.
10. Polikarpova N. I., Shalyto A. A. *Avtomatnoe programirovanie* [State Machine Programming]. Saint-Petersburg, Piter Publ., 2009. 176 p. (In Russian).
11. Smirnov A. V., Levashova T. V., Shilov N. G. Architecture and Models of Self-Contextualizing Service Network on a Case Study of Virtual Logistic Hub. *Vestnik Rostovskogo gosudarstvennogo universiteta putei soobshcheniia*, 2013, no. 4, pp. 86–93 (In Russian).
12. Smirnov A., Kashevnik A., Mikhailov S., Mironov M., Petrov M. Ontology-Based Collaboration in Multi-Robot System: Approach and Case Study. *11th System of Systems Engineering Conf. (SoSE)*, Kongsberg, Norway, June 12–16, 2016, pp. 329–334.
13. Dey A. K. Understanding and Using Context. *Personal Ubiquitous Computing*, 2001, vol. 5, no. 1, pp. 4–7.
14. Raz D., Juhola A. T., Serrat-Fernandez J., Galis A. *Fast and Efficient Context-Aware Services*. John Wiley & Sons, 2006. 222 p.
15. Qiu J. W., Lin C. P., Tseng Y. C. BLE-Based Collaborative Indoor Localization with Adaptive Multi-Lateration and Mobile Encountering. *IEEE Wireless Communications and Networking Conf.*, Doha, Qatar, April 3–6, 2016. Available at: <http://ieeexplore.ieee.org/document/7564799/> (accessed 20 May 2017).
16. Yoon P. K., Zihajezhadeh S., Kang B. S., Park E. J. Adaptive Kalman Filter for Indoor Localization using Bluetooth Low Energy and Inertial Measurement Unit. *IEEE 37th Annual Intern. Conf. on Engineering in Medicine and Biology Society (EMBC)*, Milan, Italy, August 25–29, 2015, pp. 825–828.