

# Структуры семантических метамodelей цифровых производств на базе концептуальных графов

С. В. Власенко  
СПбГЭТУ «ЛЭТИ»  
s\_v\_vlasenko@mail.ru

**Аннотация.** В статье рассматривается подход к применению теории и нотаций концептуальных графов при построении семантических метамodelей, позволяющих формировать слой интеллектуальных сервисов в программных платформах цифровых производств.

**Ключевые слова:** интеллектуальные системы; цифровые производства; «Интернет вещей»; Industry 4.0; концептуальные графы

Продолжающееся развитие широкомасштабных национальных, международных и крупнобюджетных корпоративных программ в сфере технологий Industry 4.0 и «Интернета вещей» (к числу которых относятся и российские программы различных уровней, включая национальную программу «Цифровая экономика РФ») в последние годы определяет и динамичный рост объемов разработок по аналогичным направлениям в сегменте открытого программного обеспечения. Сказанное относится, в том числе, и к платформам «Интернета вещей» (IoT) и «Промышленного Интернета вещей» (IIoT), являющимся одними из основных образующих элементов цифровых производств ([1, 2]).

Данное обстоятельство, в свою очередь, открывает дополнительные возможности для создания оригинальных решений в области IIoT путем формирования профильных надстроек для свободно распространяемых платформ. Одним из перспективных направлений развития подобных разработок представляется внедрение в среду цифровых производств интеллектуальных компонентов различного назначения, что и является предметом анализа в настоящей статье.

Следует отметить, что, несмотря на широкий интерес к тематике IoT, понятие IoT-платформы на данный момент, вероятно, нельзя считать вполне определенным. В первую очередь, имеются заметные расхождения в формальном описании функциональности платформ, которую следует считать типовой и относить к базовым свойствам систем рассматриваемого класса. Весьма затрудняют ситуацию и многочисленные структурные и функциональные классификации, используемые в различных источниках, в которых базовые системные функции зачастую включаются в вариативную часть понятия «IoT-платформа» и наоборот.

К числу наиболее часто упоминаемых типов функций, выполняемых «полноценными» IoT-платформами, можно отметить следующие:

1. Connectivity & normalization – обеспечение связи (между всеми компонентами IoT-среды) и

нормализация данных (приведение форм представления и процедур обработки данных из различных источников к единому принятому в рамках платформы базису);

2. Device management – управление устройствами, входящими в IoT-среду (с поддержкой, в том числе, функций конфигурирования, управляемого обновления встроенных программных средств и целевых приложений, работающих на вычислительных ресурсах «умных вещей», и т.п.);
3. Data Storage – реализация функций управления хранилищем данных (включая функции масштабирования, интеграции и т.п.);
4. Processing & action management – управление процессами и отдельными действиями по обработке данных (с поддержкой единых событийных моделей, функций диспетчеризации и других механизмов);
5. External interfaces – поддержка программных интерфейсов (API), обеспечивающих внешний доступ к информационным и функциональным ресурсам платформы (часто отождествляется с понятием Application Enablement – обеспечение работы приложений);
6. Security and Privacy – обеспечение информационной безопасности и конфиденциальности, реализация политик прав доступа и т.д.;
7. Visualization – визуализация данных (включая информацию о самой платформе и ее состоянии, о «вещах», о процессах обработки данных и т.д.);
8. Analytics – поддержка различных функций анализа данных, содержащихся в хранилище данных платформы;
9. Application Development – поддержка функций разработки интегрируемых с платформой приложений (иногда данные функции частично объединяют с элементами «External interfaces», называя весь набор функций термином «дополнительные инструментальные средства» – «Additional tools»).

В ряде источников принят иной подход к описанию функциональности IoT-платформ, основанный на выделении следующих типовых профилей:

- CMP (Connectivity Management Platforms) – платформы по управлению коммуникациями;
- NM (Network, Data & Subscriber Management) – платформы по управлению сетями, данными и абонентами;
- DMP (Device Management Platforms) – платформы по управлению устройствами («вещами»);
- AEP (Application Enablement Platforms) – платформы для обеспечения работы приложений;
- ADP (Application Development Platform) – платформы для разработки приложений.

При этом считается, что IoT-платформа может относиться к одному или нескольким профилям одновременно.

Дополнительно следует учитывать, что все перечисленные выше классификационные категории (включая группы типовых функций 1-9) имеют весьма слабо формализованные определения и допускают, тем самым, интерпретацию основных понятий в достаточно широких пределах.

Тем не менее, на качественном уровне для дальнейшего изложения представляется достаточным условно выделить минимальный состав функциональных компонентов IoT-платформ, который позволяет считать их платформами, применимыми в задачах IoT, и поддерживать при этом уровень универсальности, обеспечивающий относительно широкие возможности по применению в различных проблемных областях и классах систем. К указанным компонентам отнесем компоненты, поддерживающие функции 1-6 и профили CPM, NM, DMP, AEP и ADP со следующими поправками:

- поддержка работы внешних приложений (External interfaces) может сводиться к предоставлению открытых API, обеспечивающих, как минимум, доступ к содержимому хранилища данных и к слоям управления устройствами («вещами») и процессами, непосредственно реализуемыми на базе платформы;
- функции ADP могут быть ограничены вариантами предоставления SDK, утилит, фреймворков, программных библиотек и т.п., и не предполагают обязательного наличия IDE с развитыми интерактивными средствами разработки приложений, моделирования, визуализации и т.п.;
- средства поддержки функций 7 и 8 («визуализация» и «анализ данных») являются факультативными;
- конкретный состав эксплуатационных функций, технологий связи и передачи данных, а также классов и моделей оконечных устройств («вещей»), для которых поддерживаются профили CPM, NM и DMP, далее не рассматривается, т.к. полностью зависит от специфики задач, которые конечный пользователь предполагает решать с использованием платформы;

- системные модели, используемые компонентами AEP и ADP, должны быть доступны через открытые API.

Принятые выше допущения позволяют говорить о том, что предметом дальнейшего рассмотрения могут быть не только «развитые» IoT- и PoT-платформы ведущих производителей (Amazon, Google, Microsoft, IBM, Oracle, GE, PTC, SAP, Bosch, Cisco, Huawei, Samsung и др.), но и платформы «каркасного» типа, характерные, в том числе, для сегмента открытого и свободного программного обеспечения. Основная цель подобного «упрощения» в данном случае – распространение области анализа на IoT-разработки малого и среднего масштаба, реализуемые на базе открытых программных продуктов. В общем случае применение данной категории программных средств может обеспечить:

- возможность независимого эволюционного развития нового, оригинального функционального оснащения IoT-платформ, способного к гибкой интеграции с уже имеющимися программными продуктами разработчика и внешними приложениями;
- отсутствие необходимости в жесткой привязке к закрытым коммерческим решениям;
- упрощенное формирование собственных средств AEP и ADP, адаптированных под непосредственные потребности конечного пользователя;
- высокий потенциал развития оригинальных решений до уровня тиражируемых продуктов и технологий;
- независимое решение задач интеграции с требуемыми (для достижения конкретных целей эксплуатации платформы) внешними приложениями (в том числе – с применением оригинальных элементов AEP).

В настоящее время разработчикам доступно весьма значительное количество открытых IoT-платформ. Среди них к числу наиболее распространенных платформ, потенциально применимых в нише промышленного IoT, можно, вероятно, отнести такие проекты, как Blynk, Thingsboard, Cayenne, Kaa, ThingSpeak, Ubidots, Mainflux, WSo2, OpenIoT, PlatformIO, Eclipse IoT, Thinger.io и ряд других. Можно заметить, что некоторые из перечисленных выше платформ по многим определяющим системным характеристикам и уровню сервисов достаточно близки к лидерам коммерческого рынка PoT (в 2019 г. аналитики Gartner отнесли к данной категории такие продукты, как Cumulocity IoT от Software AG, ThingWorx от PTC, Predix от GE Digital, Watson IoT от IBM, Lumada от Hitachi, Connected Platform as Service от Accenture, Bezons от Atos и некоторые другие).

Использование интеллектуальных подсистем в среде PoT-платформ и комплексных решений на их основе является достаточно распространенным подходом к повышению функциональной мощи PoT-приложений на качественном уровне (более того, в укрупненном плане применение технологий искусственного интеллекта (ИИ) в

сфере Industry 4.0 исходно считается одним из приоритетных направлений развития). Примеры внедрения интеллектуальных компонентов в состав ПОТ-платформ имеются и для некоторых перечисленных ранее лидеров рынка ПОТ, причем, как среди проприетарных решений, так и среди продуктов open source. Тем не менее, анализ подобных примеров на содержательном уровне позволяет сделать ряд негативных выводов о текущем положении в области кооперации ПОТ и технологий ИИ. Рассмотрим некоторые из них.

1. В подавляющем большинстве случаев в среде комплексных ПОТ-решений интеллектуальные системные функции реализуются во внешних приложениях, которые интегрируются с базовой конфигурацией платформы через слой сервисов АЕР. Т.е., базовые конфигурации ПОТ-платформ, как правило, лишены «интеллектуальных» свойств.
2. Интеллектуальные компоненты практически не применяются в профиле ADP (т.е., «интеллектуализация» платформ ПОТ не затрагивает сферу проектирования целевых комплексных решений).
3. В средах ПОТ практически не представлены разработки, основанные на технологиях инженерии знаний и систем, основанных на знаниях (полностью доминируют подходы, ориентированные на использование нейронных сетей, хотя последние имеют целый ряд характерных ограничений и недостатков, которые могут оказаться принципиальными для конкретных ПОТ-приложений, особенно связанных с задачами реального времени, с обеспечением высокой надежности и точности решений и многих других). Как следствие – базовые конфигурации ПОТ-платформ, как правило, исходно не подготовлены к эффективному использованию потенциала априорных (в том числе «глубинных») знаний о предметной области производства, технологических объектах и процессах, о применяемых моделях, методах и средствах и т.п. Кроме того, зачастую решение целевых задач интеллектуальной обработки данных (управление, диагностика, поддержка принятия решений, прогнозирование, оптимизация и т.д.) исходно связывается с решением сопутствующих задач интеллектуального анализа данных (включая Big Data) и машинного обучения (реализуемых, в основном, на базе нейросетевых технологий), что далеко не всегда оправдано и не всегда логически вытекает из постановок целевых задач.

Даже ограничиваясь представленными выше выводами, можно предположить, что интеграция технологий систем, основанных на знаниях (Knowledge Based Systems – KBS), в среду ПОТ-решений и ПОТ-платформ является весьма перспективным направлением повышения эффективности и функциональности тиражируемых разработок в сфере цифровых производств. При этом общие цели интеграции должны охватывать не только область интеллектуализации функционирования, но и область интеллектуализации проектирования приложений.

В контексте настоящей статьи ограничимся далее рассмотрением конкретного класса функциональных задач – задач профиля ADP, которые могли бы решаться на базе KBS в среде ПОТ-платформ. К числу таких задач можно отнести, например, следующие:

- поддержка семантической структуризации единого информационного хранилища платформы и интеграции всех компонентов модельного базиса целевого решения;
- поддержка построения и обработки моделей знаний различного назначения (включая «исполнение» моделей процедурного типа);
- автоматическое формирование и/или оптимизация конфигураций аппаратных и программных средств на базе семантических паттернов;
- интеграция цифровых моделей различного профиля (в том числе – в процессах конструирования моделей по принципу «plug and play») на основе знаний;
- поддержка механизма семантических запросов к единому хранилищу данных и знаний платформы;
- интеллектуальный выбор (или формирование) сценариев получения решений функциональных задач на основании знаний о предметной области (о цифровом производстве) и внутренней организации платформы.

Одни из возможных подходов к построению платформ цифровых производств, способных решать перечисленные и подобные им задачи, рассматривался, в частности, в [3]. Основная идея данного подхода заключается в создании специальной «интеллектуальной надстройки» над ПОТ-платформой (в [3] – программный комплекс «AIS-2»), выступающей в роли «интеллектуального посредника» между платформой и ее программным окружением (и/или пользователем). В качестве базового инструмента работы с моделями знаний при этом предлагается использовать аппарат концептуальных графов (КГ) [4,5]. К числу основных системных компонентов рассматриваемого в [3] программного комплекса, реализующего функции «надстройки» (AIS-2), относятся:

1. виртуальная машина КГ (CGVM), осуществляющая управление исполнением сценариев, представленных в виде КГ, обслуживание единого модельного базиса среды цифрового производства, реализуемого в ПОТ-платформе, а также обработку интеллектуальных запросов внешних систем, включая запросы слоев АЕР и ADP;
2. интегрированная среда разработки (входящая в состав слоя ADP), реализующая функции поддержки проектирования, обработки и исполнения компонентов модельного базиса, построенных или интегрированных на базе КГ;
3. система трансляции КГ-моделей, реализующая их приведение к требуемым (в контекстах конкретных задач) видам;
4. комплекс интерфейсных модулей, обеспечивающих взаимодействие «надстройки» с

хранилищем платформы и слоями АЕР/ADP, а также с внешними приложениями.

При этом информационной основой рассматриваемого «интеллектуального посредника» являются комплексные, полностью интегрированные модели данных и знаний, реализуемые в среде единого хранилища данных и знаний ПОТ-платформы.

Детализируя предлагаемый в [3] подход, остановимся подробнее на способах организации подобных моделей, которые будем называть семантическими метамоделями (СММ) целевой системы (цифрового производства).

Компоненты СММ представляется удобным разделить на следующие уровни системной иерархии:

1. семантические модели элементарных объектов СММ («вещей», приложений, массивов данных и т.п.);
2. семантические модели комплексов и систем;
3. семантические модели процессов, сценариев работы, задач и методов их решения и т.п.;
4. метамодель предметной области.

Назначение каждого из перечисленных уровней можно обобщенно охарактеризовать следующим образом: (1) – уровень атомарных элементов (объектов); (2) – уровень структур объектов; (3) – уровень процессов в структурах объектов; (4) – уровень знаний о предметной области в целом, объединяющей понятия, относящиеся к сфере эксплуатации цифрового производства, к его организации, к способам и средствам реализации целевых функций, а также к семантическому описанию самой ПОТ-платформы.

Функции информационной интеграции, возлагаемые на СММ, предполагают согласование (в едином модельном пространстве) данных, знаний и программных средств, предназначенных для их обработки. Непосредственное связывание моделей с соответствующими физическими ресурсами может осуществляться в СММ на уровне 1 (уровне объектов) на основе определения специальных семантических свойств объектов, интерпретируемых в дальнейшем исполнительной системой «надстройки». На уровнях 2, 3 и 4 модели опосредованно связываются с данными ресурсами через модели уровня 1. Верхний уровень (уровень 4) СММ отвечает за «понятийный» аспект отображения знаний для всей совокупности моделей и их субъектов. Важными компонентами данного уровня являются общие схемы типизации элементов СММ и (как частный случай механизмов типизации) комплексы паттернов различного назначения, описывающих модели абстрактных знаний, которые могут использоваться далее в задачах семантического поиска, проектирования «по прототипу», сборки композитных приложений и моделей, структурной оптимизации и многих других.

Дополнительным видом структуризации СММ могут являться, в общем случае, библиотеки моделей (любых уровней), выполняющие функции архивов, справочников типовых модельных примитивов, аналогов и т.п.

Определяющей характеристикой реализации СММ в рамках рассматриваемого подхода к интеллектуализации цифровых производств является выбор аппарата КГ в

качестве базового средства представления и обработки моделей знаний. В контексте тематики настоящей статьи можно особо подчеркнуть ряд свойств данного аппарата, обеспечивающих его преимущество перед, например, онтологическими моделями и языками Semantic Web (OWL, RDF/RDFS) и другими аналогами применительно к рассматриваемым задачам построения и обработки СММ:

1. КГ, имеющие рекурсивную организацию и гибкий аппарат «контекстов», точно соответствуют принципам формирования структур СММ, рассмотренным выше;
2. КГ лишены недостатков «обычных» онтологических моделей в части отображения структурных и процессных схем различных видов, что особенно важно для СММ предлагаемого типа при работе с компонентами моделей на уровнях 2, 3 и 4;
3. принципиально важным является совмещение в КГ средств описания и обработки как декларативных, так и процедурных знаний, поскольку в СММ ПОТ-платформ необходима поддержка «исполнения» моделей знаний некоторых видов (включая, например, запуск приложений, связанных с соответствующими компонентами моделей процессов, систем, сценариев решения задач и т.д.);
4. аппарат КГ позволяет эффективно оперировать с агрегатами данных произвольной структуры, что упрощает процессы интеграции моделей знаний с массивами данных различного назначения, размещаемыми в среде хранилища данных платформы.

Приведенный перечень не исчерпывает положительных качеств КГ, которые могут быть эффективно использованы при реализации рассматриваемого подхода к построению интеллектуальных компонентов цифровых производств. Как представляется, данный подход в дальнейшем может также развиваться не только в части расширения спектра предоставляемых «интеллектуальных» инструментов, но и в направлении поиска оптимальных путей обеспечения высокой интегрируемости с внешними приложениями и различными открытыми ПОТ-платформами.

#### СПИСОК ЛИТЕРАТУРЫ

- [1] Pethuru Raj, Anupama C. Raman. The Internet of Things: Enabling Technologies, Platforms, and Use Cases. Auerbach Publications, Boca Raton, FL, USA, 2017. 364 p.
- [2] Fuqaha A., Guizani M., Mohammadi M., Aldehary M., Ayyash M., "Internet of Things: A Survey on Enabling Technologies, Protocols and Applications // IEEE Communications Surveys & Tutorials, Volume 17, Issue 4, 2015.
- [3] Vlasenko S., Efimenko G., Gnezdilov D., Brikova O. Approaches to Conceptual Graphs Notations Using in Digital Manufacturing Software Environments. // Proceedings. 2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (2019 ElConRus) p. 731-735. 10.1109/ElConRus.2019.8656842.
- [4] John F. Sowa, Knowledge Representation: Logical, Philosophical, and Computational Foundations, Brooks Cole Publishing Co., Pacific Grove, CA USA, 2000. 594 p.
- [5] Michel Chein, Marie-Laure Mugnier: Graph-based Knowledge Representation - Computational Foundations of Conceptual Graphs. Advanced Information and Knowledge Processing, Springer 2009, ISBN 978-1-84800-285-2, pp. 1-427.