Средства представления знаний и извлечения данных для интеллектуального анализа ситуаций

О.О. Карташов, М.А. Бутакова, А.В. Чернов, А.В. Костюков, Ю.И. Жарков

Ростовский государственный университет путей сообщения, Ростов-на-Дону

Аннотация: в данной работе выполнен анализ проблем и задач, возникающих при проектировании средств ситуационной осведомленности в результате которого установлено, что имеющиеся в данной области подходы не удовлетворяют требованиям, которые предъявляются к современным интеллектуальным средствам оперативной поддержки принятия решений по причинам отсутствия в них методов и средств, отражающих динамику информационных процессов и распределённую архитектуру обработки информации, обладающей свойствами слабой структурированности. Предложены методы оперирования и извлечения знаний из слабоструктурированной динамической информации.
Ключевые слова: Ситуационная осведомленность, слабоструктурированные данные, темпоральная информация, динамичные структуры, интеллектуальный анализ.

Проблемы и задачи интеллектуального анализа данных о ситуационной осведомленности. Разработка подхода к проектированию средств ситуационной осведомленности (СО), представляющих собой, фактически, особый класс информационной поддержки принятия оперативных решений и управления в разных областях вызывает интерес, как у отечественных [1 – 4], так и зарубежных исследователей [5 – 7]. Неформально, но достаточно информативно для понимания сама «ситуационная осведомленность» может быть определена как процесс восприятия, осознания и интерпретирования текущих сведений о ситуации вместе с оценками и прогнозированием возможных развитий и исходов в будущем. При этом естественным представляется широкое различие моделей и методов, математических подходов и инструментариев, используемых в этом процессе на разных его стадиях.

В нашей стране наибольшее влияние на формирование моделей и методов интеллектуального и ситуационного управления, которые сейчас

используются в области СО, оказали работы Д.А. Поспелова [8 – 10]. В указанных работах впервые для своего времени поднимался вопрос о невозможности полной формализации сложной, большой системы в терминах традиционного математического аппарата теории систем автоматического управления, в частности описаний в виде систем интегро-дифференциальных уравнений. Отмечалась также особенность больших систем, заключающаяся в непостоянстве структуры и функционировании самого объекта управления и возможность изменения целей функционирования и критериев оптимальности управления. Главными причинами, по которым работы Д.А. Поспелова до настоящего времени остаются актуальными, являются сформулированные в его работах принципы ситуационного подхода, распространенные впоследствии на широкий класс систем, которые в настоящее время принято называть «интеллектуальными системами». Фундаментальность данных принципов уже на протяжении нескольких десятилетий остается неизменной, поэтому следует обсудить их в данной работе.

Основополагающее значение в ситуационном подходе имеет вид описания ситуаций, который в дальнейшем должен предоставлять возможности трансформации исходного описания ситуаций к новым, либо изменяющимся в процессе функционирования условиям. Для такого описания был предложен логико-лингвистический подход, который в данное время можно, отчасти, считать весьма очевидным подходом к синтезу интеллектуальных систем, однако, обращаясь в прошлое следует указать на достаточно существенные аспекты первичного его непринятия в научных кругах данной области исследований.

Математический аппарат логико-лингвистического подхода на середину 1970-х годов невозможно отнести к детально проработанному, в сравнении с методами теории классических динамических систем, теории

автоматического управления, теории надежности технических систем. Тем не менее, проникновение ЭВМ в научную область моделирования человеческого интеллекта уже началось, и первым принципом ситуационного подхода был выдвинут принцип описания моделей ситуаций на естественном языке, то есть семиотическими методами. Следующий принцип определяет, что формирование модели поведения объекта в различных ситуациях сначала описывается человеком-специалистом в выбранной области, а затем перекладывается в ЭВМ, в которой имеются некоторые методы для автоматизации анализа ситуаций. Третий принцип предусматривает уточнение ситуационных моделей в связи с вновь поступающей информацией, причем такое уточнение может проводиться, опять-таки человеком-специалистом, либо некоторыми автоматизированными возможностями самообучения, закладываемыми в методы, реализуемые ЭВМ. И, наконец, четвертый принцип формулирует требование в модели ситуационного управления механизмов обобщения, переход от микроописаний ситуаций к макроописаниям, то есть наличие методов рассуждений о пополнении исходного набора ситуаций для дальнейшего принятия ситуационных решений. Перечисленные принципы иллюстрируются схемой ситуационного подхода из работы [10], показанной на рис. 1.

Начальным этапом работы схемы ситуационного управления является анализ текущей ситуации, который выполняется блоком Анализатора. Важной особенностью рассматриваемого подхода является наличие обширной базы предварительных описаний ситуаций, связанных с ними инцидентов и последствий. В связи с большой размерностью данных и сходностью описаний ситуаций часто возникает задача сокращения размерности признакового пространства, то есть формальная задача кластеризации или агрегирования данных.

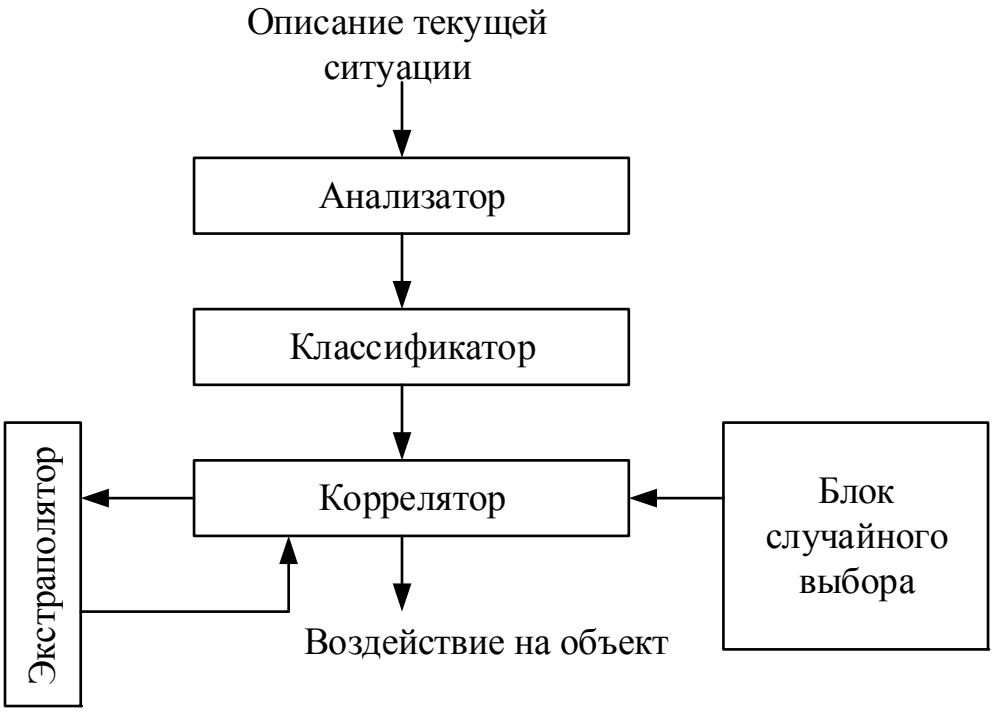

Рис. 1 – Схема ситуационного подхода

Блок классификатора принимает предварительно обработанные описания ситуаций для разделения их на классы и типы. При этом форма описания должна позволять выполнение однозначной классификации той или иной ситуации. На данном этапе зачастую применяют специализированный язык описания, позволяющий отражать кроме количественных знаний также и качественные знания о ситуациях. Таким образом, задача классификации оказывается связанной с методами теории искусственного интеллекта, позволяющими оперировать с не полностью формализуемой информацией. Та или иная классифицированная ситуация передается в блок коррелятора, хранящего правила логической трансформации ситуаций в некоторые критерии и режимы воздействия на объект. В случае, если подходящих правил логической трансформации ситуации несколько, то блок экстраполятора позволяет выбрать из них наиболее подходящее. Если подходящее или лучшее правило выбрать не получается, то правило устанавливается блоком случайного выбора.

За рубежом первые элементы теоретического фундамента в области СО начали формироваться после выхода монографий [11, 12]. В них изложен наиболее известный на данное время, показанный на рис. 2 трехуровневый процесс: 1) восприятие элементов окружающей среды; 2) понимание текущей ситуации; 3) прогноз будущего состояния. Такой процесс зачастую называют моделью Эндсли для ситуационной осведомленности.

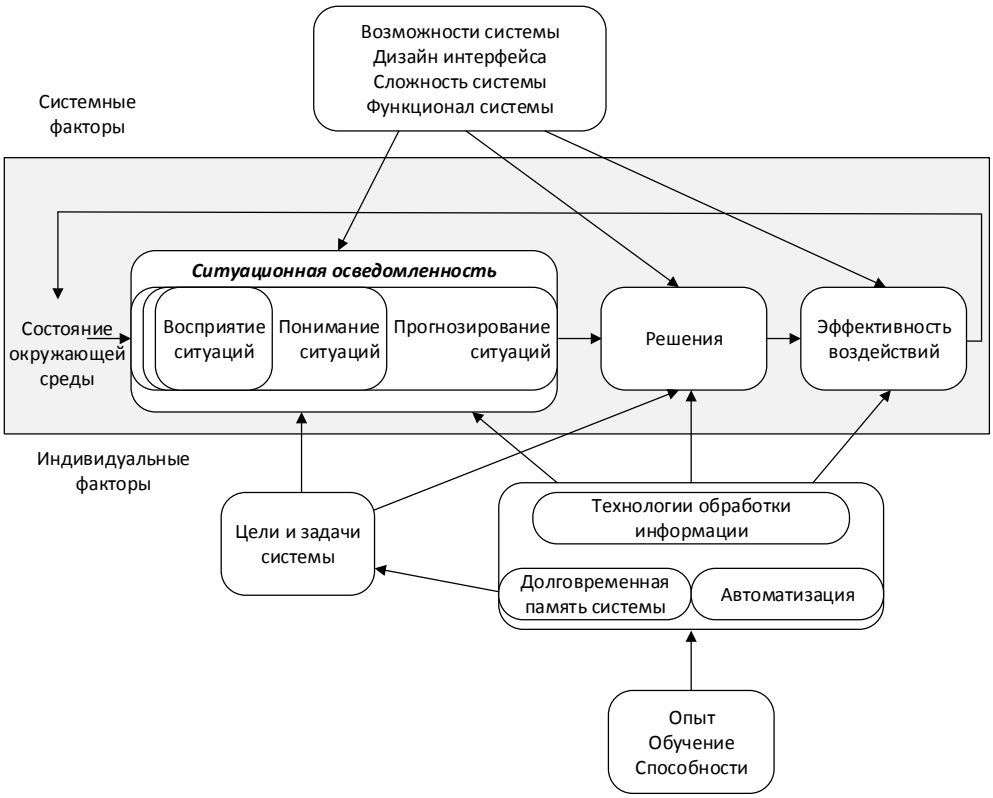

Рис. 2 – Модель Эндсли для ССО

Формальное математическое представление данной модели, как таковое отсутствует, а ситуационная осведомленность описывается в виде итеративного нелинейного процесса, подкрепляющего информированность некоторого заинтересованного лица. Уровень восприятия предназначен для определения признаков, состояний и динамических процессов, которые сопутствуют возникновению и развитию ситуаций. В техническом плане, естественным, представляется сбор информации с датчиков ввода различных

величин и измерителей, однако, Эндсли указывает на важность сопутствующих ситуации факторов, которые выделяет сам человек. Данное обстоятельство приводит нас к наличию в ситуационном подходе неполностью формализуемых и лингвистически значимых факторов. На уровне понимания текущей ситуации с технической точки зрения, реализуемой как набор некоторых распознавателей, либо алгоритмов обработки и трактования полученных данных в рассматриваемом подходе значительную роль играет опыт человека в понимании ситуаций на основе прежнего опыта, аналогий или ассоциаций. Могут быть выделены также условные веса и приоритеты для определения значимости и важности некоторой информации над другой. На третьем уровне, при установлении прогноза будущего состояния в результате ситуации, главная роль у Эндсли также отведена человеку-эксперту и уровню его уверенности в том или ином, плохом либо хорошем исходе. Данный подход к развитию ситуационной осведомленности упоминается достаточно часто, однако для современного развития информационных технологий одной лишь теоретической базы оказывается недостаточно, не говоря даже о развитии практических подходов к СО. В результате анализа упомянутых работ [11, 12]. какие-либо математические конструкции для формального синтеза систем ситуационной осведомленности обнаружить не удается, но стоит заметить, что 50 принципов проектирования СО имеют скорее интерес с организационно-управленческой стороны рассмотрения.

Чрезвычайно широкое трактование как самого термина «ситуация», так и значительное число возможных подходов к извлечению знаний о ситуациях, а также способов ситуационного принятия решений следует при переходе к практическому построению средств СО ограничивать. Безусловно, искусственно вводимые ограничения на задачи подхода к средствам СО, сами по себе также могут быть различными, однако,

дальнейшая общая классификация подходов к обработке данных в средствах СО, обладающих интеллектуальными возможностями оказывается полезной для целей уточнения и разграничения отличий предлагаемого подхода от ранее известных.

Идентификационные признаки ситуаций, которые их отличают от событий, прочих обстоятельств и факторов им сопутствующих заключаются, по крайней мере, в наличии следующих характеристик: времени возникновения, продолжительности развития, частоте проявления, последовательности и взаимодействии друг с другом, а также месте и условиям свершения ситуаций. Практически во всех случаях для ситуаций можно определить роль и участие некоторого лица (группы лиц) или объекта (объектов) в их возникновении. Также, в отличие от событий, ситуации можно обобщать, агрегировать по различным признакам, например, по схожести. Над ними можно выполнять композицию и декомпозицию, устанавливать меру их темпоральности и зависимости между собой, накладывать ограничения совместимости и взаимоисключения. Таким образом, в задачах извлечения знаний о ситуациях можно достаточно уверенно выделить задачи идентификации и классификации, спецификации и представления, рассуждений, выводов и прогнозирования. Возможные теоретические подходы к обработке данных СО с интеллектуальными возможностями с точки зрения идентификации ситуаций [13] показаны на рис. 3.

Безусловно, большинство из представленных подходов можно отнести к методам, которые используются в теории интеллектуальных систем. Необходимо отметить, что для проектирования средств СО, обладающих интеллектуальными возможностями обычно выбирается лишь один из представленных подходов, а достаточно редко выбираются несколько на разных этапах синтеза.

Существенным фактором для разработки интеллектуализации, тем более интеллектуализации СО (ИСО) является наличие функций для интерактивного взаимодействия системы с окружающей обстановкой. Такое взаимодействие естественным образом необходимо осуществлять в виде адекватной реакции, имеющей вполне рациональные намерения, а также не деструктивные последствия, как для самой ИСО, так и для окружающего мира.

Рис. 3 – Подходы к проектированию средств СО с точки зрения ухудшения качества исходных данных и усложнения описания ситуаций

Подходы, представленные на рис. 3, обладают различной сложностью использования и начальных знаний у разработчика средств ИСО. Очевидным является также факт, что и упрощение, и усложнение подхода к разработке средств ИСО должно быть разумно оправданным, и, конечно, логически обоснованным. Одним из способов такого обоснования является анализ информационных процессов в системах, для которых предполагается построение и использование ИССО. Такой системой, в рамках

диссертационной работы, является интеллектуальная система управления железнодорожным транспортом (ИСУЖТ).

В связи с этим требуется рассмотреть подробнее способы описания предметной области, имеющих слабоструктурированную и темпоральную информацию, а также наиболее распространённые способы представления знаний о такой информации.

Способы представления знаний о слабоструктурированной темпоральной информации. Слабоструктурированные документы играют важную роль в обмене данными в различных средах. С непрерывным ростом их объема, возникают вопросы, касающиеся организации и управления, а также природы источников данных. Следствием является необходимость создания автоматических процедур извлечения необходимой информации. Также имеется потребность в применении методов интеллектуального анализа данных, для извлечения и обработки огромного количества слабоструктурированных данных. Большинство таких методов не предназначены для решения поставленных в данной работе задач и требуют адаптации.

В последние годы *XML* (*eXtensible Markup Language*) получил широкое признание в качестве релевантного стандарта для представления слабоструктурированных данных. Данный формат документа имеет преимущество в виде явной структуры, которая облегчает представление и использование данных в различных контекстах. Слабоструктурированные документы получают все большее распространение в различных областях, позволяя совместно представлять текстовую информацию с помощью единой структуры.

Эта особенность *XML*-документов характерна и для других типов слабоструктурированных документов, таких как *RDF* (*Resource Description*

Framework) и *OWL* (язык веб-онтологии). *RDF* описывает семантические данные, а *OWL* является стандартом представления и обмена онтологиями.

Формирование слабоструктурированных документов – очень перспективная область для интеллектуального анализа данных, требующая новых эффективных методов извлечения знаний, структуры и содержания документов. Необходимо отметить, что при работе с данными типами документов уместно рассмотрение как структуры, так и информации о контенте.

Слабоструктурированные данные часто описывают как «с отсутствующей схемой» или «самоописывающие». Это означает, что не существует наложенной схемы или типа, и необходима система для интерпретации слабоструктурированных данных, которая обычно подчиняется некоторой графической форме. Формирование *XML* начинается с терминологического понятия, что документ «хорошо сформирован». Это очень «слабое» состояние в синтаксисе *XML*, гарантирующее лишь представление данных в виде некого древа. Данная ситуация наглядно формулирует вопрос об отсутствии структуры в заданном документе. Большая часть исследований по слабоструктурированным данным и *XML*, посвящены этой тематике.

Тема слабоструктурированных данных представляет собой ряд направлений исследований о новых способах представления и извлечения данных, которые не соответствуют традиционной модели. Данный подход широко освящен в работе Сержа Абитубула [14].

Одним из требований, предъявляемых к новой форме данных, является необходимость описания ее традиционными технологиями баз данных. Документы, рассмотренные в работах [15, 16] и форматы данных представленные в [17, 18], становятся причиной появления более выразительных языков запросов и новых методов оценки, требующих

«мягких» расширений существующих моделей данных [19]. Данные расширения требуют предварительного наложения структуры, что является затруднительным для некоторых форм данных.

Примером, может послужить система управления базами данных (СУКВ) *ACeDB* [20]. Она является объектно-ориентированной, имеющей язык во многом схожий с объектно-ориентированной СУКВ; но данная структура накладывает только частичные ограничения на данные. Также связь между данными и структурой не являются легко описываемыми в объектно-ориентированных терминах, естественно выраженные в *ACeDB*, например, произвольная глубина древа, которая не может быть запрошена с использованием обычных методов.

Следующее требование, предъявляемое к обмену данными, послужило причиной создания проекта Циммиса [21, 22] в Стэнфорде. Основанием которого является отсутствие существования всеобъемлющей модели данных, как следствие усложнение создания программного обеспечения, легко конвертирующего данные между двумя моделями.

Object Exchange Model (*OEM*) предлагает максимально гибкую структуру, которая может быть использована для большинства данных и обеспечивать субстрат, представленный практически любой другой схемой. *OEM* – это внутренняя структура для обмена между СУКВ, но наличие такой схемы требует прямого запроса данных. Объединяющей идеей является формирование графоподобной или древовидной структуры. Хотя допускается использование циклов в данных, ссылающихся на эти графы, как на деревья.

Одним из главных преимуществ неструктурированных данных является отсутствие на них ограничений. Однако имеется возможность налагать (или обнаруживать) некоторую форму структуры в данных. В [23] схема определена как граф, края которого помечены предикатами, а свойство

моделирования – используется для описания взаимосвязи между данными и схемой. В [24, 25] схема также является графовым представлением, но используется более сильное соотношение эквивалентности. В [26] структура используются для дальнейшей оптимизации. Схема полезна для просмотра и частичных ответов на запросы, что является предпосылкой перехода от слабоструктурированных к структурированным данным, которым необходимо более полное понятие схемы.

Далее следует отметить сходство на техническом уровне между слабоструктурированными базами данных [27] и мобильными вычислениями [28]. Обе области направлены на более эффективное использование, в условиях имеющихся ограничений. Технические сходства, которые возникают, в большинстве случаев являются случайными, но они должны по-прежнему наследовать некоторые методы обобщения этих областей. Более того, если есть возможность воспользоваться сходствами и обобщить их, возможно получение более широкой модели данных и вычислений.

В работах [28, 29] были описаны динамичные структуры с указанием разнообразия способов связи. Во всех этих случаях пространственная часть структуры может быть представлена в качестве дерева с надрезом.

Например, рис. 4 показывает в левом верхнем углу представление вложенного блока географической информации. В левом нижнем углу мы имеем эквивалентное представление в синтаксисе вложенных скобок исчисления окружения [29]. Когда иерархическая информация используется для представления структур документов, более подходящее графическое представление в терминах вложенных папок, как показано в правом нижнем углу. Наконец, в верхнем правом углу мы имеем более схематическое представление иерархии в терминах отмеченных краем деревьев.

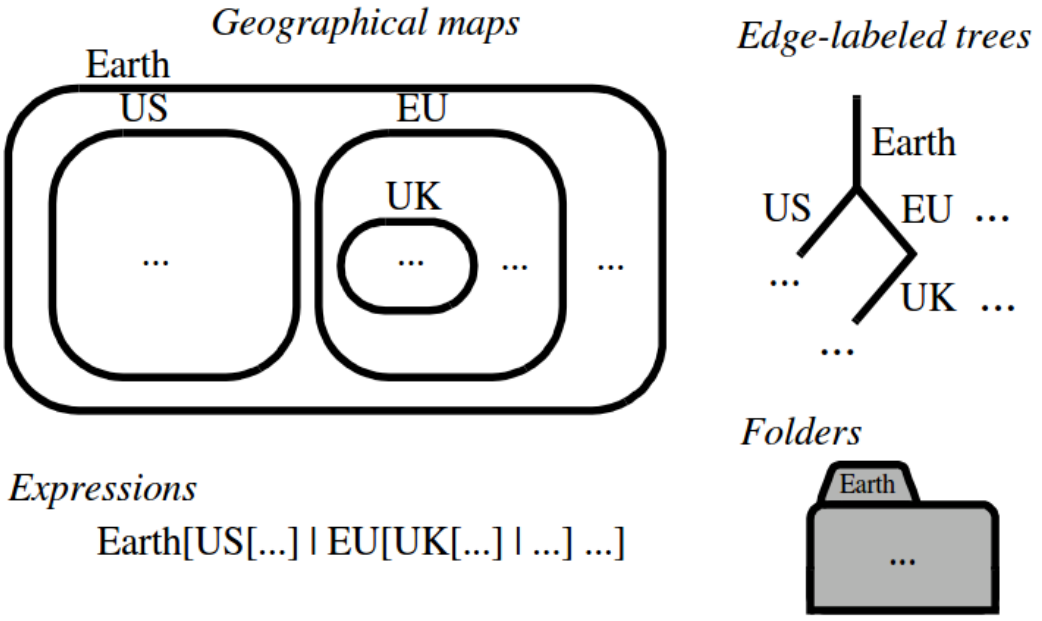

Рис. 4 – динамичные структуры с различными способами связи [29]

Определение эмбиентного исчисление позволяет сформировать общую модель мобильных вычислений. Исчерпывающее окружение до сих пор было ограничено деревьями, отмеченными краями, но возможно представление расширения (полученное путем добавления рекурсии), его можно отметить помеченными краями ориентированных графов, что является удобным предложением организации неструктурированных данных [27]. Таким образом, базовые структуры, используемые для представления слабоструктурированных данных и мобильных вычислений, согласуются. Следует подчеркнуть, что деревья и графы, помеченные краем, являются рудиментарным способом представления информации. Например, нет точного представления записи или вариантов структуры данных, которые лежат в основе почти всех современных языков программирования.

Причина этого, заключается в том, что в слабоструктурированных базах данных нельзя полагаться на фиксированное количество поддеревьев для заданного узла (следовательно, нет записей), и нельзя даже полагаться на фиксированный набор возможных структур под узлом (следовательно, нет

вариантов). Аналогично, в сети мобильных объектов нельзя полагаться на фиксированное число агентов на данном узле, или доступных ресурсов, а также ни на одно правило произвольной реконфигурации сети. Таким образом, возникают сходства в представлении ограничений на данные.

Для дальнейшего рассмотрения будет использовано представление данных об окружающем пространстве. Эта модель возникла независимо от слабоструктурированных данных; благодаря этому есть возможность получение иного результата в решении проблемы динамического представления данных. Подробное описание синтаксиса информационных выражений, отображено подмножеством исчисления окружающей среды, которое относится к структурам данных. Синтаксис представлен неупорядоченными деревьями с ограниченной глубиной (информационные деревья).

Эмбиентные исчисления предоставляют операции для описания преобразования данных. Исчисления окружения в данном описании предназначены для представления мобильных агентов, а не для манипулирования данными. Их можно представить в качестве примерного набора операций над информационными деревьями. Это обобщение на направленные графы не кажется совершенно очевидным.

Информационные деревья являются особым случаем эмпирических выражений и окружения деревьев; последнее можно представить, как динамические аспекты вычисление и изменения информация. Эмпирическое древо является информационным, где каждый узел может иметь ассоциированную совокупность параллельных потоков, которые может выполнять определенные операции. Тот факт, что потоки связаны с узлами, означает, что операции являются «локальными»: они воздействуют только на небольшое число узлов рядом с узлом потока (обычно три). Поэтому полное исчисление окружения имеет как пространственную, так и временную

составляющую. Пространственный компонент состоит из информационных деревьев, то есть слабоструктурированных данных, временной компонент включает операции, которые локально изменяют пространственный компонент. Вместо синтаксиса этих операций применяется схематичное описание.

Исходя из этого можно утверждать о наличии фундаментального сходства представления в области слабоструктурированных данных и мобильных вычислений. Более того, в случае мобильных вычислений, есть способы описания манипуляции данными (в неструктурированных базах данных, обработка данных является частью языка запросов). Вариант v представляет собой структуру вида $[l = v]$, где l - метка, а v - ассоциированная изменяемая величина и где l ограничено, как член конечного набора меток $l_1 \ldots l_n$. В случае анализа – операция может использоваться для определения, наличия меток в варианте, и дальнейшее извлечение связанных с ними значений.

Данный вариант может быть легко представлен в области слабоструктурированных данных, так как край поддерева v , с пониманием того, что l является единственным ребром его родительского узла и что l является членом конечного набора $l_1 \ldots l_n$. Но последнее ограничение не применимо в неструктурированных данных. Узел, предназначенный для представления варианта, может иметь нулевой исходящий край, или два и более краев с разными метками или даже двух ребер, чья метка не относится к предполагаемому набору. Во всех этих случаях операции стандартного анализа становятся бессмысленными.

Аналогичная ситуация возникает, в случае мобильных вычислений. Даже если ограничения вариантов конструкций соблюдаются в данный момент времени, вариант может решить оставить свой родительский узел в какой-то момент, или же другие варианты могут присоединиться к

родительскому узлу. Результатом приведения в пример нетипичной ситуации стало осознание того, что ментальные понятия типов в языках программирования становятся неприменимыми.

Для слабоструктурированных баз данных были разработаны гибкие способы запроса, несмотря на отсутствие жестких конструкций в соответствии со схемами [27]. В теории реляционных баз данных, языки запросов хорошо связаны с исчислением и логикой запросов. Была разработана логика спецификации для исчисления окружения [30]. Изначально предназначавшуюся для мобильных систем, логику можно интерпретировать (с некоторым расширением), как мощный язык запросов для слабоструктурированных данных. И наоборот, эффективные методы вычисления запросов в базах данных могут быть использованы для классов мобильных спецификаций.

В заключение можно утверждать, что неструктурированные данные и мобильные вычисления естественным образом связаны из-за скрытого сходства в проблемах, которые они пытаются решить.

Было обнаружено, что модель эмбиентного исчисления подходит для неструктурированных данных. Как следствие, система типов, разработанная для исчисления окружения, может рассматриваться, как «слабая» схема для слабоструктурированных данных. Более того, эмбиентная логика с некоторыми изменениями может рассматриваться, как язык запросов для слабоструктурированных типов данных.

Все знания, запомненные, хранящиеся на носителе, фиксирующиеся путем записи или перезаписи, механическими, физическими, химическими или электронными средствами представляют собой документ [31].

Слабоструктурированный документ является мостом между структурированными и неструктурированными данными [32]. Неструктурированные данные (также называемые плоскими данными) - это

данные, о которых не известно ни контекста, ни способа исправления информации. Структурированные данные имеют основную регулярную структуру, основанную на описательной разметке [33].

Слабоструктурированные данные возникают, когда источник не налагает жесткую структуру и когда данные объединяются из нескольких гетерогенных источников [34].

В отличие от традиционных хорошо структурированных данных, схема которых известна заранее, слабоструктурированные данные не имеют фиксированную схему, она самоописана. Они характеризуются наличием гибкой структуры, которая определяет их неоднородное содержание. Структура слабоструктурированных документов часто подразумевается, а не является жесткой или полной, как в случае традиционных баз данных [35].

Слабоструктурированные документы характеризуются тем, что они содержат сочетание коротких неграмматических (или слабо грамматических) фрагментов, а также меток [36]. *HTML* (язык разметки гипертекста), *SGML* (стандартный обобщенный язык разметки), *XML, RDF, RSS* (*Rich Site Summary*), *OWL, RDFS* (Схема описания ресурсов) и *DC* (Дублинское ядро) являются примерами слабоструктурированных документов.

Слабоструктурированные данные возникают в различных формах для широкого спектра приложений, таких как геном базы данных, научные базы данных, библиотеки программ, цифровые библиотеки, онлайн документация [35].

Слабоструктурированные документы недавно стали важной темой исследования по целому ряду причин [37]:

Во-первых, есть источники данных, которые удобно рассматривать как базы данных, но с отсутствием ограничений по схеме.

Во-вторых, необходимость иметь чрезвычайно гибкий формат для обмена между разрозненными базами данных.

В-третьих, даже при работе со структурированными данными может быть полезно рассматривать их, как слабоструктурированные (на основе само описательной схемы) для целей просмотра.

Следовательно, особое внимание привлекают автоматические методы извлечения полезной информации, в частности, обнаружение правил или шаблонов из большой коллекции слабоструктурированных документов. Разработка таковых - это адаптация методов интеллектуального анализа данных, с учетом их особенностей.

При работе со слабоструктурированными документами, согласно предварительной информации, имеющейся в коллекции, может быть уместным рассматривать только структурную информацию или рассматривать не только структуру, но и информацию о контенте. Существуют две основные и взаимодополняющие категории подходов. Рассмотрим подробное описание используемых критериев сравнения:

1. *Doc* (Полуструктурированный тип документа): популярные типы слабоструктурированных документов, такие как *HTML, XML, RDF и OWL*;
2. Технология (методы интеллектуального анализа данных): методы интеллектуального анализа данных, такие как методы кластеризации, классификации и ассоциации широко используются для слабоструктурированных документов. Для каждого подхода исследуется, как методы интеллектуального анализа данных могут быть приняты;
3. Представление. Слабоструктурированное представление документа включает в себя преобразование документа в более удобный формат;
4. Вклад: кратко описываются ключевые подходы, представляя их вклад, а также основные и предлагаемые инновации;
5. Алгоритм. Традиционные алгоритмы интеллектуального анализа данных, также новые предлагаемые варианты;

6. *TS* (Структура дерева): Слабоструктурированные документы обычно имеют иерархическую структуру. Они могут концептуально интерпретироваться, как древовидная структура, которая содержит несколько путей, связанных именованными узлами. Некоторые популярные подходы моделируют слабоструктурированный документ, как дерево, другие игнорируют древовидную структуру;

7. *NO* (Порядок узлов): Является одной из основных проблем для *XML* [27]. Проверка каждого подхода в представлении документов – узлы, элементы последовательны или независимы;

8. *ST* (семантическое представление): семантика - это изучение значения на языке [38]. В слабоструктурированных документах, семантическая обработка (лексическая, не грамматическая) имеет целью изучить семантические отношения между словами;

Следовательно, проблема заключается в определении различий множества определений, которые может иметь слово (многозначность) или разницу между словами, которые могут иметь такое же значение (синонимичность). Целью является использование семантического сходства терминов, составляющих структуру и текстовое содержание слабоструктурированных документов (теги и текст). Семантическое представление может использовать внешние семантические ресурсы, такие как онтологии, тезаурусы и таксономии. Онтологии для слабоструктурированных документов становятся серьезной проблемой в реализации семантического сбора данных.

С другой стороны, были созданы многие инструменты, системы, технологии, стандарты для реализации видения семантической сети, сети данных, состоящей из объектов (или сущностей) с фактами (или тройками), которые описывают их отношения, атрибуты в разумно структурированном виде, используя модель данных описания ресурсов (*RDF*). В частности,

особый интерес вызывают принципы связанных данных для их публикации, как часть проекта сообщества *Link Data Open Data* (*LOD*) [39]. Данные принципы значительно усиливают адаптацию, способность и удобство использования данных. Часть успеха проекта *LOD* опирается на инструменты, которые помогают издателям в публикации связанных данных из существующих структурированных источников.

В частности, класс инструментов, известных как системы *RDB2RDF* [40], трансформируют существующие реляционные источники или динамически предоставляют *RDF*. Однако, существует очень мало инструментов, которые помогают издателям в экспозиции – представлять слабоструктурированные данные, как связанные. Это в основном связано с присущим разнообразием таких преобразований, которые не могут обеспечить качественный перевод модели данных (перевод *XML*-данных на *RDF*) или вывода схемы.

Большинство слабоструктурированных форматов данных содержат неявную структуру, которую можно использовать для идентификации объектов и их типов. Помимо неявной структуры, также имеется несколько языков, специально предназначенных для описания схемы, такие как *DTD*, *XSD* или *RELAX*. Языки схемы *NG* для *XML* или *JSON*. В отсутствие таких схем описания, проблемы понимания и экстракта для неструктурированных и слабоструктурированных данных был широко изучен в литературе [41, 42].

Ограничение *XML*, а также модели иерархических данных заключаются в выражении отношений. Обнаружение всеобъемлющего набора имеет решающее значение для создания таких типов данных. Основным способом выражения отношений в *XML* и форматах на его основе является определение данных самодостаточными. Первым этапом подготовки данных является загрузка записей из внешних источников с последующими преобразование в стандартную форму документов *JSON* [43].

Обозначение объектов *JavaScript* (*JSON*) - это облегченный формат обмена данными. [43]. *JSON* - текстовый формат, полностью независимый от языков программирования. Это свойство делает *JSON* - идеальным языком обмена данными. *JSON* [44] построен на двух структурах: набора значений и упорядоченного списка. В большинство языков, набор значений реализуется как объект, структура, запись, хеш-таблица, ассоциативный массив, словарь или список клавиш. На разных языках упорядоченный список значений реализуется как массив, список, вектор, или последовательность. Основными типами *JSON* являются числа, строки, массивы, объект. Основное преимущество *JSON* заключается в том, что он переводится непосредственно в универсальные структуры данных.

Отличительная характеристика предлагаемого подхода заключается в том, что исходные источники данных остаются «изолированными» и неизменными. Ресурсы периодически загружают данные в файлы *JSON* с помощью шаблонов, связанных с онтологическими моделями. При этом они сами определяют состав, количество и актуальность для загружаемых данных. Этот тип взаимодействия является пассивным, в отличие от активного, когда клиент может использовать интерфейс *JDBC* или *ODBC* для доступа к базе данных.

Основной единицей хранения является структурированный текстовый документ, записанный в формате *JSON*, один из самых удобных для обмена данными и метаданными [45]. Преимущество *JSON*-документа - текстовый язык, независимый формат, быстрота освоения, удобная форма хранения и обмена, произвольно структурированная информация. В частности, формат *JSON*, быстрее читается и записывается, может быть проанализирован стандартной функцией *JavaScript*, а не специальным синтаксическим анализатором, как в случае *XML*.

Роль онтологий заключается в введении семантики (общая интерпретация смысла) в документы, также возможность корректировать структуру данных *JSON*-документов путем редактирования онтологии.

RDF, как абстрактная модель имеет несколько форматов сериализации. *XML* является одним из форматов для хранения и передачи данных. Быстрый просмотр *W3C RDF* [46] указывает на предпочтительный синтаксис *RDF*, которым является *RDF/XML*, но стоит отметить, что *RDF* не является строго *XML*-форматом.

Предлагаемый *RDF/JSON* представляет собой легкий текстовый синтаксис, который может быть легко модифицирован людьми, серверами и клиентами. Преимущество этого синтаксиса заключается в том, что он может легко преобразовывать другие синтаксисы, например, *RDF/XML*, используя *XSLT* [47] или *XQuery* в собственный формат. Еще одно преимущество сериализации графиков *RDF* в *JSON* заключается в том, что существуют многие программные библиотеки и встроенные функции, которые поддерживают синтаксис [48]. Еще одно преимущество *RDF/JSON* заключается в том, что этот синтаксис не имеет ограничений по *XML* и *N*-тройкам. Еще одно преимущество сериализация состоит в том, что имеется возможность управления *ECMAScript* [49]. В настоящем документе представлена возможность запроса *RDF* в качестве выходного сигнала, либо ответа *RDF* в качестве входного от служб типа *JSON*, чтобы сделать данные доступными в средах сценариев без накладных расходов других синтаксических парсеров.

Однако эти преимущества обусловлены снижением качества данных, от ошибок в содержании до ошибок в структуре. Ошибки в содержании хорошо изучены в литературе [50, 51], в то время как очень мало внимания уделялось ошибкам в структуре, при этом большинство из них фокусируется

на правильности и достоверности [52]. Существование таких ошибок может привести к неправильным результатам по запросам.

Исследование слабоструктурированных данных за последние несколько лет связано с данными, языками запросов и системами, в которых база данных делится, как некоторая форма ориентированного графа [35, 37]. Использование *EXtensible Markup Language (XML)* в качестве стандарта для представления и обмена данными привлекает значительное внимание [53]. Рассмотрим возможность переноса базы данных, *Lore* системы управления слабоструктурированными данными [54] для работы с *XML*. Следует учесть, что *XML* является всего лишь текстовым языком. Основываясь на этой модели стоит помнить об изменениях в языке запросов *Lorel*, *Lore*. Также важными будут, изменения в динамических структурных сводках Lore (*DataGuides* [55]) и взаимосвязь *DataGuides* с *XML* – определения типа документа (*DTD*).

Lore - это полная система управления базами, разработанная для обработки слабоструктурированных данных [54]. Первоначальные данные *Lore*, *OEM* (для модели обмена объектами), являются простыми, самоописательными, вложенными в объект, который можно интуитивно рассматривать как ориентированный граф [21]. В *OEM* все объекты могут быть либо атомарными, либо комплексными. Каждый объект имеет уникальный идентификатор объекта (*oid*).

Атомные объекты содержат значение от одного из атомных типов, например, *integer, real, string, gif* и т. Д. А комплексное значение *object* представляет собой набор пар, где каждая дает текстовое описание отношения между объектом и его субъектом. В графическом представлении базы данных OEM, сложные объекты имеют исходящие границы, связанные с их подчиненными объектами, а атомные объекты содержат их значение. Одна *OEM* может иметь несколько ролей, если имеет несколько входящих

краев. Язык запросов *Lore*, *Lorel*, имеет знакомый синтаксис *select-from-where* и основан на *OQL* [19], с определенными модами и расширениями, которые полезны при запросе слабоструктурированных данных. Подробная информация о *Lore* отражена в работе [56].

XML - текстовый язык, быстро набирающий популярность для представления данных и обмена [53]. Вложенные тегированные элементы являются базовыми элементами *XML*. Каждый помеченный элемент имеет последовательность нулевой или более пары атрибут/значение и последовательность из нуля или более подэлементов. Эти вспомогательные элементы могут содержать тегированные элементы, могут «не терять» сегменты текстовых данных. Поскольку *XML* был определен как текстовый язык, а не модель данных, *XML* документ всегда будет иметь неявный порядок: порядок, который может или не может быть актуальным, но тем не менее неизбежным в текстовом представлении. Сформированный *XML* документ не содержит ограничений на теги, имена атрибутов или шаблоны вложенности. В качестве альтернативы, может сопровождаться определением типа документа (*DTD*), по существу грамматикой для ограничения тегов и структуры. *XML*-документ, удовлетворяющий грамматике *DTD*, считается действительным.

В новых данных *Lore*, основанных на *XML*, элемент *XML* представляет собой пару *heid, valuei*, где *heid* - это уникальный элемент, и значение представляет собой либо текстовую строку атома, либо комплексное значение, содержащее следующие четыре компоненты:

1. Тег, привязан к строкам;

2. Упорядоченный список пар атрибут-имя/атом-значение, где каждое имя атрибута является строкой и каждое атомное значение имеет атомный тип, полученный из целого, реального, строкового и т. д., или *ID, IDREF* или *IDREFS*.

3. Упорядоченный список скрещиваемых подэлементов вида label, heidi, где *label* – строка. Поперечная связь подэлементов вводится через атрибут типа *IDREF* или *IDREFS*.

4. Упорядоченный список нормальных подэлементов вида *label, heidi*, где *label* - строка. Нормальный подэлемент вводится с помощью лексической вложенности в *XML*-файл.

После того, как *XML*-файл обрабатывается его данные удобно визуализировать, как направленные, в виде упорядоченных графов. Стоит обратить внимание, что представление графа изоморфно модели данных, поэтому они могут обсуждаться взаимозаменяемо.

Работа выполнена при финансовой поддержке РФФИ, проекты 17-07-00620а, 18-01-00402а, 18-08-00549а.

Литература

1. Карташов, О.О. Разработка информационных сервисов ситуационной осведомленности об инцидентах на основе гранулярных вычислений // Технологии разработки информационных систем (ТРИС-2017): сборник статей. − Таганрог: Издательство ЮФУ, 2017. – С. 123–128.

2. Массель, А.Г., Иванов Р.А. Ситуационный полигон как инструмент ситуационного управления в энергетике // Open Semantic Technologies for Intelligent Systems, OS-TIS 2014. – С. 277–280.

3. Рожнов, А.В. Проблематика обеспечения ситуационной осведомленности в новых задачах многопрофильных ситуационных и ситуативных центров // Материалы 21-й межд. науч. техн. конф. «Системы безопасности – 2012». – М.: Академия ГПС МЧС России, 2012. – С. 86–88.

4. Колисниченко, А.В., Федунов Б.Е. Бортовая интеллектуальная информационная система «Ситуационная осведомленность экипажа вертолета» // Мехатроника, автоматизация, управление, 2016. – Т. 17. – № 10, – С. 703–708.

5. Tretmans, J., van de Laar P., Borth M. (eds) Introduction: Situation Awareness, Systems of Systems, and Maritime Safety and Security // In: Situation Awareness with Sys-tems of Systems: Springer, 2013. – pp. 3–20.

6. Jajodia, S., Albanese M., Liu P., Jajodia S., Wang C. (eds) An Integrated Framework for Cyber Situation Awareness // Theory and Models for Cyber Situation Awareness. Lecture Notes in Computer Science: Springer, 2017. – Vol. 10030. – pp. 29–46.

7. Mozzaquatro, B.A., Jardim-Goncalves R., Agostinho C. Situation awareness in the Internet of Things // 2017 International Conference on Engineering, Technology and Innovation (ICE/ITMC), Madeira Island, Portugal, 2017. – pp. 982–990.

8. Поспелов, Д. А. Принципы ситуационного управления // Техническая кибернетика, 1971. – № 2. – С. 10–18.

9. Поспелов, Д.А. Логико-лингвистические модели в системах управления. – М.: Энергия, 1981. – 231 с.

10. Поспелов, Д.А. Ситуационное управление. Теория и практика. – М.: Наука, 1986. – 284 с.

11. Endsley, M.R., Bolte B., Jones D.G. Designing for situation awareness: An approach to human-centered design // London: Taylor & Francis, 2003, p.345.

12. Endsley, M.R., Garland D.G. (Eds.) Situation awareness analysis and measurement // Atlanta, GA: CRC Press, 2001, p.391.

13. Ye, J., Dobson, S., McKeever, S. Situation identification techniques in pervasive computing: A review // Pervasive and Mobile Computing, №.8, 2012. pp.: 36-66. doi:10.1016/j.pmcj.2011.01.004.

14. Abiteboul Serge. Querying semistructured data. In Proceedings of ICDT, Jan 1997, p.260.

15. Abiteboul Serge, Cluet Sophie, and Milo Tova. Querying and updating the file. In Proceedings of 19th International Conference on Very Large Databases, pages 73784, Dublin, Ireland, 1993.

16. Abiteboul Serge, Cluet Sophie, Christophides Vassilis, Milo Tova, and Siméon Jerome. Querying documents in object databases. In Journal of Digital Libraries, 15,volume 1:1, 1997.

17. Buneman P., Davidson SB., Hart K., Overton C., and Wong L. A data transformation system for biological data sources. In Proceedings of VLDB, Sept 1995.

18. Davidson Susan B., Overton Christian, Tannen Val, and Wong. Biokleisli Limsoon: A digital library for biomedical researchers. In Journal of Digital Libraries, volume 1:17 November 1996.

19. Cattell R. G. G., editor. The Object Database Standard: ODMC—95'. Morgan Kaufmann, San Mateo, California, 1996.

20. Thierry-Mieg Jean and Durbin Richard. ACeDB 7 A C. elegans Database: Syntactic definitions for the ACeDB data base manager, 1992.

21. Papakonstantinou Yannis, Garcia-Molina Hector, and Widom Jennifer. Object exchange across heterogenous information sources. In Proceedings of IEEE International Conference on Data Engineering, pp. 251—260, March 1995.

22. Garcia-Molina H., Papakonstantinou Y., Quass D., Rajaraman A., Sagiv Y., Ullman J., and Widom J. The tsimmis approach to mediation: Data models and languages. In Proceedings of Second INternational Workshop on Nea't Generation Information Technologies and Systems, pp. 1857193, June 1995.

23. Buneman P., Davidson S., Fernandez Mary, and Suciu D. Adding structure to unstructured data. In Proceedings of ICDT, January 1997.

24. Nestorov S., Ullman J, Weiner J, and Chawathe S. Representative objects: Concise representations of semistructured hierarchical data. In

Proceedings of the Thirteenth International Conference on Data Engineering, Birmingham, England, April 1997.

25. Goldman Roy and Widom Jennifer. Dataguides: Enabling query formulation and optimization in semistructured databases. Technical report, Stanford, 1977. p.21.

26. Fernandez Mary and Suciu Dan. Query optimizations for semi-structured data using graph schemas, 1996. URL: research. att .com/info/{mff , suciu}.

27. Abiteboul, S., Buneman, P., Suciu, D.: Data on the Web. Morgan Kaufmann Publishers, San Francisco, 2000.

28. Cardelli, L.: Abstractions for Mobile Computation. Jan Vitek and Christian Jensen, Editors. Secure Internet Programming: Security Issues for Mobile and Distributed Objects. LNCS. 1603, pp.51-94, Springer, 1999.

29. Cardelli, L., Gordon, A.D.: Mobile Ambients. FoSSaCS'98, LNCS 1378, pp.140-155, Springer, 1998.

30. Cardelli, L., Gordon, A.D.: Anytime, Anywhere. Modal Logics for Mobile Ambients. Proceedings POPL'00, pp.365-377, 2000.

31. Кондратьева Т.Н., Эксузян К.А. Анализ данных на базе технологии частного облака // Инженерный вестник Дона, 2018, №3. URL: ivdon.ru/ru/magazine/archive/n3y2018/5165.

32. Trippe, B. 2001. Do XML Editors Matter? Transform Magazine Volume 10 Issue 10, p. 27. PublisherCMP Media, Inc., USA.

33. Tannier, X. 2006. Extraction et recherche d'information en langage naturel dans les documents semi-structurés. PhD thesis, France, p.233.

34. Wang, K., and Liu H. 1997. Schema Discovery for Semistructured Data, In Proc. KDD'97, pp. 271–274.

35. Abiteboul, S. 1997. Querying semistructured data. In F. N. Afrati and P. G. Kolaitis, editors, Database Theory - ICDT ’97, 6th International Conference, Delphi, Greece, January 8-10, Proceedings, pp. 1–18.

36. Wong, T.L, and Lam W. 2004. Text Mining from Site Invariant and Dependent Features for Information Extraction Knowledge Adaptation, Proc. SIAM Int'l Conf. Data Mining (SDM), pp. 45-56.

37. Buneman, P. 1997. Semistructured data. In Proceedings of the Sixteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS’97), pp. 117-121. Tucson, Arizona.

38. Hurford, J. R. 1983. Semantics: a coursebook. Cambridge University Press, 366p.

39. Bizer C., Heath T., and Berners-Lee T. Linked Data: Principles and State of the Art. In WWW 2008.

40. Sahoo S. S., Halb W., Hellmann S., Idehen K., Thibodeau T. Jr, Auer S., Sequeda J., and Ezzat A. A Survey of Current Approaches for Mapping of Relational Databases to RDF. Technical report, W3C RDB2RDF incubator group, 2009, 15p.

41. Abiteboul S., Buneman P., and Suciu D. Data on the Web: From Relations to Semistructured Data and XML. Morgan Kaufmann, 1999, 260 p.

42. Bex G. J., Gelade W., Neven F., and Vansummeren S. Learning Deterministic Regular Expressions for the Inference of Schemas from XML Data. In WWW 2008.

43. Cowlishaw M. ECMAScript language specification. International Organization for Standardization, 1998, 191 p.

44. Crockford D. The application/json Media Type for JavaScript Object Notation (JSON). Internet Engineering Task Force, 2006.

45. Шаякбаров Н.Ф., Зорин Д.С. Анализ производительности систем управления базами данных при работе с большим объемом информации //

Инженерный вестник Дона, 2015, №2 (часть 2). URL: ivdon.ru/ru/magazine/archive/n2p2y2015/2974.

46. Beckett D. RDF/XML Syntax Specification (Revised). World Wide Web Consortium, 2004.

47. Kay M. XSL Transformations (XSLT) Version 2.0. World Wide Web Consortium, 2007.

48. Lakshman P. and Wirfs-Brock A. ECMAScript language specification 5th Edition. Ecma International, 2009.

49. Cowlishaw M. ECMAScript language specification. International Organization for Standardization, 1998.

50. Batini C. and Scannapieco M. Data Quality: Concepts, Methodologies and Techniques. Data-Centric Systems and Applications. Springer, 2006, 132p.

51. Fan W. and Geerts F. Foundations of Data Quality Management. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2012, p.217.

52. Grijzenhout S. and Marx M. The Quality of the XML Web. In CIKM, pp. 1719–1724, 2011.

53. Editors: Bray T., Paoli J, and Sperberg-McQueen C. Extensible markup language (XML) 1.0, February 1998. W3C. URL: w3.org/TR/1998/REC-xml-19980210.

54. McHugh J., Abiteboul S., Goldman R., Quass D., and Widom J. Lore: A database management system for semistructured data. SIGMOD Record, 26(3):54{66, September 1997.

55. Goldman R. and Widom J. DataGuides: Enabling query formulation and optimization in semistructured databases. In Proceedings of the Twenty-Third International Conference on Very Large Data Bases, pages 436{445, Athens, Greece, August 1997.

56. Abiteboul S., Quass D., McHugh J., Widom J., and Wiener J. The Lorel query language for semistructured data. Journal of Digital Libraries, 1(1):68{88, April 1997, 21 p.

References

1. Kartashov, O.O. Tekhnologii razrabotki informacionnyh sistem (TRIS-2017): sbornik statej. Taganrog: Izdatel'stvo YUFU, 2017. pp. 123–128.

2. Massel', A.G., Ivanov R.A. Open Semantic Technologies for Intelligent Systems, OS-TIS 2014. pp. 277–280.

3. Rozhnov, A.V. Materialy 21-j mezhd. nauch. tekhn. konf. «Sistemy bezopasnosti – 2012». M.: Akademiya GPS MCHS Rossii, 2012. pp. 86–88.

4. Kolisnichenko, A.V., Fedunov B.E. Mekhatronika, avtomatizaciya, upravlenie, 2016. V. 17. № 10, pp. 703–708.

5. Tretmans, J., van de Laar P., Borth M. (eds). In: Situation Awareness with Sys-tems of Systems: Springer, 2013. pp. 3–20.

6. Jajodia, S., Albanese M., Liu P., Jajodia S., Wang C. (eds) Theory and Models for Cyber Situation Awareness. Lecture Notes in Computer Science: Springer, 2017. Vol. 10030. pp. 29–46.

7. Mozzaquatro, B.A., Jardim-Goncalves R., Agostinho C. 2017 International Conference on Engineering, Technology and Innovation (ICE/ITMC), Madeira Island, Portugal, 2017. pp. 982–990.

8. Pospelov, D. A. Tekhnicheskaya kibernetika, 1971. № 2. pp. 10–18.

9. Pospelov, D.A. Logiko-lingvisticheskie modeli v sistemah upravleniya. [Logical and linguistic models in control systems]. M.: EHnergiya, 1981. 231 p.

10. Pospelov, D.A. Situacionnoe upravlenie. Teoriya i praktika. [Situational management. Theory and practice]. M.: Nauka, 1986. 284 p.

11. Endsley, M.R., Bolte B., Jones D.G. London: Taylor & Francis, 2003, p.345.

12. Endsley, M.R., Garland D.G. (Eds.) Atlanta, GA: CRC Press, 2001, p.391.

13. Ye, J., Dobson, S., McKeever, S. Pervasive and Mobile Computing, №.8, 2012. pp.: 36-66. doi:10.1016/j.pmcj.2011.01.004

14. Abiteboul Serge. Querying semistructured data. In Proceedings of ICDT, Jan 1997, p.260.

15. Abiteboul Serge, Cluet Sophie, and Milo Tova. Querying and updating the file. In Proceedings of 19th International Conference on Very Large Databases, pages 73784, Dublin, Ireland, 1993.

16. Abiteboul Serge, Cluet Sophie, Christophides Vassilis, Milo Tova, and Siméon Jerome. Querying documents in object databases. In Journal of Digital Libraries, 15, volume 1:1, 1997.

17. Buneman P., Davidson SB., Hart K., Overton C., and Wong L. A data transformation system for biological data sources. In Proceedings of VLDB, Sept 1995.

18. Davidson Susan B., Overton Christian, Tannen Val, and Wong. Biokleisli Limsoon: A digital library for biomedical researchers. In Journal of Digital Libraries, volume 1:17 November 1996.

19. Cattell R. G. G., editor. The Object Database Standard: ODMC 95'. Morgan Kaufmann, San Mateo, California, 1996.

20. Thierry-Mieg Jean and Durbin Richard. ACeDB 7 A C. elegans Database: Syntactic definitions for the ACeDB database manager, 1992.

21. Papakonstantinou Yannis, Garcia-Molina Hector, and Widom Jennifer. Object exchange across heterogenous information sources. In Proceedings of IEEE International Conference on Data Engineering, pp. 251—260, March 1995.

22. Garcia-Molina H., Papakonstantinou Y., Quass D., Rajaraman A., Sagiv Y., Ullman J., and Widom J. The tsimmis approach to mediation: Data

models and languages. In Proceedings of Second INternational Workshop on Nea‘t Generation Information Technologies and Systems, pp. 1857193, June 1995.

23. Buneman P., Davidson S., Fernandez Mary, and Suciu D. Adding structure to unstructured data. In Proceedings of ICDT, January 1997.

24. Nestorov S., Ullman J, Weiner J, and Chawathe S. Representative objects: Concise representations of semistructured hierarchical data. In Proceedings of the Thirteenth International Conference on Data Engineering, Birmingham, England, April 1997.

25. Goldman Roy and Widom Jennifer. Dataguides: Enabling query formulation and optimization in semistructured databases. Technical report, Stanford, 1977. p.21.

26. Fernandez Mary and Suciu Dan. Query optimizations for semi-structured data using graph schemas, 1996. URL: research. att .com/info/{mff , suciu}.

27. Abiteboul, S., Buneman, P., Suciu, D.: Data on the Web. Morgan Kaufmann Publishers, San Francisco, 2000.

28. Cardelli, L.: Abstractions for Mobile Computation. Jan Vitek and Christian Jensen, Editors. Secure Internet Programming: Security Issues for Mobile and Distributed Objects. LNCS. 1603, pp.51-94, Springer, 1999.

29. Cardelli, L., Gordon, A.D.: Mobile Ambients. FoSSaCS’98, LNCS 1378, pp.140-155, Springer, 1998.

30. Cardelli, L., Gordon, A.D.: Anytime, Anywhere. Modal Logics for Mobile Ambients. Proceedings POPL’00, pp.365-377, 2000.

31. Kondrat'eva T.N., EHksuzyan K.A. Inženernyj vestnik Dona (Rus), 2018, №3. URL: ivdon.ru/ru/magazine/archive/n3y2018/5165.

32. Trippe, B. 2001. Do XML Editors Matter? Transform Magazine Volume 10 Issue 10, p. 27. PublisherCMP Media, Inc., USA.

33. Tannier, X. 2006. Extraction et recherche d'information en langage naturel dans les documents semi-structurés. PhD thesis, France, p.233.

34. Wang, K., and Liu H. 1997. Schema Discovery for Semistructured Data, In Proc. KDD'97, pp. 271–274.

35. Abiteboul, S. 1997. Querying semistructured data. In F. N. Afrati and P. G. Kolaitis, editors, Database Theory - ICDT '97, 6th International Conference, Delphi, Greece, January 8-10, Proceedings, pp. 1–18.

36. Wong, T.L, and Lam W. 2004. Text Mining from Site Invariant and Dependent Features for Information Extraction Knowledge Adaptation, Proc. SIAM Int'l Conf. Data Mining (SDM), pp. 45-56.

37. Buneman, P. 1997. Semistructured data. In Proceedings of the Sixteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS'97), pp. 117-121. Tucson, Arizona.

38. Hurford, J. R. 1983. Semantics: a coursebook. Cambridge University Press, 366p.

39. Bizer C., Heath T., and Berners-Lee T. Linked Data: Principles and State of the Art. In WWW 2008.

40. Sahoo S. S., Halb W., Hellmann S., Idehen K., Thibodeau T. Jr, Auer S., Sequeda J., and Ezzat A. A Survey of Current Approaches for Mapping of Relational Databases to RDF. Technical report, W3C RDB2RDF incubator group, 2009, 15p.

41. Abiteboul S., Buneman P., and Suciu D. Data on the Web: From Relations to Semistructured Data and XML. Morgan Kaufmann, 1999, 260 p.

42. Bex G. J., Gelade W., Neven F., and Vansummeren S. Learning Deterministic Regular Expressions for the Inference of Schemas from XML Data. In WWW 2008.

43. Cowlishaw M. ECMAScript language specification. International Organization for Standardization, 1998, 191 p.

44. Crockford D. The application/json Media Type for JavaScript Object Notation (JSON). Internet Engineering Task Force, 2006.

45. Shayakbarov N.F., Zorin D.S. Inženernyj vestnik Dona (Rus), 2015, №2 (part 2) URL: ivdon.ru/en/magazine/archive/n2p2y2015/2974.

46. Beckett D. RDF/XML Syntax Specification (Revised). World Wide Web Consortium, 2004.

47. Kay M. XSL Transformations (XSLT) Version 2.0. World Wide Web Consortium, 2007.

48. Lakshman P. and Wirfs-Brock A. ECMAScript language specification 5th Edition. Ecma International, 2009.

49. Cowlishaw M. ECMAScript language specification. International Organization for Standardization, 1998.

50. Batini C. and Scannapieco M. Data Quality: Concepts, Methodologies and Techniques. Data-Centric Systems and Applications. Springer, 2006, 132p.

51. Fan W. and Geerts F. Foundations of Data Quality Management. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2012, p.217.

52. Grijzenhout S. and Marx M. The Quality of the XML Web. In CIKM, pp. 1719–1724, 2011.

53. Editors: Bray T., Paoli J, and Sperberg-McQueen C. Extensible markup language (XML) 1.0, February 1998. W3C. URL: w3.org/TR/1998/REC-xml-19980210.

54. McHugh J., Abiteboul S., Goldman R., Quass D., and Widom J. Lore: A database management system for semistructured data. SIGMOD Record, 26(3):54{66, September 1997.

55. Goldman R. and Widom J. DataGuides: Enabling query formulation and optimization in semistructured databases. In Proceedings of the Twenty-Third

International Conference on Very Large Data Bases, pages 436{445, Athens, Greece, August 1997.

56. Abiteboul S., Quass D., McHugh J., Widom J., and Wiener J. The Lorel query language for semistructured data. Journal of Digital Libraries, 1(1):68{88, April 1997, 21 p.