



Безопасность
в техносфере
rintd.ru

Система управления
эвакуацией людей
eesystem.ru



ISBN 978-5-4344-0523-2



9 785434 405232

Безопасность в техносфере 12

Удмуртское региональное отделение
Общероссийской общественной организации
«Российское научное общество анализа риска»

ФГБОУ ВО «Удмуртский государственный университет»

Безопасность в техносфере

12



Ижевск 2018

Удмуртское региональное отделение
Общероссийской общественной организации
«Российское научное общество анализа риска»

ФГБОУ ВО «Удмуртский государственный университет»

БЕЗОПАСНОСТЬ В ТЕХНОСФЕРЕ

Сборник статей

Выпуск 12



Ижевск
2018

УДК 614.84, 681.51, 004.031.4, 004.492

ББК 68.9я431

Б40

Научный редактор:

доктор технических наук, профессор В. М. Колодкин

Председатель организационного комитета:

руководитель Российского научного общества анализа риска,

кандидат психологических наук М. И. Фалеев

Б40 Безопасность в техносфере : сборник статей / науч. ред. В. М. Колодкин. — Ижевск : Институт компьютерных исследований, 2018. — 164 с.

ISBN 978-5-4344-0523-2

Данный сборник, в основном, соответствует материалам, представленным на XII Международной конференции «Безопасность в техносфере». Главная тема Конференции — Цифровые системы обеспечения безопасности.

Рассмотрены вопросы интеграции цифровых подсистем, таких как подсистема автоматического контроля количества людей в помещениях здания, подсистема автоматического мониторинга среды в горящем здании и т.д., в интегрированную систему автоматического формирования указаний людям путей эвакуации из горящего здания в режиме реального времени.

В статьях раскрываются принципы организации системы, вопросы создания математического, алгоритмического и программного обеспечения. Содержание статей дает системное представление о современных проблемах безопасности в техносфере и способах их решения.

УДК 614.84, 681.51, 004.031.4, 004.492

ББК 68.9я431

ISBN 978-5-4344-0523-2

© УРО ООО «Российское научное общество анализа риска», 2018

© ФГБОУ ВО «Удмуртский государственный университет», 2018

СОДЕРЖАНИЕ

I Цифровые системы обеспечения безопасности	5
<i>В.М. Колоджин, Б.В. Чирков, Д.Е. Ушаков</i>	
Повышение эффективности системы оповещения и управления эвакуацией людей при пожаре в здании . . .	6
<i>Б.В. Чирков</i>	
Методы совершенствования и алгоритмы управления эвакуацией из здания	19
<i>А.М. Сивков</i>	
Протокол последовательной передачи данных	45
<i>Д.Е. Ушаков, Б.В. Чирков</i>	
Исследование ограничений расстановки беспроводных узлов на базе микроконтроллера ATmega128RFA1	48
<i>А.М. Сивков, А.Н. Семакина</i>	
Об электрической схеме подключения сенсора инфракрасных лучей	53
<i>А.М. Сивков, Д.А. Пухова</i>	
О влиянии солнечного света на инфракрасный сенсор .	56
<i>С.В. Шархун, Н.Ф. Сирина</i>	
Результаты разработки, реализации и внедрения программного комплекса «СОУЭ-ПК» на инфраструктурных объектах ОАО «РЖД»	58
II Техносферная безопасность	67
<i>М.Э. Галиуллин</i>	
Картографическая подсистема веб-сервиса оценки риска на техногенных объектах	68
<i>А.В. Радикова, В.О. Анашин</i>	
Анализ и оценка аварийного риска с точки зрения системного анализа	83
<i>В.О. Анашин, А.В. Радикова</i>	
Ранжирование территорий по уровню коллективного риска при авариях на техногенных объектах на примере автозаправочных станций	88
<i>А.В. Романенко, Г.М. Чигвинцев, С.В. Широков, Д.В. Варламов, С.Ю. Загуменов</i>	
Проект противопожарного комплекса для повышения безопасности людей и эффективности применения средств индивидуальной защиты и средств пожаротушения	96

<i>Д.М. Варламова</i>	
Обзор существующих методов по оценке экономической эффективности систем пожарной безопасности	108
<i>И.М. Янников, В.С. Кужлин, В.И. Молчанов, А.Е. Любаков</i>	
О некоторых аспектах применения спринклерных установок пожаротушения на производстве	115
<i>А.С. Соловьева, М.В. Телегина</i>	
Поддержка принятия решений по обеспечению безопасности химически опасных объектов	121
<i>Ф.В. Недопекин, Н.С. Шестакин, А.В. Несова</i>	
Анализ потенциала поглощения диоксида углерода на перспективных участках его хранения в Донбассе	126
<i>И.М. Янников, М.В. Шабардин, М.В. Телегина</i>	
Экологическое картографирование реабилитируемых территорий	133
<i>А.Ю. Лучина</i>	
Двухскоростная модель движения газожидкостной смеси в аэротенках с пневматической системой аэрации	138
<i>И.М. Янников, И.Н. Вологжанин, Р.Г. Бадамшина</i>	
Проблематика автоматизации прогнозирования паводков и наводнений	143
<i>Д.М. Костин</i>	
Автоматизированная система персонифицированного учета нарушений требований охраны труда	148
<i>В.В. Бодряга, Ф.В. Недопекин, В.В. Белоусов</i>	
Экологическая проблема утилизации графитной спели при переливах чугуна	154

Раздел II

Техносферная безопасность

УДК 004.031.42

КАРТОГРАФИЧЕСКАЯ ПОДСИСТЕМА ВЕБ-СЕРВИСА ОЦЕНКИ РИСКА НА ТЕХНОГЕННЫХ ОБЪЕКТАХ

М.Э. Галиуллин

Удмуртский государственный университет г. Ижевск, Россия

e-mail: *m.a.galiullin@gmail.com*

В статье описываются подходы, модели данных и программное обеспечение, использованное для построения веб-сервиса risktools.ru, интерфейс которого в значительной степени основан на географических картах. Дается обзор и описываются преимущества открытой JavaScript-библиотеки Leaflet.JS, предназначенной для внедрения картографического функционала в веб-приложения. Кратко перечислены использованные в проекте программные дополнения к библиотеке Leaflet.JS, расширяющие ее возможности, а также некоторые сторонние JS-библиотеки. Описывается используемая модель данных и делается упор на ее сходство с моделями данных, применяемых в ГИС-системах. Приводится краткая инструкция для пользователей, начинающих работать с описываемым в статье веб-сервисом.

Ключевые слова: оценка риска, техногенные объекты, веб-сервис, геоинформационные системы, javascript, leaflet.js, json-schema.

Назначение сервиса

Цель создания веб-сервиса risktools.ru – обеспечить доступ к тематическим моделям прогнозирования последствий аварий, к тематическим моделям оценки и анализа рисков и соответствующим программным продуктам широкому кругу пользователей (представителям экспертных, страховых организаций, представителям собственников объектов и т.д.). Веб-сервис предоставит конечному пользователю инструментарий для прогнозирования уровня опасности объектов и территорий, который может использоваться, в том числе, и для выявления экономически обоснованных мероприятий по снижению ущерба при возможных авариях.

Примечание. На момент написания статьи рабочая бета-версия сервиса доступна по адресу <http://beta.risktools.ru>.

Общее видение сервиса

Сервис позволяет проводить расчеты риска в двух направлениях:

- 1) прогнозирование последствий аварии определенного типа (огненный шар, пожар пролива и т.п.);
- 2) прогнозирование комплексных последствий аварий на площадке техногенного объекта определенного типа. В данный момент из техногенных объектов есть реализация только для АЗС. В дальнейшем будут подключены другие объекты, такие как АГЗС, Котельная, и др.

Для первого направления расчетов пользователю необходимо задать на карте точку — источник опасности, затем выбрать тип аварии и ввести соответствующие ей начальные расчетные данные (например, тип и массу взрывчатого вещества и т.п.). После этого можно запустить расчет и увидеть его результаты в виде таблиц, а также в виде изолиний потенциального риска на карте.

Результатами комплексных расчетов по второму направлению являются:

- 1) таблицы с результатами расчетов;
- 2) план-схема размещения опасного объекта на карте;
- 3) потенциальный риск на карте;
- 4) зоны поражения человека на карте;
- 5) величина коллективного риска на объекте.

Сами комплексные расчеты риска по техногенному объекту требуют ввода всех источников опасности, присутствующих на объекте (геометрическое расположение на карте местности и начальные расчетные данные).

Таким образом, значительная часть вводимой пользователем информации является геометрическими фигурами (точками, линиями, многоугольниками), наложенными поверх карты местности. Поэтому при реализации сервиса в качестве пользовательского интерфейса ввода данных был выбран картографический интерфейс, основанный на открытой и широко используемой JavaScript-библиотеке — Leaflet.js [1]. Реализация повторяет известные стили дизайна, используемые в популярных картографических веб-сервисах [2,3], когда все пространство окна браузера целиком занято картой.

Внутреннее устройство веб-сервиса. Используемое программное обеспечение

На стороне сервера работает известная связка программного обеспечения (ПО), состоящая из Web-сервера (Apache, Nginx) + MySQL DB + PHP.

На стороне клиента (браузера) картография обеспечивается JS-библиотекой Leaflet. Для динамической подгрузки данных (передаваемых в формате JSON) используются AJAX-вызовы через интерфейс REST API [4], созданный также с помощью языка программирования PHP.

Библиотека Leaflet.JS

На рынке ПО для построения картографических веб-приложений существует несколько популярных продуктов. Самые известные из них:

- Google Maps JavaScript API [2];
- Яндекс.Карты JavaScript API [3];
- OpenLayers [5];
- Leaflet.JS [1].

Сделанный выбор в пользу последней из списка обосновывается следующими факторами:

Leaflet.JS является полностью свободной альтернативой Google Maps API и Яндекс.Карты API, не имеющей каких-либо ограничений на использование и число запросов;

Leaflet.JS очень легковесная, объем кода занимает 38 КБ;

Библиотека разработана, чтобы быть расширяемой, благодаря чему существует огромное количество плагинов, покрывающих все потребности разработчиков для создания функционального картографического веб-приложения;

Совместно с Leaflet.JS можно использовать любые тайловые сервисы карт [6].

Модель данных

В основе любого программного проекта лежит модель данных, которая проектируется так, чтобы максимально близко моделировать объекты из реального мира. Если классифицировать существующие реализации моделей данных, то можно, с некоторыми

оговорками, поделить их на реляционные (SQL, табличные), NoSQL (Not Only SQL) и Объектно-Ориентированные.

Поскольку в данном проекте мы работаем с картографией, то неизбежно сталкиваемся с понятиями, пришедшими из сферы ГИС-технологий (Гео-Информационные Системы [7]). Одним из таких ГИС-понятий является термин «слой» - группа объектов, имеющих сходную внутреннюю структуру в модели данных. Работая с моделью, поделенной на «слои», становится очень удобным хранить данные в реляционных СУБД, таких как MySQL, PostgreSQL и т.п. Для этого достаточно добавить к хранимому объекту в соответствующей таблице поле, содержащее его географическое представление на карте в виде векторных примитивов (точки, линии, полигоны). Недостатком «слоеного» подхода является необходимость жесткого деления всех объектов по принадлежности к определенному типу-слою, что не всегда является удобным, гибким и расширяемым.

Тем не менее, в данном проекте для описания модели данных для техногенных объектов было принято решение использовать реляционную модель данных с разбивкой всех объектов/элементов на несколько тематических слоев данных. Дополнительные геометрические поля (без топологии) содержат в себе данные в формате поля «geometry» из стандарта GeoJSON [8]. В отличие от стандартных требований многих распространенных ГИС-форматов хранения данных, таких как «shapefile» [9], в разрабатываемой модели тип векторной геометрии объектов может быть различным в пределах одного слоя (точки, линии, мультилинии, полигоны, мультиполигоны).

Таблицы, присутствующие в базе данных, практически повторяют набор тематических слоев для данного типа техногенного объекта. Например, для объектов типа АЗС набор таблиц оказался следующий:

dobjects — хранит техногенные объекты всех типов, в том числе и типа АЗС (без гео-поля);

azs — общая информация об АЗС (без гео-поля);

buildings — строения на территории АЗС (операторская, газон, бордюр, обвалование и т.д.);

storetanks — цистерны с топливом;

dispensers — Топливо-Раздаточные Колонки, ТРК;

crowds — места скопления людей.

Список таблиц в базе данных и их начальное наполнение фиксируется с помощью программной системы миграций данных LiquiBase [10]. Данная система позволяет хранить все изменения в структуре базы данных вместе с самим кодом проекта в репозитории кода, например в Git [11].

Все данные для техногенного объекта передаются на сторону браузера в виде JSON-структуры, содержащей в себе как общую информацию, так и данные по каждому тематическому слою. При вызове API данные по каждому слою, взятые из колонок соответствующих таблиц БД, собираются вместе в структуру формата GeoJSON. Следующий пример демонстрирует возможное содержимое формата GeoJSON:

```
{
  "type": "Feature"
  "geometry": {
    "type": "Point"
    "coordinates": [125.6, 10.1]
  },
  "properties": {
    "name": "Dinagat Islands"
  }
}
```

Для того чтобы отобразить для пользователя атрибутивную информацию для каждого объекта (не геометрическую), а также сделать возможным ее редактирование, была задействована JavaScript-библиотека «JSON-Editor» [12], позволяющая сформировать и отобразить формы ввода данных на основе декларативного описания полей в виде стандартных схем JSON-Schema [13]. Сами JSON-схемы, для техногенного объекта в целом и для каждого из слоев, подгружаются браузером на страницу через AJAX-запросы.

Библиотека Leaflet.JS. Описание и возможности библиотеки

Leaflet.JS - библиотека с открытым исходным кодом, написанная на JavaScript, предназначенная для отображения карт на веб-сайтах. Поддерживает большинство десктопных и мобильных платформ (HTML5 и CSS3).

Наряду с OpenLayers, Google Maps API, Яндекс.Карты API и еще несколькими известными программными продуктами это одна

из наиболее популярных картографических JavaScript-библиотек, использующаяся на многих известных крупных сайтах.

Leaflet позволяет разработчику отображать растровые карты, поделенные на маленькие фрагменты — тайлы [16], с дополнительными слоями (как векторными, так и растровыми), накладываемыми поверх основного. Накладываемые слои могут быть интерактивными.

Пример использования

Следующий пример задает координаты центра карты и масштаб, затем отображает на странице в указанном HTML-контейнере `<div>` базовую карту, подгружаемую с сервиса тайловых карт «mapbox.com». Поверх карты по указанным координатам накладываются векторные элементы — маркер и полигон, которые имеют всплывающую интерактивную подсказку. И в конце примера создается еще один интерактивный элемент, всплывающий и показывающий координаты места на карте, где был совершен клик мышкой.

```
<div id="mapid" style="width: 600px; height: 400px;"></div>
<script>
var mymap = L.map('mapid').setView([51.505, -0.09], 13);
L.tileLayer('https://api.tiles.mapbox.com/v4/{id}/{z}/{x}/{y}.png?
access_token=here_is_your_token_xxxxxxx', {
  maxZoom: 18,
  id: 'mapbox.streets'
}).addTo(mymap);
L.marker([51.5, -0.09]).addTo(mymap)
  .bindPopup(«b>Hello world!</b><br />I am a
popup.").openPopup();
L.polygon([
  [51.509, -0.08],
  [51.503, -0.06],
  [51.51, -0.047]
]).addTo(mymap).bindPopup("I am a polygon.");
var popup = L.popup();
function onMapClick(e) {
  popup
  .setLatLng(e.latlng)
  .setContent("You clicked the map at "+ e.latlng.toString())
```

```
.openOn(муаар);  
}  
муаар.он('click', онМаарСlick);  
</script>
```

Плагины Leaflet.JS и другие JS-библиотеки

Функциональность Leaflet.js очень легко расширяется за счет множества существующих плагинов, написанных сообществом разработчиков. Написать собственный плагин не сложно, используя очень хорошо документированное API к библиотеке и легко читаемый код самого ядра.

Далее перечислены используемые в проекте плагины Leaflet и другие JS-библиотеки.

Редактор векторной графики

Leaflet.PM [14] — плагин для создания и редактирования векторных слоев (рисунок 1). При редактировании элементов плагин реализует удобную для пользователя возможность «прилипать» новыми узлами к уже существующим. Поддерживаются основные векторные примитивы и формат GeoJSON.

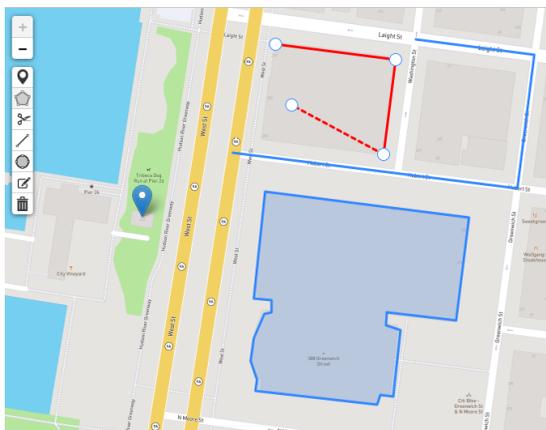


Рисунок 1 — Плагин к библиотеке Leaflet - Leaflet.PM.

Используя данный плагин можно реализовать на веб-странице возможность так называемой «оцифровки» объектов, то есть возможность очерчивать кликами мышки контуры объектов, видимых на растровой карте. Процедура «оцифровки» выполняется для создания векторных (цифровых) аналогов объектов, которые лишь визуально определяются пользователем на «спутниковой» базовой карте.

Наложение изображений на карту

Плагин ImageOverlay.Rotated [15] позволяет накладывать на карту дополнительные изображения, давая возможность пользователю поворачивать их и растягивать в нужном направлении (рисунок 2).

Для проведения процедуры «оцифровки» пользователю порой не хватает детализации и качества спутниковых снимков, лежащих в основе базовой карты. Используя данный плагин можно реализовать возможность подгружать дополнительные, имеющиеся в распоряжении пользователя, чертежи/схемы/планы в виде изображений, и подгонять их «по месту» (поворачивать, растягивать), визуально опираясь на базовую карту.

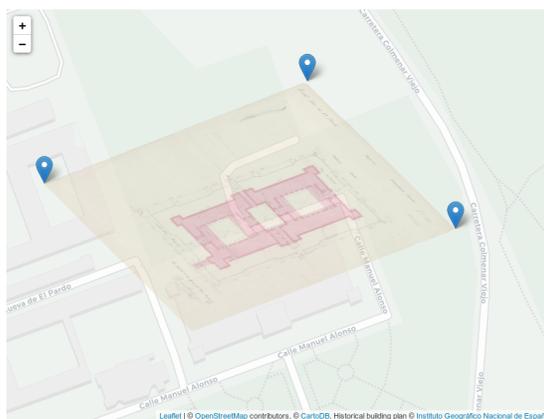


Рисунок 2 — Плагин к библиотеке Leaflet - ImageOverlay.Rotated

Геопоиск

Плагин OSMGeocoder [16] позволяет быстро перемещаться по карте в точку на земной поверхности, которая максимально близко совпадает с адресом, указанным в строке поиска. Например, пользователь может набрать «Красноярск» - в этом случае центр карты и масштаб будет выбран таким образом, чтобы целиком уместить границы города Красноярска в окне браузера. В другом примере, таком как «Красноярск, улица Ленина, 150», позиционирование и масштабирование произойдет по зданию с указанным точным адресом.

При редактировании уже существующего техногенного объекта инструмент геопоиска можно не показывать пользователю, поскольку местонахождение объекта уже было определено ранее и сохранено в базе данных.

Json Editor

JavaScript-библиотека «Json Editor» [12] позволяет сгенерировать HTML-формы на основе предоставленной JSON-схемы [13] (рисунок 3). Библиотека поддерживает стандарт Json-Schema версий 3 и 4 и может быть интегрирована в такие популярные CSS-фреймворки как Bootstrap, Foundation и jQueryUI.

Json-Schema, описывающая данные, передаваемые в формате JSON, позволяет:

- добавить к данным описание каждого поля;
- описать и валидировать ограничения на диапазон допустимых значений любого поля;
- перечислить списки возможных значений поля, а также сопоставить каждому возможному значению понятную и отображаемую для пользователя величину.

Leaflet.BeautifyMarkers и Font Awesome

Плагин BeautifyMarkers [17] позволяет использовать иконки из известной библиотеки Font Awesome [18] в качестве маркеров на карте. Это в итоге позволяет пользователю визуально отличать тип и назначение каждого отображаемого маркера.



Рисунок 3 — JavaScript-библиотека JSON Editor, позволяющая генерировать HTML-формы на основе описания в виде JSON-Schema

Leaflet.Dialog

Плаги́н Leaflet.Dialog [19] позволяет добавлять на карту всплывающие диалоговые окна, с помощью которых можно добавить необходимую функциональность карте наподобие контекстной помощи и т.п.

Библиотека JS EnjoyHint

JavaScript-библиотека EnjoyHint [20] позволяет создать интерактивные подсказки для пользователя, впервые попавшего на сайт и испытывающего трудности с тем, чтобы быстро разобраться, как там всё устроено.

Краткая инструкция по использованию редактора

Основная последовательность действий в редакторе при создании нового техногенного объекта рисунок (рисунок 4) может быть следующей:

В строке геопоиска наберите примерный адрес географического местоположения, в котором находится интересующий вас техногенный объект.

Спозиционируйте карту более точно, используя инструменты масштаба «+», «-» и колесо мыши. Также можно сдвигать карту, используя мышь с зажатым левой кнопкой.

В правом верхнем углу выберите подходящую базовую карту из списка доступных – упрощенная схема, схема, спутниковые снимки (2 варианта).

Выберите активный слой, в который начнете добавлять векторные элементы. Например, для техногенного объекта типа АЗС можно выбрать слой «Строительные сооружения».

После выбора активного слоя в правой стороне появится Панель инструментов для ввода векторных элементов. Нажмите на кнопку определенного типа векторного элемента (например «многоугольник») и начните щелчками левой кнопки мыши «оцифровывать» (обводить контур) видимого на карте объекта, такого как, например, «операторская АЗС». Для завершения ввода многоугольника следует еще раз щелкнуть на первой введенной вершине. Для завершения же ввода линии следует еще раз щелкнуть на последней введенной в линии вершине. Если вы решили отказаться от ввода элемента, то щелкните на Панели инструментов еще раз.

После завершения ввода геометрии нового элемента в правом нижнем углу появится окно ввода дополнительных атрибутов. Введите все требуемые значения полей.

Щелчок на пустом месте карты снимет активное выделение только что введенного элемента и уберет окно атрибутов.

При наведении мыши на любой из существующих элементов на карте окно атрибутов будет показывать значения полей данного элемента в режиме «только для чтения».

Если требуется отредактировать любой элемент, следует щелкнуть по нему на карте. В этом случае он станет активным, контур будет окрашен в красный цвет, а окно атрибутов в правом нижнем углу позволит изменить значения полей элемента. Также в окне атрибутов будут присутствовать кнопки редактирования геометрии элемента и его удаления.

Для редактирования геометрии элемента следует сделать его активным, и затем в окне атрибутов нажать кнопку «Править геометрию». Вершины геометрии элемента станут доступными для перемещения, добавления и удаления. После завершения всех исправлений следует еще раз нажать ту же кнопку, которая будет называться «Завершить правку».

Удалить любой элемент можно в любой момент, сделав его активным и нажав кнопку «Удалить».

В левом верхнем углу карты расположено раскрывающееся дополнительное меню. В нем вы можете сохранить свою работу, предварительно введя название для нового техногенного объекта.



Рисунок 4 — Внешний вид редактора техногенного объекта

В выпадающем меню также можно ввести дополнительные параметры, касающиеся техногенного объекта, а также подгрузить собственный план/схему/чертеж в виде изображения, путь к которому следует указать в поле «URL-адрес изображения».

После сохранения произойдет перегрузка страницы. Если вы добавили путь в поле «URL-адрес изображения», то оно отобразится поверх карты после сохранения, и вам следует подогнать его по месту, визуальнo опираясь на базовые спутниковые снимки, используя маркеры, расположенные по углам изображения.

Список литературы

1. Leaflet.JS - открытая JavaScript-библиотека для интерактивных карт [Электронный ресурс]. - <http://leafletjs.com/> (режим доступа: свободный, дата обращения 10.03.2018).
2. Google Maps APIs [Электронный ресурс]. - <https://developers.google.com/maps/> (режим доступа: свободный, дата обращения 10.03.2018).
3. API Яндекс.Карт [Электронный ресурс]. - <https://tech.yandex.ru/MAPS/> (режим доступа: свободный, дата обращения 10.03.2018).
4. REST // Википедия. [Электронный ресурс]. Дата обновления: 14.02.2018. URL: <http://ru.wikipedia.org/?oldid=90910995> (режим доступа: свободный, дата обращения: 14.02.2018).

5. OpenLayers — динамические карты на любых веб-страницах [Электронный ресурс]. - <http://openlayers.org/> (режим доступа: свободный, дата обращения 10.03.2018).
6. A tiled web map is a map displayed in a browser by seamlessly joining dozens of individually requested image files over the internet // Википедия [Электронный ресурс]. Дата обновления: 06.09.2017. URL: https://en.wikipedia.org/wiki/Tiled_web_map (режим доступа: свободный, дата обращения 10.03.2018).
7. ГИС технологии // Википедия. [Электронный ресурс]. Дата обновления: 29.09.2017. URL: <http://ru.wikipedia.org/?oldid=87932278> (режим доступа: свободный, дата обращения: 29.09.2017).
8. GeoJSON. [Электронный ресурс]. - <http://geojson.org/> (режим доступа: свободный, дата обращения 10.03.2018).
9. Shapefile file format // Википедия [Электронный ресурс]. Дата обновления: 31.05.2017. URL: <http://ru.wikipedia.org/?oldid=85712924> (режим доступа: свободный, дата обращения: 31.05.2017).
10. LiquiBase - source control for your database [Электронный ресурс]. - <http://www.liquibase.org/> (режим доступа: свободный, дата обращения 10.03.2018).
11. Git is distributed version control system [Электронный ресурс]. - <https://git-scm.com/> (режим доступа: свободный, дата обращения 10.03.2018).
12. JSON Editor takes a JSON Schema and uses it to generate an HTML form [Электронный ресурс]. - <https://github.com/jdorn/json-editor> (режим доступа: свободный, дата обращения 10.03.2018).
13. JSON Schema is a vocabulary that allows you to annotate and validate JSON documents [Электронный ресурс]. - <http://json-schema.org/> (режим доступа: свободный, дата обращения 10.03.2018).
14. Leaflet.PM - Plugin For Creating And Editing Geometry Layers in Leaflet 1.0 [Электронный ресурс] - <https://github.com/codeofsumit/leaflet.pm> (режим доступа: свободный, дата обращения 10.03.2018).
15. Leaflet.ImageOverlay.Rotated - Display rotated and skewed images in your LeafletJS maps [Электронный ресурс]. -

- <https://github.com/IvanSanchez/Leaflet.ImageOverlay.Rotated>
(режим доступа: свободный, дата обращения 10.03.2018).
16. OSM Geocoder is a simple geocoder that uses the OpenstreetMap geocoder Nominatim to locate places [Электронный ресурс]. - <https://github.com/k4r573n/leaflet-control-osm-geocoder> (режим доступа: свободный, дата обращения 10.03.2018).
 17. Leaflet.BeautifyIcon, a lightweight plugin that adds colorful iconic markers without images for Leaflet by giving full control of style to end user [Электронный ресурс]. - <https://github.com/marslan390/BeautifyMarker> (режим доступа: свободный, дата обращения 10.03.2018).
 18. Font Awesome предлагает вам масштабируемые векторные иконки [Электронный ресурс]. - <http://fontawesome.ru/> (режим доступа: свободный, дата обращения 10.03.2018).
 19. A dialog modal window that is resizable and positionable on the map [Электронный ресурс]. - <https://github.com/NBTSolutions/Leaflet.Dialog> (режим доступа: свободный, дата обращения 10.03.2018).
 20. EnjoyHint - Интерактивные подсказки для Вашего веб-сайта или приложения [Электронный ресурс]. - <https://github.com/xbsoftware/enjoyhint/> (режим доступа: свободный, дата обращения 10.03.2018)

CARTOGRAPHY SUBSYSTEM OF WEB-SERVICE RISK EVALUATION OF TECHNOGENIC OBJECTS

Galiullin Marat Enzimovich

Udmurt State University, 426034 Russia, Izhevsk, Universitetskaya, 1
e-mail: *m.a.galiullin@gmail.com*

The article describes approaches, data models and software used to build a web service risktools.ru, the interface of which is largely based on geographic maps. The article gives an overview and describes the advantages of the open JavaScript library Leaflet.JS, designed to introduce cartographic functionality into web applications. Briefly listed are the software additions to the Leaflet.JS library, extending its capabilities, as well as some third-party JS libraries. The data model used is described and emphasis is placed on its similarity to the data models used in GIS systems. A brief instruction is given for users who begin working with the web service described in the article.

Keywords: risk evaluation, technogenic objects, web-service, geoinformation systems, javascript, leaflet.js, json-schema.

Научное издание

Научный редактор
Колодкин Владимир Михайлович

БЕЗОПАСНОСТЬ В ТЕХНОСФЕРЕ

Сборник статей

Выпуск 12

Компьютерный набор и верстка
Радикова Анна Владимировна

Авторская редакция

Подписано в печать 18.06.2018. Формат $60 \times 84 \frac{1}{16}$.
Печать офсетная. Усл. печ. л. 9,53. Уч.-изд. л. 10,12.
Гарнитура Computer Modern Roman. Бумага офсетная № 1.
Тираж 100 экз. Заказ № 18-34.

АНО «Ижевский институт компьютерных исследований»
426057, г. Ижевск, ул. К. Маркса, д. 250, кв. 55
E-mail: mail@rcd.ru Тел./факс: +7 (3412) 50-02-95