

Электронная библиотека прикладных радиоэкологических моделей

Р. А. Микаилова, А. Ю. Шестериков, С. И. Спиридонов

Всероссийский научно-исследовательский институт радиологии и агроэкологии
Национального исследовательского центра «Курчатовский институт», Обнинск, Россия

Аннотация. В работе обосновывается необходимость создания электронной библиотеки прикладных радиоэкологических моделей, как инструмента прогнозирования миграции радионуклидов в окружающей среде и оценки действия радиационного фактора на население и биоту. Разработана пилотная версия библиотеки, в рамках которой реализовано основное функциональное ядро системы, включая обработку запросов пользователей, работу с базой данных и отображение информации на клиентской стороне. Финальная версия библиотеки будет “заполнена” комплексом прикладных моделей, охватывающих различные аспекты радиационной экологии. Модели могут быть использованы как учеными-исследователями, так и практиками, которые занимаются вопросами безопасности и защиты окружающей среды.

Ключевые слова: электронная библиотека, модель, моделирование, базы данных, радиоэкология, Ruby, веб-программирование.

DOI 10.14357/20718632240104

EDN GPZOV

Введение

Методологической основой радиационной экологии является системный подход, интегрирующий экспериментальные методы исследования и математическое моделирование. Применение метода моделирования дает возможность систематизировать эмпирические данные в рамках моделей, разработка которых, в свою очередь, позволяет планировать проведение экспериментальных работ. Модельные разработки представляют собой эффективные инструменты исследований в различных областях радиоэкологии – миграция радионуклидов, формирование дозовых нагрузок на население и биоту, действие радиационного фактора на природные сообщества и экосистемы. Особый интерес, в связи с современной геополитической обстановкой, приобретают прикладные модели, предназначенные для прогнозирования последствий

радиоактивного загрязнения окружающей среды в результате нештатных ситуаций и разработки стратегий применения защитных мероприятий.

За последние несколько десятков лет, особенно в период после аварии на Чернобыльской АЭС, разработано большое количество прикладных моделей, описывающих радиоэкологические процессы. Значительный вклад в “копилку” радиоэкологических моделей внесли специалисты Всероссийского НИИ радиологии и агроэкологии (ВНИРАЭ) [1]. Модели с успехом использовались для решения практических задач по оценке воздействия предприятий ядерной энергетики на окружающую среду [2, 3], прогнозированию последствий радиоактивного загрязнения природных и аграрных экосистем в результате чернобыльских выпадений [4, 5], испытаний ядерного оружия [6] и т.д. Как отмечено выше,

эти разработки могут быть востребованы и в настоящее время. Однако разрозненность источников хранения моделей затрудняет возможности их оперативного использования, в связи с чем возникла необходимость создания электронной библиотеки для хранения как самих моделей (в виде файлов), так и методик их построения. Электронная библиотека представляет собой информационную систему, позволяющую надежно сохранять и эффективно использовать разнообразные коллекции электронных документов (текстовых, графических, звуковых, видео и др.), локализованных в самой системе, а также доступных ей через телекоммуникационные сети.

Целью разработки пилотной версии электронной библиотеки является автоматизация процессов хранения и предоставления общего доступа к информации о прикладных радиоэкологических моделях, разработанных сотрудниками института. Этот инструмент позволит не только облегчить процесс получения информации о существующих моделях и использовании их в своей работе, но и исключить возможные варианты с повторной разработкой. Электронная библиотека дает возможность объединить модели в одном месте, систематизировать и каталогизировать все актуальные разработки. Создание такой библиотеки в значительной степени упростит обмен информацией между учеными и специалистами в области радиоэкологии, а также повысит эффективность их работы. Следует отметить, что пилотная версия библиотеки, предназначенная для сотрудников ВНИИРАЭ, может служить прототипом разработки аналогичного инструментария, предназначенного для более широкого круга пользователей.

1. Разработка электронной библиотеки

Процесс разработки электронной библиотеки радиоэкологических моделей включал несколько этапов:

- постановка задачи;
- проектирование и прототипирование;
- выбор инструментов и методов для разработки;

- кодирование (получение итогового результата).

Постановка задачи. На первом этапе проведен анализ существующих электронных библиотек, определены их преимущества и недостатки, на основе чего сформулированы требования к библиотеке радиоэкологических моделей. Библиотека должна представлять собой программное средство, предназначенное для хранения и предоставления доступа к информации о моделях (включая соответствующие публикации), исполняемым или другим типам файлов, необходимым для использования каждой конкретной модели в работе. Важной задачей является организация (структурирование) характеристик моделей, различающихся принципами построения и областями применения.

Информация о модели должна включать:

- данные об авторах (контакты);
 - наименование модели;
 - область применения (назначение);
 - ссылку на публикацию (файл статьи);
 - описание модели (входные/выходные переменные);
 - методы реализации модели;
 - ссылку на исполняемые файлы для скачивания;
 - название проекта.
- Функционал должен включать:
- добавление новых записей;
 - редактирование существующих записей;
 - добавление исполняемых файлов;
 - добавление публикаций;
 - регистрация пользователей для администрирования;
 - разделение прав доступа.

Поскольку библиотека, в первую очередь, должна быть удобной и понятной для пользователей, особое внимание при разработке следует уделить, кроме функциональных возможностей, созданию простого интерфейса и интуитивно понятной навигации. Сделан вывод о необходимости организации базы данных с лаконичным дизайном, без перегруженных связей.

Проектирование и прототипирование. После определения требований к функционалу библиотеки было составлено техническое задание на разработку пилотной версии электронной библиотеки радиоэкологических моделей. В

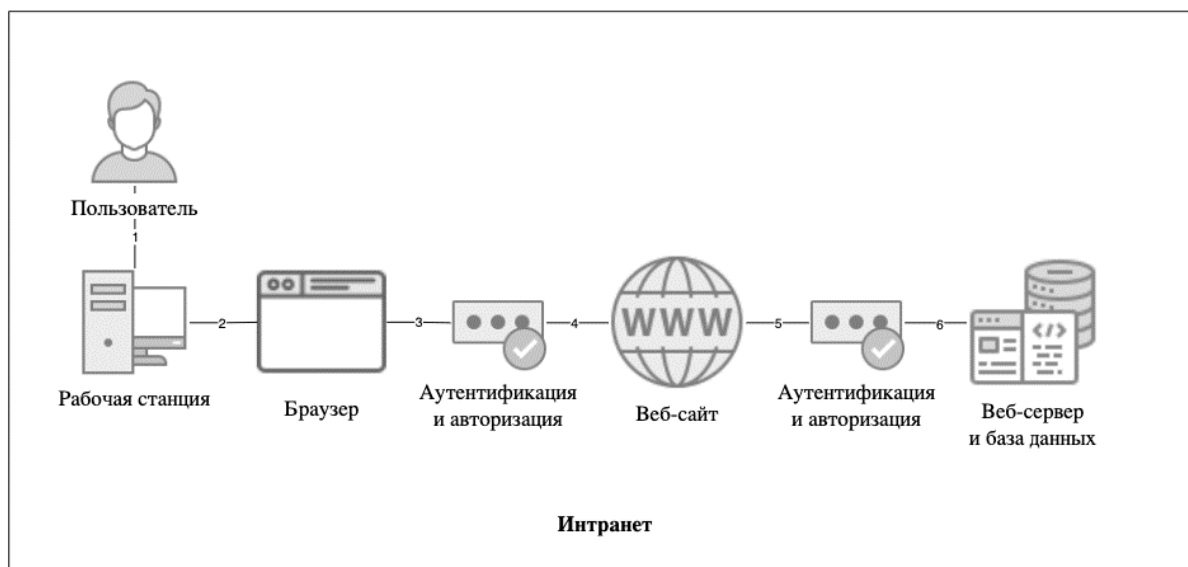


Рис. 1. Схема организации библиотеки

этом документе были описаны функциональные и нефункциональные требования, технические детали, требования к безопасности, интерфейсу и базе данных. На начальном этапе для разработки системы была выбрана монолитная архитектура, что позволило ускорить разработку и значительно упростить развертывание системы. Монолитная архитектура представляет собой единое приложение, в котором все компоненты объединены в единую кодовую базу и работают в одном процессе. Это означает, что приложение может быть развернуто на одном сервере и масштабировано вертикально путем увеличения мощности сервера.

В качестве основы электронной библиотеки использован веб-сервис. Данный подход позволит предоставлять всю необходимую информацию пользователям в привычном формате веб-сайта. В случае использования веб-сайта, нет необходимости устанавливать дополнительное программное обеспечение на рабочих станциях. В пилотной версии библиотеки доступ к ней имеют только пользователи локальной сети (интранет). Запуск пилотного проекта в интранете является хорошей стратегией, поскольку обеспечивает конфиденциальность и безопасность данных.

Уровень доступа отличается для разных типов пользователей. Так, авторизованные пользователи имеют возможность добавлять,

просматривать и редактировать записи, скачивать и просматривать файлы, связанные с моделью, а также регистрировать новых пользователей для администрирования. Неавторизованные пользователи имеют лишь возможность просматривать информацию о моделях, скачивать и просматривать файлы.

Схема взаимодействия пользователя с электронной библиотекой радиоэкологических моделей (Рис. 1):

1. Пользователь использует рабочую станцию.
2. На рабочей станции запускается веб-браузер для открытия веб-сайта электронной библиотеки радиоэкологических моделей.
3. Запрос веб-браузера к веб-сайту электронной библиотеке радиоэкологических моделей с предоставлением ответа в виде веб-страницы.
4. Определение прав пользователя для корректного отображения содержимого веб-сайта.
5. Запрос от веб-сайта к веб-серверу и базе данных для получения необходимой информации.
6. Определение прав пользователя, отправившего запрос, для его корректной обработки с предоставлением ответа.

В ходе проектирования и прототипирования была разработана структура базы данных (БД), которая предназначена для хранения всей информации библиотеки. Для построения БД была выбрана реляционная модель данных, которая

Табл. 1. Перечень сущностей

Наименование	Тип таблицы	Описание
models	основная	Таблица, в которой хранятся все данные, связанные с моделями
users	основная	Таблица, в которой хранятся все данные о пользователях

Табл. 2. Поля таблицы «models»

Имя поля	Тип данных	Описание
ID	integer	Уникальный идентификатор записи
datetime	datetime	Дата создания записи
author	string	Имя автора модели
title	string	Название модели
body	string	Область применения
description	string	Описание модели
method	string	Методы решения
project	string	Проект
file	blob	Файлы модели
user_id	index	Внешний ключ для связи с таблицей users

Табл. 3. Поля таблицы «users»

Имя поля	Тип данных	Описание
ID	integer	Уникальный идентификатор записи
datetime	datetime	Дата создания записи
name	string	Имя пользователя
email	string	Е-mail пользователя
password	string	Пароль учетной записи в виде хэша

позволяет хранить данные с логическими отношениями, где объектами базы данных являются таблицы для хранения информации, запросы, формы и скрипты. Для БД было выделено две сущности: “models” и “users” (Табл. 1).

Таблица “models” (Табл. 2) содержит информацию о моделях, такую как ID (уникальный ключ записи, автоматически заполняемое поле), дату создания записи (автоматически заполняемое поле), название, автор, описание, методы

решения, проект, а также файлы модели и внешний ключ для связи с таблицей “users”. Таблица “users” (Табл. 3) содержит информацию о пользователях, такую как (уникальный ключ записи, автоматически заполняемое поле), дату создания записи (автоматически заполняемое поле), имя, email и пароль (хэш в виде строки).

Структура БД библиотеки отображена на Рис. 2.

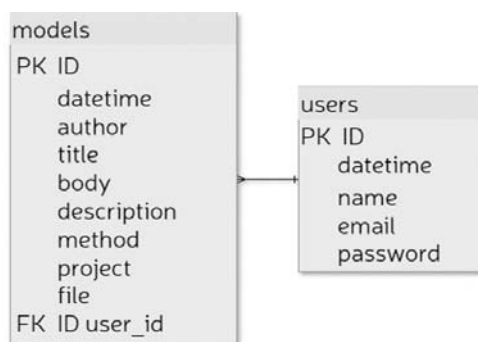


Рис. 2. Структура БД библиотеки

1.1. Выбор инструментов и методов для разработки

Основным языком для разработки был выбран Ruby, который является динамическим, объектно-ориентированным языком программирования. Этот язык был создан Юкиhiro Мацумото (Мац) и выпущен в Японии в 1995 г., после чего быстро набрал популярность благодаря своей простоте и элегантности. [7]. Ruby широко используется для разработки веб-приложений с использованием фреймворка Ruby on Rails [8], который предоставляет удобные инструменты и структуру для создания веб-приложений.

Для простоты разработки веб-сервиса был выбран фреймворк Ruby on Rails (RoR, Rails). Rails предоставляет удобную и эффективную среду для разработки веб-приложений, следуя принципам конвенции перед конфигурацией (Convention over Configuration) и модели-представление-контроллер (Model-View-Controller, MVC). Он предлагает множество готовых решений, таких как маршрутизация, обработка запросов, валидация данных, работа с базами данных и управление сессиями. Ruby on Rails поддерживает расширения (Gems [9]) для быстрой, простой и надежной разработки. Ruby on Rails является открытым программным обеспечением и распространяется под лицензией MIT.

Gems в Ruby on Rails - это модули или библиотеки, которые расширяют функциональность фреймворка. Gems представляют собой пакеты кода, которые могут быть установлены и использованы в Rails-проекте. Они содержат готовые к использованию компоненты, такие как драйверы баз данных, инструменты для аутентификации, пагинации, работу с изображениями и многое другое. Gems упрощают разработку, поскольку позволяют внедрить готовые решения, необходимые для различных задач, и ускоряют процесс разработки, так как не требуют написания кода с нуля. При разработке системы электронной библиотеки радиозэкологических моделей использовались следующие Gems:

- bcrypt [10] - расширение для хеширования паролей пользователей в приложении и обеспечения безопасности данных;
- activerecord-import [11] - расширение, которое позволяет создавать эффективные SQL-

запросы для создания, обновления или удаления множества записей в базе данных;

- saxlsx [12] - библиотека для создания файлов в формате XLSX с помощью Ruby;
- saxlsx_rails [13] - расширение, которое интегрирует saxlsx с Rails и обеспечивает возможность создания файлов в формате XLSX прямо из приложения;
- draper [14] - расширение, которое облегчает создание представлений и шаблонов для приложений на основе Rails;
- pagy [15] - расширение, которое обеспечивает простую пагинацию для больших списков данных в приложении на основе Rails;
- rubyXL [16] - библиотека для чтения, записи и манипулирования файлами в формате XLSX с помощью Ruby;
- rubyzip [17] - библиотека для создания, чтения и обновления архивов Zip с помощью Ruby;
- valid_email2 [18] - расширение, которое предоставляет проверку корректности формата электронных адресов в приложении на основе Rails;
- bootsnap [19] - расширение, которое ускоряет процесс загрузки приложения на основе Rails путем кэширования и оптимизации загрузки кода;
- faker [20] - расширение, которое позволяет генерировать реалистичные тестовые данные в приложении на основе Rails (использовалось в процессе прототипирования);
- pry-rails [21] - расширение, которое интегрирует pry - мощный отладочный инструмент для Ruby on Rails, облегчая отладку кода;
- rubocop [22] - инструмент статического анализа кода, который помогает улучшить качество и читаемость кода в приложении на основе Rails;
- rubocop-performance [22] - расширение для RuboCop, которое помогает оптимизировать производительность кода в приложении на основе Rails;
- rubocop-rails [22] - расширение для RuboCop, которое обеспечивает проверку соответствия стандартам кодирования в приложении на основе Rails.

Для создания клиентской части было решено применить Bootstrap [23]. Bootstrap - свободный

набор инструментов для создания сайтов и веб-приложений. Включает в себя HTML- и CSS-шаблоны оформления для типографики, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая JavaScript-расширения. Использование Bootstrap позволяет существенно ускорить процесс разработки визуальной составляющей веб-сайтов и, в то же время, придерживаться единого стиля, что положительно сказывается на восприятии со стороны пользователя.

В качестве БД предпочтение было отдано SQLite [24]. SQLite является реляционной БД и представляет собой легковесную встраиваемую базу данных, не требующую дополнительных настроек или установки сервера. Благодаря этому, ее установка и использование на различных платформах становится проще и быстрее. SQLite также имеет множество других преимуществ, таких как хорошую производительность и надежность, небольшой размер базы данных, поддержку транзакций, индексирование, поддержку SQL и многое другое. Она хорошо подходит для проектов с небольшим количеством пользователей, где нет необходимости в масштабируемости или сложной структуре БД.

Для удобного кодирования был выбран редактор кода Visual Studio Code (VS Code) [25], разработанный Microsoft для Windows, Linux и macOS. Данный редактор кода позиционируется, как «легкий» редактор кода для кроссплатформенной разработки веб- и облачных приложений. Включает в себя отладчик, инструменты для работы с Git, подсветку синтаксиса. Распространяется бесплатно, разрабатывается как программное обеспечение с открытым исходным кодом.

Для обеспечения работоспособности серверной части была выбрана операционная система Linux Ubuntu Server [26], поскольку это бесплатная и открытая операционная система, что позволяет сэкономить деньги на лицензировании. Также Linux Ubuntu Server имеет широкую поддержку сообщества и огромное количество пакетов и инструментов, что позволит обеспечить гибкость и масштабируемость проекта. Помимо прочего, Linux Ubuntu Server также известна своей надежностью, безопасностью и стабильностью, что является важным фактором для больших проектов с ограниченными ресурсами.

1.2. Кодирование (получение итогового результата)

После проведения подготовительных мероприятий выполнен этап кодирования, в процессе которого описывалась логика работы библиотеки со стороны сервера, а также визуализация выводимой сервером информации в браузере на стороне пользователя веб-сайта.

Поскольку текущая версия библиотеки является пилотной, то основная задача заключалась в том, чтобы опробовать созданный инструмент в работе и собрать отзывы пользователей об ошибках, предложениях по изменению и добавлению функционала, которые можно учесть в будущих обновлениях. Так, в процессе опроса пользователей была добавлена функциональность запоминания учетных данных пользователей при входе в систему. В результате, при следующем посещении библиотеки пользователю не нужно будет вводить свои данные повторно.

2. Результаты

В результате выполнения комплекса исследований, включающих анализ существующих инструментов такого рода, составление технического задания, разработку схемы организации системы и организацию взаимодействия с ней пользователя, создана пилотная версия электронной библиотеки прикладных радиоэкологических моделей. Особое внимание было уделено формированию структуры предоставляемой информации, чтобы конечный пользователь мог работать с библиотекой с наибольшей эффективностью.

Были использованы такие технологии, как Ruby, Ruby on Rails, Bootstrap и SQLite. Использование Ruby on Rails позволило значительно сократить время на разработку библиотеки и упростить ее сопровождение. Ruby является языком высокого уровня с динамической типизацией, что облегчает написание кода. Bootstrap использовался для верстки пользовательского интерфейса, а SQLite позволила построить простое, но надежное хранение данных.

В то же время, в процессе работы были выявлены некоторые вопросы, требующие своего решения при создании следующих версий (обновлений) библиотеки. Одной из главных проблем

является отсутствие быстрого поиска с учетом различных текстовых структур поискового запроса. Для реализации поиска в библиотеке прикладных радиологических моделей на Ruby on Rails возможно использовать следующие расширения и утилиты:

- Elasticsearch [27] - мощный инструмент для полнотекстового поиска. Он может быть легко интегрирован в Rails-приложение через Gem `elasticsearch-model` [28] и `elasticsearch-rails` [28]. Elasticsearch поддерживает расширенный поиск по многим параметрам, таким как фразовый поиск, поиск по префиксу и суффиксу, сортировку, фильтры и т.д.

- Solr [29] - поисковый движок с открытым исходным кодом, который может интегрироваться с Rails через Gem `sunspot` [30]. Solr поддерживает множество функций поиска, включая фразовый поиск, поиск по синонимам, поиск по префиксу и суффиксу, сортировку, фильтры и т.д.

- Ransack [31] - расширение для поиска и сортировки в Rails, позволяющее быстро создавать сложные запросы, поддерживающее фразовый поиск, поиск по префиксу и суффиксу, а также полнотекстовый поиск.

- Thinking Sphinx [32] - расширение для полнотекстового поиска, которое интегрируется с Sphinx Search Engine [33]. Thinking Sphinx поддерживает фразовый поиск, поиск по префиксу и суффиксу, сортировку, фильтры и т.д.

Кроме того, в процессе эксплуатации библиотеки может возникнуть необходимость хранить файлы моделей отдельно, так как они будут занимать существенный объем хранилища. В этом случае использование SQLite не позволит гибко масштабировать и управлять хранилищем этих данных. Однако этот недостаток устраняется путем использования корзины данных (Simple Storage Service). Корзина данных работает по принципу хранения и доступа к файлам и объектам через веб-интерфейс или Application Programming Interface (API) и предоставляет хранилище для файлов и объектов различных типов, таких как изображения, видео, документы и другие данные. Каждый объект в корзине данных имеет уникальный ключ, по которому он идентифицируется и доступен для запросов. Этот ключ можно использовать для получения, обновления или удаления объекта.

Корзина данных обеспечивает масштабируемость хранения, позволяя добавлять и удалять объекты по мере необходимости, что позволяет поддерживать большие объемы данных и обеспечивает гибкость при расширении проекта. Реализовать хранение большого количества файлов с использованием корзины данных возможно при помощи сторонних решений, таких как:

- MinIO [34], которое представляет собой высокопроизводительное и облачное хранилище с открытым исходным кодом;

- распределенное хранилище Ceph [35] с открытым исходным кодом;

- OpenStack Swift [36] - облачное хранилище с открытым исходным кодом, которое может быть развернуто на локальном сервере.

Перечисленные решения для хранения данных являются свободным программным обеспечением с открытым исходным кодом и предоставляют доступ по API. Благодаря чему реализовать их добавление в существующий проект достаточно просто.

Следует отметить, что использование библиотеки может представлять сложность для пользователей, не имеющих опыта работы с подобными системами. В связи с этим требуются временные затраты, необходимые для разработки документации и проведения обучения пользователей.

Заключение

В результате разработки пилотной версии электронной библиотеки радиэкологических моделей реализовано основное функциональное ядро системы, включая обработку запросов пользователей, работу с базой данных и отображение информации на клиентской стороне. Основное функциональное ядро является надежным фундаментом для дальнейшего развития и продвижения системы. Одним из основных преимуществ электронной библиотеки является ее доступность для широкого круга пользователей. Модели могут быть использованы как исследователями, так и практиками, которые занимаются вопросами радиэкологии и защиты окружающей среды. Библиотека также предоставляет возможность обмена знаниями и опытом между подразделениями института, а при

запуске в открытый доступ и между различными организациями и научными сообществами.

Финальная версия электронной библиотеки прикладных радиоэкологических моделей будет содержать широкий спектр прикладных моделей, охватывающих различные аспекты радиационной экологии. К ним относятся модели для прогнозирования миграции радионуклидов в природных и аграрных экосистемах, оценки доз облучения населения и объектов окружающей среды, а также радиоэкологических рисков. Следует подчеркнуть, что электронная библиотека представляет собой эффективный инструмент для оперативного решения прикладных задач, связанных с оценкой различных радиоэкологических ситуаций. Пилотная версия библиотеки, описанная в настоящей статье, является основой для дальнейшего развития и улучшения системы.

Литература

- Фесенко С.В., Спиридонов С.И. Применение математических методов в радиоэкологии // История науки и техники. 2020. № 7. С. 115-125.
- Фесенко С.В., Спиридонов С.И., Алексахин Р.М. Оценка последствий воздействия аварийных выбросов АЭС на лесные насаждения // Атомная энергия. 1993. Т. 75. Вып. 4. С. 312-319.
- Спиридонов С.И., Микаилова Р.А. Ранжирование реакторных установок на основе оценки потенциального радиационного воздействия на природную среду // Радиационная биология. Радиоэкология. 2020. Т. 60. № 5. С. 541-549.
- Фесенко С.В., Спиридонов С.И., Санжарова Н.И., Алексахин Р.М. Моделирование биологической доступности ^{137}Cs в почвах, подвергшихся загрязнению после аварии на Чернобыльской АЭС // Радиационная биология. Радиоэкология. 1996. Т. 36. Вып. 4. С. 479-487.
- Спиридонов С.И., Фесенко С.В., Санжарова Н.И. Моделирование поведения ^{137}Cs в системе почва - растения после применения мелиорантов. // Радиационная биология. Радиоэкология. 2001, Т. 41, № 3, с. 337-334
- Спиридонов С.И., Гонтаренко И.А., Мукушева М.К., Фесенко С.В., Семиошкина Н.А. Прогнозирование накопления ^{137}Cs в продукции животноводства на территории Семипалатинского испытательного полигона // Радиационная биология. Радиоэкология. 2005. Т. 45. № 4. С. 480-487.
- D. Flanagan, Yu. Matsumoto. The Ruby Programming Language (First ed.). O'Reilly Media, 2008, p. 446.
- Ruby on Rails Guides: official site. URL: <https://guides.rubyonrails.org/> (дата обращения: 12.05.2023).
- RubyGems: official site. URL: <https://guides.rubygems.org/> (дата обращения: 12.05.2023).
- GitHub: bcrypt-ruby. URL: <https://github.com/bcrypt-ruby/bcrypt-ruby> (дата обращения: 12.05.2023).
- GitHub: ActiveRecord-Import. URL: <https://github.com/zdennis/activerecord-import> (дата обращения: 12.05.2023).
- GitHub: Caxlsx (Community Continued Version). URL: <https://github.com/caxlsx/caxlsx> (дата обращения: 12.05.2023).
- GitHub: Axlax-Rails — Spreadsheet templates for Rails. URL: https://github.com/caxlsx/caxlsx_rails (дата обращения: 12.05.2023).
- GitHub: Draper: View Models for Rails. URL: <https://github.com/drapergem/draper> (дата обращения: 12.05.2023).
- GitHub: Pagy. URL: <https://github.com/ddnexus/pagy> (дата обращения: 12.05.2023).
- GitHub: rubyXL. URL: <https://github.com/weshatheleopard/rubyXL> (дата обращения: 12.05.2023).
- GitHub: rubyzip. URL: <https://github.com/rubyzip/rubyzip> (дата обращения: 12.05.2023).
- GitHub: ValidEmail2. URL: https://github.com/micke/valid_email2 (дата обращения: 12.05.2023).
- GitHub: Bootstrap Ruby Gem. URL: <https://github.com/twbs/bootstrap-rubygem> (дата обращения: 12.05.2023).
- GitHub: Faker. URL: <https://github.com/faker-ruby/faker> (дата обращения: 12.05.2023).
- GitHub: pry-rails URL: <https://github.com/pry/pry-rails> (дата обращения: 12.05.2023).
- RuboCop: official site URL: <https://rubocop.org/> (дата обращения: 12.05.2023).
- Bootstrap: official site URL: <https://getbootstrap.com/> (дата обращения: 12.05.2023).
- SQLite: official site URL: <https://www.sqlite.org/> (дата обращения: 12.05.2023).
- Visual Studio Code: official site URL: <https://code.visualstudio.com/> (дата обращения: 12.05.2023).
- Ubuntu Server: official site URL: <https://ubuntu.com/server/docs> (дата обращения: 12.05.2023).
- Elastic: official site URL: <https://www.elastic.co/> (дата обращения: 12.05.2023).
- GitHub: Elasticsearch Rails URL: <https://github.com/elastic/elasticsearch-rails> (дата обращения: 12.05.2023).
- Solr: official site URL: <https://solr.apache.org/> (дата обращения: 12.05.2023).
- GitHub: Sunspot URL: <https://github.com/sunspot/sunspot> (дата обращения: 12.05.2023).
- GitHub: Ransack URL: <https://github.com/activerecord-hackery/ransack> (дата обращения: 12.05.2023).
- GitHub: Thinking Sphinx URL: <https://github.com/pat/thinking-sphinx> (дата обращения: 12.05.2023).
- GitHub: Sphinx URL: <https://github.com/kpumuk/sphinx> (дата обращения: 12.05.2023).
- MinIO: official site URL: <https://min.io/> (дата обращения: 12.05.2023).
- Ceph: official site URL: <https://ceph.io/> (дата обращения: 12.05.2023).
- OpenStack: official site URL: <https://wiki.openstack.org/wiki/Swift> (дата обращения: 12.05.2023).

Микаилова Рена Александровна. Всероссийский научно-исследовательский институт радиологии и агроэкологии, Обнинск, Россия. Научный сотрудник. Область научных интересов: моделирование радиоэкологических процессов, информационные технологии. E-mail: mik_r_a@ira.ru

Шестериков Александр Юрьевич. Всероссийский научно-исследовательский институт радиологии и агроэкологии, Обнинск, Россия. Младший научный сотрудник. Область научных интересов: информационные технологии. E-mail: radombota@gmail.com

Спиридонов Сергей Иннокентьевич. Всероссийский научно-исследовательский институт радиологии и агроэкологии, Обнинск, Россия. Главный научный сотрудник, доктор биологических наук, профессор. Область научных интересов: радиационная экология, радиобиология. E-mail: spiridonov.si@gmail.com

Electronic Library of Applied Radioecological Models

R. A. Mikailova, A. Yu. Shesterikov, S. I. Spiridonov, S. V. Fesenko

Russian Institute of Radiology and Agroecology of National Research Centre «Kurchatov Institute, Obninsk, Russia

Abstract. With the development of technology and the increase in the volume of information, storage and access to it have become increasingly complex tasks. One of these tasks is the storage and provision of access to application software that is used in various fields of science and technology. In this regard, it became necessary to create an electronic library that would contain applied radioecological models and other software products used in this area. The electronic library of applied radioecological models is a tool for predicting and assessing the impact of radioactive substances on the population and biota. We have successfully developed a pilot version of the library, which includes all the essential features of the system. This version encompasses processing user requests, database management, and effectively displaying information on the client side. The final version of the library will be “filled” with a set of applied models covering various aspects of radiation ecology. The models can be used by both researchers and practitioners who deal with issues of safety and environmental protection.

Keywords: electronic library, model, modeling, databases, radioecology, Ruby, web programming

DOI 10.14357/20718632240104

EDN GPZOVF

References

1. Fesenko S.V., Spiridonov S.I. Mathematical modeling in solving problems of radiation ecology. *Istoriya nauki i tekhniki*. 2020; 7: 115-125 (In Russ). doi: 10.25791/intstg.07.2020.1202
2. Fesenko S.V., Spiridonov S.I., Aleksakhin R.M. Assessment of the effect of discharges from the Novovoronezh atomic power plant on forest stands. *At Energy*. 1993;75(4): 819–824. doi: 10.1007/BF00739365
3. Spiridonov S.I., Mikailova R.A. Ranking of reactor facilities based on the assessment of potential radiation impact on the environment. *Radiatsionnaya Biologiya Radioekologiya*. 2020; 60(5): 541-549 (In Russ). doi: 10.31857/S0869803120050082
4. Fesenko S.V., Spiridonov S.I., Sanzharova N.I., Alexakhin R.M. Modeling the bioavailability of ^{137}Cs in soils contaminated after the accident at the Chernobyl NPP. *Radiatsionnaya Biologiya Radioekologiya*. 1996; 36(4): 479-487 (In Russ).
5. Spiridonov S.I., Fesenko S.V., Sanzharova N.I. Modeling of ^{137}Cs behavior in the soil-plants system following the application of ameliorants. *Radiatsionnaya Biologiya Radioekologiya*. 2001; 41(3): 337-334 (In Russ).
6. Spiridonov S.I., Gontarenko I.A., Mukusheva M.K., Fesenko S.V., Semioshkina N.A. Prediction of ^{137}Cs accumulation in animal products within the Semipalatinsk Test Site. *Radiatsionnaya Biologiya Radioekologiya*. 2005; 45(4): 480-487 (In Russ).
7. D. Flanagan, Yu. Matsumoto. *The Ruby Programming Language* (First ed.). O'Reilly Media, 2008, p. 446.
8. Ruby on Rails Guides: official site. URL: <https://guides.rubyonrails.org/> [Accessed 12 May 2023].
9. RubyGems: official site. URL: <https://guides.rubygems.org/> [Accessed 12 May 2023].
10. GitHub: bcrypt-ruby. URL: <https://github.com/bcrypt-ruby/bcrypt-ruby> [Accessed 12 May 2023].
11. GitHub: ActiveRecord-Import. URL: <https://github.com/zdennis/activerecord-import> [Accessed 12 May 2023].

12. GitHub: Caxlsx (Community Continued Version). URL: <https://github.com/caxlsx/caxlsx> [Accessed 12 May 2023].
13. GitHub: Axlxs-Rails — Spreadsheet templates for Rails. URL: https://github.com/caxlsx/caxlsx_rails [Accessed 12 May 2023].
14. GitHub: Draper: View Models for Rails. URL: <https://github.com/drapergem/draper> [Accessed 12 May 2023].
15. GitHub: Pagy. URL: <https://github.com/ddnexus/pagy> [Accessed 12 May 2023].
16. GitHub: rubyXL. URL: <https://github.com/weshatheleopard/rubyXL> [Accessed 12 May 2023].
17. GitHub: rubyzip. URL: <https://github.com/rubyzip/rubyzip> [Accessed 12 May 2023].
18. GitHub: ValidEmail2. URL: https://github.com/micke/valid_email2 [Accessed 12 May 2023].
19. GitHub: Bootstrap Ruby Gem. URL: <https://github.com/twbs/bootstrap-rubygem> [Accessed 12 May 2023].
20. GitHub: Faker. URL: <https://github.com/faker-ruby/faker> [Accessed 12 May 2023].
21. GitHub: pry-rails. URL: <https://github.com/pry/pry-rails> [Accessed 12 May 2023].
22. RuboCop: official site URL: <https://rubocop.org/> [Accessed 12 May 2023].
23. Bootstrap: official site URL: <https://getbootstrap.com/> [Accessed 12 May 2023].
24. SQLite: official site URL: <https://www.sqlite.org/> [Accessed 12 May 2023].
25. Visual Studio Code: official site URL: <https://code.visualstudio.com/> [Accessed 12 May 2023].
26. Ubuntu Server: official site URL: <https://ubuntu.com/server/docs> [Accessed 12 May 2023].
27. Elastic: official site URL: <https://www.elastic.co/> [Accessed 12 May 2023].
28. GitHub: Elasticsearch Rails URL: <https://github.com/elastic/elasticsearch-rails> [Accessed 12 May 2023].
29. Solr: official site URL: <https://solr.apache.org/> [Accessed 12 May 2023].
30. GitHub: Sunspot URL: <https://github.com/sunspot/sunspot> [Accessed 12 May 2023].
31. GitHub: Ransack URL: <https://github.com/activerecord-hackery/ransack> [Accessed 12 May 2023].
32. GitHub: Thinking Sphinx URL: <https://github.com/pat/thinking-sphinx> [Accessed 12 May 2023].
33. GitHub: Sphinx URL: <https://github.com/kpumuk/sphinx> [Accessed 12 May 2023].
34. MinIO: official site URL: <https://min.io/> [Accessed 12 May 2023].
35. Ceph: official site URL: <https://ceph.io/> [Accessed 12 May 2023].
36. OpenStack: official site URL: <https://wiki.openstack.org/wiki/Swift> [Accessed 12 May 2023].

Mikailova Rena A. Researcher, Russian Institute of Radiology and Agroecology of National Research Centre «Kurchatov Institute», Kievskoe shosse 109 km, bld. 1/1, Obninsk, 249032, Russia. E-mail: mik_r_a@rirae.ru

Shesterikov Aleksandr Yu. Junior researcher, Russian Institute of Radiology and Agroecology of National Research Centre «Kurchatov Institute», Kievskoe shosse 109 km, bld. 1/1, Obninsk, 249032, Russia. E-mail: radombota@gmail.com

Spiridonov Sergei I. Chief researcher, Doctor of biological sciences, professor, Russian Institute of Radiology and Agroecology of National Research Centre «Kurchatov Institute», Kievskoe shosse 109 km, bld. 1/1, Obninsk, 249032, Russia. E-mail: spiridonov.si@gmail.com