

УДК 004.652; 004.9  
DOI 10.25205/1818-7900-2018-16-2-86-94

**С. Н. Трошков**

*Новосибирский государственный университет  
ул. Пирогова, 1, Новосибирск, 630090, Россия*

*kamronis@yandex.ru*

## **ОБ ОПЫТЕ МИГРАЦИИ ПРИЛОЖЕНИЙ НА СВОБОДНО РАСПРОСТРАНЯЕМОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ С ОТКРЫТЫМ КОДОМ**

Важной проблемой современного программирования является поддержка и сопровождение наследственного программного обеспечения (ПО). Функциональность приложений, написанных в старых окружениях, ценна и по-прежнему актуальна. Устаревшее ПО не позволяет использовать их на современных машинах и развивать в дальнейшем. В работе описан опыт миграции на примере двух приложений – Архива академика А. П. Ершова и системы «Библиотека», которые используются в ИСИ СО РАН не один десяток лет. В качестве платформы с открытым кодом для создания новых приложений был выбран CMF Drupal, который значительно облегчает разработку и перенос модели данных. Миграция включает в себя реинжиниринг приложения с сохранением бизнес-логики, модели данных, а также перенос самих данных.

*Ключевые слова:* миграция, база данных, модель данных, реинжиниринг, drupal, ms sql server, fox pro, mysql, postgresql.

### **Введение**

Многие научные организации столкнулись с необходимостью поддержки большого количества информационных систем, разработанных в течение последних двух десятилетий с использованием проприетарного программного обеспечения. Такие системы, как правило, создавались с помощью грантов или с использованием спонсорской помощи компаний, предоставляющих программное обеспечение (ПО) окружения – серверные платформы, СУБД, системы программирования, интернет-серверы. Гранты закончились, и ресурсов на обновление ПО окружения больше нет, а приложения используются по сей день и работают на устаревшем оборудовании, в устаревшем окружении, в то время как своей актуальности они не потеряли и требуют дальнейшего развития и сопровождения.

В связи с этим становится актуальной задача миграции с проприетарного ПО на свободное. В данной статье будет описан метод такой миграции на опыте двух проектов – Архива академика А. П. Ершова и библиотечной системы «Библиотека».

### **Что такое миграция приложений и для чего она нужна?**

Определим, что мы будем понимать под миграцией приложений. «Миграция (от англ. *migration*) приложений – процедура перевода программных продуктов (исходного кода и структуры базы данных) с одной платформы (технологии) на другую (чаще всего из уста-

*Трошков С. Н. Об опыте миграции приложений на свободно распространяемое программное обеспечение с открытым кодом // Вестн. НГУ. Серия: Информационные технологии. 2018. Т. 16, № 2. С. 86–94.*

ревшей в более современную)»<sup>1</sup>. В противоположность миграции данных из одного хранилища в другое (с одной аппаратной платформы на другую, из одной СУБД в другую), в данной статье речь пойдет именно о миграции приложения в целом.

Причин миграции может быть множество.

1. Экономическая причина. Стоимость лицензии от мировых лидеров – производителей самых популярных СУБД Oracle и Microsoft велика. А в связи с новой экономической политикой компаний продавать лицензии только на год, причем указывается стоимость лицензии на один процессор, т. е. в случае 4-процессорного сервера (обычная конфигурация) стоимость увеличивается в 4 раза, платить такую цену за СУБД под силу только крупным компаниям. Но, как показывает опыт, крупные компании тоже не всегда идут на это. Так, например, компания «Яндекс» в период с 2012 по 2016 г. планомерно мигрировала все свои сервисы на свободно распространяемую СУБД PostgreSQL<sup>2</sup>.

Заметим, что ПО с открытым кодом не всегда бывает бесплатным – компании, предоставляющие собственные сборки, строят свой бизнес на продаже и поддержке собственных сборок. И все же стоимость такой поддержки не сравнима с ценами, установившимися в связи с новой лицензионной политикой мировых брендов ПО.

2. Политическая причина. Зачастую именно политическая причина играет важную роль в принятии решения о миграции, если речь идет о приложениях, имеющих национальное значение. Уже упомянутая компания «Яндекс» приняла решение о миграции с Oracle еще в 2012 г., до объявления санкций и связанной с ними политикой импортозамещения<sup>3</sup>. Политическая ситуация была названа руководством одной из причин миграции.

3. Вопросы безопасности. Как ни странно, даже компании, предоставляющие решения в области безопасности, предпочитают использовать свободно распространяемое ПО с открытым кодом. Разработчиками ПО в open source (контрибуторами) являются и отдельные энтузиасты, и целые группы разработчиков, и компании. Политика же контрибуции модулей при разработке открытого ПО обычно такова, что модуль при добавлении подвергается многократному аудиту и ревизии, и если в нем присутствовал бы вредоносный код или просто уязвимость, они были бы выявлены на ранней стадии из-за широкого применения пользователями, которые одновременно являются и разработчиками.

Обнаруженные в процессе эксплуатации уязвимости исправляются очень быстро выпуском обновлений безопасности, зачастую в течение нескольких часов с момента обнаружения, в то время как в случае проприетарного ПО известны случаи, когда уязвимость не устранялась в течение длительного времени, пока о ней не становилось известно большому количеству пользователей.

Что касается ОС для серверных платформ, выбор открытого ПО (Ubuntu, CentOS, Debian, Fedora, FreeBSD) давно признан предпочтительным с точки зрения безопасности.

4. Выход свободного ПО на качественно новый уровень. Еще совсем недавно свободно распространяемые продукты, такие, например, как операционные системы Linux, использовались либо исключительно разработчиками ПО, либо только на серверных платформах. Сейчас благодаря Wine, Open Office, Linux Mint пользователи, не являющиеся специалистами в области IT, могут даже не заметить, что на их персональном компьютере установлена Linux, а не Windows.

Что касается платформ для разработки веб-приложений, многие системы, начинавшиеся двадцать лет назад как системы управления содержимым сайтов (CMS – Content Management System) переросли в платформы (frameworks) для разработки CMS. За двадцать лет они проделали путь от нехитрых конструкторов для построения сайтов до фреймворков, содержащих десятки тысяч библиотечных модулей с функциональностью, которая может потребоваться при разработке собственных CMS и веб-приложений.

5. Поддержка. При правильном выборе свободного ПО пользователь получает не только само ПО, но и бесплатную круглосуточную поддержку от членов сообщества, работающих с этим ПО. Поскольку круг пользователей очень широк, вопрос, который может возникнуть

<sup>1</sup> URL: [http://wp.wiki-wiki.ru/wp/index.php/Миграция\\_приложений/](http://wp.wiki-wiki.ru/wp/index.php/Миграция_приложений/).

<sup>2</sup> URL: [http://www.tadviser.ru/index.php/Проект:Яндекс\\_\(миграция\\_с\\_Oracle\\_на\\_PostgreSQL\)/](http://www.tadviser.ru/index.php/Проект:Яндекс_(миграция_с_Oracle_на_PostgreSQL)).

<sup>3</sup> Друзьягин Р. Теория и практика миграции веб-систем на PostgreSQL. 2015. URL: <https://habr.com/post/254667/>.

у одного из них, скорее всего, уже возникал не у одного десятка пользователей, и на форумах сообщества найдется не только формулировка вопроса, но и решение проблемы.

6. Рефакторинг кода. Неотъемлемым свойством жизненного цикла приложения является развитие. По мере добавления все новых и новых возможностей, а также роста объема данных первоначально спроектированная архитектура приложения перестает удовлетворять современным требованиям. Приложение начинает работать медленно, понижается его отказоустойчивость. Миграция приложения является хорошим поводом для пересмотра функциональных модулей системы, его архитектуры и рефакторинга кода.

### **Миграция – прихоть или вынужденная мера?**

Миграция приложений, безусловно, требует затрат и человеческих ресурсов. Даже если речь идет о миграции на другую СУБД с сохранением кода приложения и структуры базы данных, это может стать долгим и трудоемким процессом<sup>4</sup>, поскольку оказывается, что SQL-запросы и хранимые процедуры не переносятся «один в один» из СУБД в СУБД. Более того, даже если речь идет о миграции от версии к версии одной СУБД, задача может оказаться нелегкой и потребовать изучения бизнес-логики самого приложения и переписывания части его кода.

Если же в рамках миграции требуется также и полное переписывание кода приложения в другую технологию, процесс может растянуться на годы. Поэтому миграция приложения – это, прежде всего, вынужденная мера, на которую приходится идти, чтобы сохранить возможность дальнейшей эксплуатации приложения.

Но, как мы покажем далее, даже миграция с полным переписыванием кода вручную требует значительно меньших трудозатрат, чем требовалось на этапе создания исходного приложения. Очевидно, объясняется это тем, что при миграции мы имеем дело с уже сформулированной бизнес-логикой приложения и готовой моделью данных, к тому же проверенными годами эксплуатации. Кроме того, тщательный подход к выбору платформы для миграции позволяет использовать богатый арсенал готовых библиотечных модулей для воссоздания функциональности приложения на новой платформе. И здесь самое время рассказать о выборе платформы Drupal.

### **Платформа Drupal**

Для осуществления миграции была выбрана свободно распространяемая платформа Drupal<sup>5</sup>, которая активно развивается с 2001 г. Первоначально развиваемая как CMS, впоследствии Drupal стала позиционироваться как CMF (framework – платформа). По данным на январь 2018 г., Drupal включает в себя более 39 000 модулей, что сильно ускоряет разработку приложений. Drupal поддерживает все популярные СУБД, отлично справляется с большими проектами, хорошо документирован. Платформа Drupal поддерживается широким сообществом разработчиков в мире (и в России в том числе).

Кроме того, выбор Drupal был также обусловлен тем, что в Институте систем информатики имени А. П. Ершова СО РАН уже был накоплен значительный опыт работы с этой платформой. В период с 2005 г. по настоящее время сотрудниками института реализовано около 30 веб-проектов на этой платформе.

Из недостатков можно отметить подход Drupal к структуре базы данных. В базе данных Drupal хранятся не только данные приложения, но и сама система управления приложением, поэтому создавать структуру базы данных произвольно нельзя – она формируется модулями ядра Drupal и модулями третьих сторон. Работать с базой данных напрямую достаточно сложно (и не рекомендуется).

<sup>4</sup> Бородин В. История успеха Яндекс-почты. URL: [https://pgday.ru/files/papers/61/2016.07.08\\_История\\_успеха\\_Яндекс.Почты.pdf](https://pgday.ru/files/papers/61/2016.07.08_История_успеха_Яндекс.Почты.pdf)

<sup>5</sup> Официальный сайт сообщества Drupal: <https://www.drupal.org/>.

Другим недостатком Drupal является то, что, несмотря на жесткую модерацию модулей, некоторые из них могут вести себя некорректно при определенных условиях либо больше не поддерживаются разработчиками.

### Постановка задачи миграции

В рамках изучения методов миграции на свободное ПО требовалось выполнить миграцию приложений, разработанных несколько десятилетий назад в ИСИ СО РАН, а именно Архива академика А. П. Ершова [1; 2] и системы «Библиотека» [3]. Миграция включает в себя следующие работы, которые могут быть выполнены как поэтапно, так и параллельно:

- 1) воссоздание структуры приложения с сохранением (по возможности) модели данных;
- 2) воссоздание бизнес-логики приложения для различных ролей пользователей, включая неавторизованных пользователей;
- 3) организация рабочих мест для авторизованных пользователей;
- 4) разработка дополнительных возможностей для повышения удобства использования приложений;
- 5) тестирование и отладка на тестовом пуле данных;
- 6) перенос данных на новое приложение.

### Электронный архив академика А. П. Ершова

Электронный архив академика А. П. Ершова [4] – проект ИСИ СО РАН, выполненный при поддержке Microsoft Research. Проект был начат в 2000 г. и реализован в технологиях Microsoft (MS SQL Server, .Net, Microsoft Windows Server, IIS).

После кончины академика А. П. Ершова остался уникальный архив [5]. Это более 500 папок с документами, отражающими жизненный путь академика и историю развития информатики в России [6–9]. В настоящее время архив включает в себя следующие коллекции документов: архив А. П. Ершова, архив С. С. Лаврова, архив ИСИ СО РАН, архив ВНТК «Старт».

В архиве содержится следующая информация:

- документы – 42 386;
- изображения документов – 156 033;
- описанные персоналии – 6 431;
- сведения об организациях – 3 047.

В архиве реализованы два вида просмотра документов: папки и листы, как собирал их сам А. П. Ершов, и разбиение по темам и группам, как распределили их архивисты (рис. 1, 2).

### Система «Библиотека»

Система «Библиотека» разработана Я. М. Курляндчиком в начале 1980-х гг. на БЭСМ-6, затем была перенесена на ПК с использованием средств MS DOS и FoxPro [10]. Система используется до настоящего времени в Мемориальной библиотеке А. П. Ершова для хранения, управления фондами, обработки и публикации новых поступлений в библиотеку. Система «Библиотека» представляет собой десктопное однопользовательское приложение [3]. И база данных, и само приложение находятся на одном компьютере, который представляет собой и рабочее место библиотекаря, и рабочее место читателя. Система написана не в архитектуре клиент-сервер. Это означает, в частности, что она не предоставляет возможности доступа к приложению с другого компьютера. Фактически, база данных представляет собой набор файлов, управляемых специально разработанной системой [11].

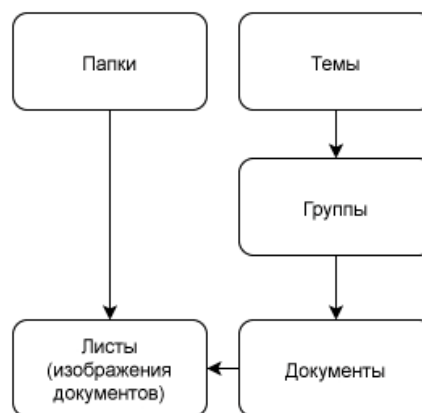


Рис. 1. Способы просмотра документов в архиве А. П. Ершова

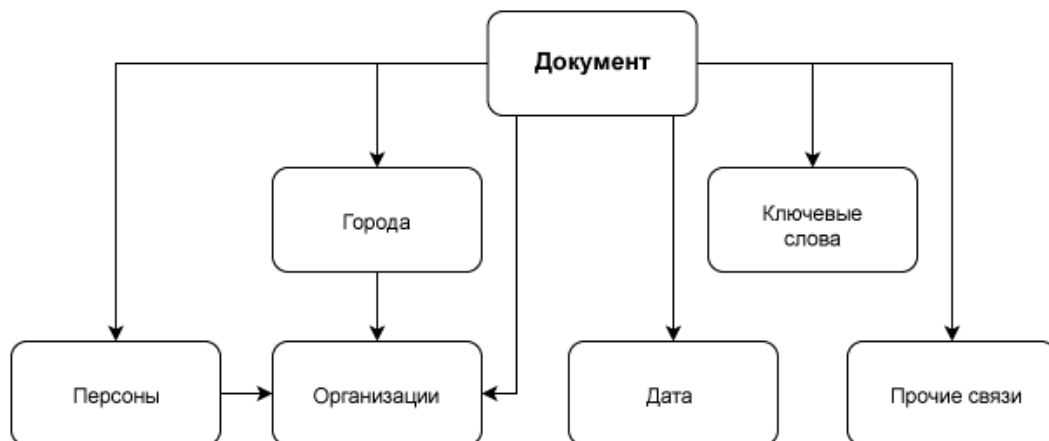


Рис. 2. Архив А. П. Ершова: связь с документами

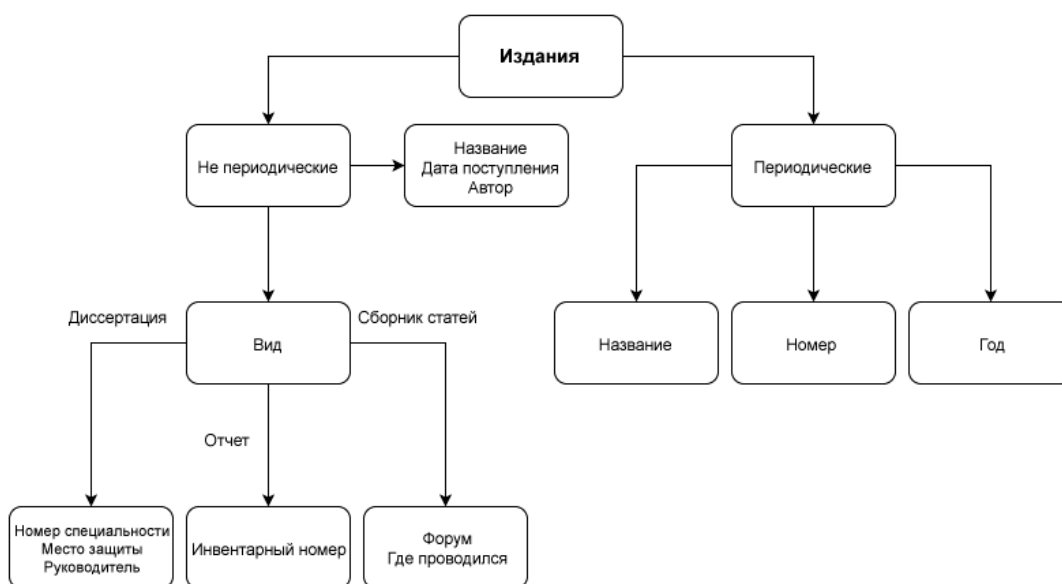


Рис. 3. Система «Библиотека»: модель данных

Радио											
Год	Том	Номер									
2015	1	*2	*3	*4	*5	6	7	8	9	10	
		11	12								
2016	1	2	3	4	5	6					

1 Подсказка 2 Поступление 3 Поиск 4 Год      Просмотр формуляра      Esc Выход

Рис. 4. Таблица номеров журнала «Радио»

В системе содержатся следующие данные:

- журналы – 722 (15 955 номеров);
- описания – 54 646;
- персоналии читателей – 114.

Объекты, с которыми работает Библиотека, – периодические (журналы) и непериодические (описания) издания (рис. 3). Каталог периодических изданий представляет собой список названий журналов. Выпуски журналов представлены таблицей, символом «\*» помечены номера, находящиеся на руках (рис. 4). Описания хранятся в массивах по 1 000 элементов. Каждый массив – это файл с таблицей FoxPro. В описании указываются вид, язык, авторы, УДК, ББК, редактор, редакция, серия, год, название, количество страниц, источник, выпуск, том, ISSN и другие свойства описываемого фонда библиотеки.

### Разработка приложения

Перед тем как приступить к миграции приложения «Библиотека», был изучен и обобщен опыт построения библиотечных систем<sup>6</sup>.

Для сохранности данных приложения миграцию не следует проводить из работающего приложения. Непосредственно перед осуществлением миграции приложения пришлось осуществить аппаратную миграцию работающего приложения «как есть», а именно создать виртуальную машину с окружением, необходимым для запуска исходного приложения, затем скопировать данные приложения и запустить (эмулировать) приложение в современном окружении. С этой целью для приложения «Библиотека» была создана виртуальная машина, что позволило осуществить удаленный доступ к копии исходного приложения.

Для приложения «Архив академика А. П. Ершова» была скопирована база данных на виртуальную машину с СУБД MS SQL SERVER. Только после этого стало возможным приступить к первому этапу разработки приложения – воссозданию модели данных.

Требовалось проанализировать исходную модель данных и полностью воссоздать ее на платформе Drupal. Работа состояла из нескольких этапов:

- 1) создание типов сущностей;
- 2) создание словарей таксономии;
- 3) настройка связей между сущностями и терминами таксономий;
- 4) разграничение уровня доступа для пользователей.

В случае с системой «Библиотека» требовалось не просто воссоздать модель данных, но и формализовать ее, поскольку все данные в системе были строковыми и хранились нерационально.

### Миграция данных в новое приложение

Как говорилось ранее, Drupal не позволяет напрямую работать с базой данных. При миграции каждой сущности для Drupal нужно указывать сразу все связи этой сущности с другими, после чего Drupal распределит их по таблицам в базе данных. Это очень удобно для пользователей приложений, чтобы заполнять данные о конкретной сущности, но вызывает сложности при миграции. По этой причине очень полезным оказался фреймворк Migrate, который позволяет писать собственные модули для миграции (рис. 5). Migrate поддерживает миграцию из всех популярных CMS и БД в Drupal, а также между версиями Drupal. Каждый модуль – это расширение класса Migrate на языке PHP.

Для каждого типа сущности был написан модуль, который преобразует извлеченные данные и записывает их в соответствующие поля сущностей Drupal. Миграции запускаются поочередно, итеративно, в зависимости от модели данных. Есть возможность тестирования миграций: включение миграции на небольшом объеме данных, остановка миграции, откат изменений, вызванных конкретной миграцией.

---

<sup>6</sup> Проект ELIBCONSULT: создание электронной библиотеки от проекта до реализации (<http://www.elibconsult.ru/>).



Рис. 5. Схема работы модуля Migrate

### Подводные камни миграции

Во время миграции приложений пришлось столкнуться с определенными сложностями, которые были вызваны структурой исходных данных, а также особенностями самого метода миграции.

- В архиве А. П. Ершова данные, относящиеся к одной сущности, зачастую хранились в разных таблицах, такие таблицы приходилось отыскивать для каждой сущности и соединять операцией JOIN.

- В системе «Библиотека» все данные были строковые, т. е. не подразумевали использование словарей, типизации. Пришлось определить типы данных и на их основе составить словари для связей.

- В системе «Библиотека» после миграции возникло больше количество дубликатов сущностей. Дублировались сущности, которые библиотекарь вводил с ошибками или используя разные варианты написания и сокращения. После миграции были написаны скрипты на языке PHP, которые позволили объединить дубликаты сущностей и восстановить связи между ними.

- Платформа Drupal может работать только с кодировкой UTF-8. Архив А. П. Ершова работал в среде MS SQL, где использовалась стандартная кодировка Windows CP1251. В системе «Библиотека» использовалась DOS кодировка CP866. Были написаны функции, декодирующие данные из исходных кодировок в UTF-8 перед записью в базу данных.

### Заключение

По итогам работы предложенный метод миграции был исследован и рекомендуется для использования в дальнейшем.

Благодаря поддержке словарей таксономии на уровне ядра Drupal отлично подходит для систем со сложной моделью данных, с большим количеством сущностей и связей между ними (архивы, библиотеки, каталоги). Ввиду высоких требований Drupal к аппаратным ресурсам следует с осторожностью принимать решение о применении предложенного метода для миграции высоконагруженных систем с большим количеством одновременных пользователей.

Предложенный метод показал весомое снижение трудозатрат на написание приложения. Исходное приложение «Архив академика А. П. Ершова» разрабатывалось командой из четырех разработчиков на протяжении нескольких лет, в то время как разработка на Drupal заняла несколько месяцев у одного разработчика. Дальнейшее ведение проектов архивист (либо библиотекарь) может осуществлять самостоятельно, не прибегая к помощи разработчика.

Поскольку платформа Drupal широко распространена и поддерживается мощным сообществом, в том числе в России, найти разработчика для поддержки, сопровождения и дальнейшего развития проекта не составляет проблемы.

В рамках миграции по каждому приложению были выполнены следующие задачи:

- воссоздана модель данных исходного приложения в окружении Drupal;
- разработаны веб-приложения с набором тестовых данных, создан интерфейс для пользователей и редакторов, настроены права доступа для пользователей;
- проведено тестирование новых приложений на выборке данных;
- разработаны механизмы для миграции данных исходных приложений в БД новых приложений с воспроизведением модели данных;
- осуществлен ряд итераций для миграции данных;
- приложение «Архив академика А. П. Ершова» находится в работе два года, доступно по адресу <http://ershov.iis.nsk.su/>;
- приложение «Библиотека» в настоящий момент находится в стадии тестовых испытаний.

### Список литературы

1. Филиппов В. Э. и др. Электронный архив академика А. П. Ершова – методика создания и научной интерпретации // Информационные технологии в гуманитарных исследованиях. Новосибирск, 2006. Вып. 11. С. 51–56.
2. Филиппов В. Э., Крайнева И. А., Филиппова М. Я., Черемных Н. А. Интернет-технологии как средство сохранения и публикации материалов научного, культурного и исторического наследия на примере электронного архива академика А. П. Ершова. Улан-Удэ, 2003. С. 257–261.
3. Курляндчик Я. М. Система «БИБЛИОТЕКА». Руководство по использованию. Новосибирск, 1991.
4. Antiufeev S., Boulyonkova A., Kraineva I., Nemov A. Creation and Scientific interlretation of Acad. Ershov's Electronic Archive // The Fourth SEEDI Conference «Digitization of Cultural and Scientific Heritage». Belgrade, Serbia, 2008. Book of Abstracts, li.11.
5. Крайнева И. А., Черемных Н. А. Личный архив академика А. П. Ершова в Интернете // Отечественные архивы. 2001. № 5. С. 53–55.
6. Крайнева И. А., Черемных Н. А. Научное наследие академика А. П. Ершова // Проблемы культурного наследия в области инженерной деятельности / Под ред. Г. Г. Григоряна. М.: Информ-Знание, 2007. Вып. 5. С. 140–172.
7. Крайнева И. А. Архив академика А. П. Ершова как источник по социальной истории научного сообщества // Документ в парадигме междисциплинарного подхода: Материалы II Всерос. науч.-практ. конф. Томск: ТГУ, 2006. С. 146–149.
8. Крайнева И. А., Черемных Н. А. Академик А. П. Ершов и его архив // Развитие вычислительной техники в России и странах бывшего СССР: история и перспективы (SORUCOM 2006): Материалы Междунар. конф. Петрозаводск, 2006. Ч. 2. С. 50–56.
9. Крайнева И. А. Научная биография академика А. П. Ершова: Автореф. дис. ... канд. ист. наук: 07.00.10. Томск, 2008. 32 с.
10. Курляндчик Г. В. Судьба семьи в эпоху компьютеров // Sorucom-2014: Материалы Междунар. конф. Казань, 2014.
11. Курляндчик Я. М. Система управления файлами. М., 1980. 20 с. (Препринт / ИТМ и ВТ АН СССР; № 13).

Материал поступил в редколлегию 10.04.2018

**S. N. Troshkov**

*Novosibirsk State University  
1 Pirogov Str., Novosibirsk, 630090, Russian Federation*

*kamronis@yandex.ru*

### ON EXPERIENCE IN MIGRATING APPLICATIONS TO THE FREELY DISTRIBUTABLE OPEN SOURCE SOFTWARE

An important problem of modern programming is the support and maintenance of legacy software. The functionality of applications written in older environments is valuable and still relevant.



Outdated software environments do not allow to use the applications on modern machines and prevents their further development.

The paper describes the experience of migration to open source software. The migration was performed for two applications: the Archive of academician A. P. Ershov and the Library system. These applications work and have been used in Ershov Institute of Informatics Systems of SB RAS for a number of years. CMF Drupal has been chosen as a free and open source platform for creating new applications. The advantages of CMF Drupal facilitates significantly the development of new application and migration of the data model. Migration included reengineering the application while preserving business logic, the data model, and migrating the data itself.

*Keywords:* migration, database, data model, reengineering, drupal, ms sql server, fox pro, mysql, postgresql.

## References

1. Fillipov V. E. et al. Electronic archive of academician A. P. Ershov as the methodology of creation and scientific interpretation. *Information technologies in humanitarian researches*. Novosibirsk, 2006, vol. 11, p. 51–56. (in Russ.)
2. Filippov V. E., Kraineva I. A., Filippova M. Ya., Cheremnykh N. A. Internet technologies as means of preservation and publication of materials of scientific, cultural and historical heritage by the example of academician A. P. Ershov's electronic archive. Ulan-Ude, 2003, p. 257–261. (in Russ.)
3. Kurlyandchik Ya. M. «LIBRARY» system. User's Guide. Novosibirsk, 1991. (in Russ.)
4. Antiufeev S., Boulyonkova A., Kraineva I., Nemov A. Creation and Scientific interpretation of Acad. Ershov's Electronic Archive. *The Fourth SEEDI Conference «Digitization of Cultural and Scientific Heritage»*. Belgrade, Serbia, 2008. Book of Abstracts, li.11.
5. Kraineva I. A., Cheremnykh N. A. Personal archive of academician A. P. Ershov on the Internet. *National archives*, 2001, no. 5, p. 53–55. (in Russ.)
6. Kraineva I. A., Cheremnykh N. A. Scientific heritage of academician A. P. Ershov. *Problems of cultural heritage in the field of engineering*. Ed. by G. G. Grigoryan. Moscow, Inform-Znanie, 2007, vol. 5, p. 140–172. (in Russ.)
7. Kraineva I. A., Cheremnykh N. A. Archive of academician A. P. Ershov as a source on social history of scientific community. *Second Russian scientific-practical conference «Document in the paradigm of interdisciplinary approach»*. Tomsk, TSU Press, 2006, p. 146–149. (in Russ.)
8. Kraineva I. A., Cheremnykh N. A. Academician A. P. Ershov and his archive. *Proceedings of the international conference «Development of computer technology in Russia and the former Soviet Union: history and prospects (SORUCOM 2006)»*. Petrozavodsk, 2006, pt 2, p. 50–56. (in Russ.)
9. Kraineva I. A. Scientific library of academician A. P. Ershov. Dis. ... candidate of Historical Sciences: 07.00.10. Tomsk, 2008, 32 p. (in Russ.)
10. Kurlyandchik G. V. The fate of the family in the era of computers. *Proceedings of the international conference Sorucom*. Kazan, 2014. (in Russ.)
11. Kurlyandchik Ya. M. File management system. Moscow, 1980, 20 p. (Preprint / IPMCE AS USSR; № 13). (in Russ.)

*For citation:*

Troshkov S. N. On Experience in Migrating Applications to the Freely Distributable Open Source Software. *Vestnik NSU. Series: Information Technologies*, 2018, vol. 16, no. 2, p. 86–94. (in Russ.)

DOI 10.25205/1818-7900-2018-16-2-86-94