

## СТРУКТУРЫ ДАННЫХ ДЛЯ МОДЕЛИ ТРЕНАЖЕРА ОПЕРАТИВНЫХ ПЕРЕКЛЮЧЕНИЙ

**Abstract.** Considered technological process of development the operational switching simulator. Allocated individual steps in the process simulator development. Data structures are proposed for forming and calculation simulator model. Software tool for formation of the model data is presented.

### Этапы разработки тренажера оперативных переключений

Предлагается выделить в процессе разработки тренажера оперативных переключений специалистами технологами следующие этапы:

- сбор и обработка исходных данных;
- разработка модели коммутационной структуры;
- разработка сцен;
- разработка модели навигации;
- отладка модели коммутационной структуры;
- разработка модели автоматике;
- разработка сценариев тренировок;
- комплексная отладка.

Работы выполняемы на каждом из предложенных этапов следующие:

#### 1. Сбор и обработка исходных данных

На данном этапе собираются и систематизируются исходные данные, которые будут использоваться при разработке тренажера. К таким данным относятся: должностные инструкции персонала, оперативные схемы, планы подстанций, бланки переключений, паспорта оборудования, карты уставок РЗА, фотографии реального оборудования по месту, параметры работы оборудования, возможные сценарии тренировок и др. После выполнения систематизации исходных данных выполняется оценка объема моделирования, формируются списки задействованного оборудования, элементов РЗА, согласовываются сценарии тренировок.

#### 2. Разработка модели коммутационной структуры

Модель коммутации включает список элементов тренажера с соответствующими характеристиками. Список элементов формируется по оперативным схемам и мнемосхемам подстанций. Элементами модели являются источники, выключатели, разъединители, шины, трансформаторы, нагрузки и т.п. Также модель коммутационной структуры включает связи между элементами, узлами модели.

#### 3. Разработка сцен

Разрабатывается перечень сцен, которые будут использоваться в тренажере и начинается формирования этих сцен. Каждая сцена представляет собой

отдельный swf-файл, которые будут вызываться главной программой во время работы тренажера. На сцене размещается статическая графическая подложка – фон сцены, и над фоном устанавливаются элементы ввода-вывода, которые копируются из готовой БИБЛИОТЕКИ. Для каждого элемента ввода-вывода указывается идентификатор элемента модели и имя свойства элемента модели. Элементы ввода-вывода будет связан с данным элементом модели и свойством во время выполнения.

#### 4. Разработка модели навигации

Модель навигации обеспечивает переходы между составными частями тренажера и включает сцены, переходы между сценами и правила по которым эти переходы осуществляются. Детально построение модели навигации описано в [2].

#### 5. Отладка модели коммутационной структуры

Выполняется проверка и отладка модели коммутационной структуры в тренажере на соответствие реальным параметрам. Для этого этапа необходимо разработать сцены со схемами основного оборудования (п.2) и связать их навигацией (п.3). Работы по этапам 3,4 могут быть не завершены до начала этого этапа.

#### 6. Разработка модели автоматки

После окончания отладки модели коммутационной структуры (п.4) разрабатывается модель автоматки. Модель автоматки включает набор элементов, правил автоматки разных типов и их свойств.

#### 7. Разработка сценариев тренировок

На данном этапе выполняется разработка и программирование нормальных и аварийных сценариев тренировок.

#### 8. Комплексная отладка

Комплексная отладка тренажера это внесение правок и коррективов во все составляющие. Во время выполнения комплексной отладки возможен возврат на любой из предыдущих этапов для устранения всех возможных неточностей или ошибок.

Технологический процесс разработки тренажера можно представить в виде блок-схемы изображенной на рис.1.

Предполагается, что работы будут проводить не программисты, а технологи со знаниями в сфере эксплуатации энергетического оборудования. Для ускорения процесса разработки необходима возможность коллективной работы, что обеспечивается разделением общей модели тренажера отображения на составные части, например по отдельным подстанциям. Также ускорить разработку может запараллеливания некоторых этапов, например разработки модели коммутационной структуры, работки сцен и модели навигации.

Технология разработки тренажеров «Модус» описана на сайте компания Модус [1]. Даная технология предполагает разбиение модели тренажера на несколько составных частей, разделение модели и наборов данных для расчёта, независимую разработку модели и сцен тренажера.

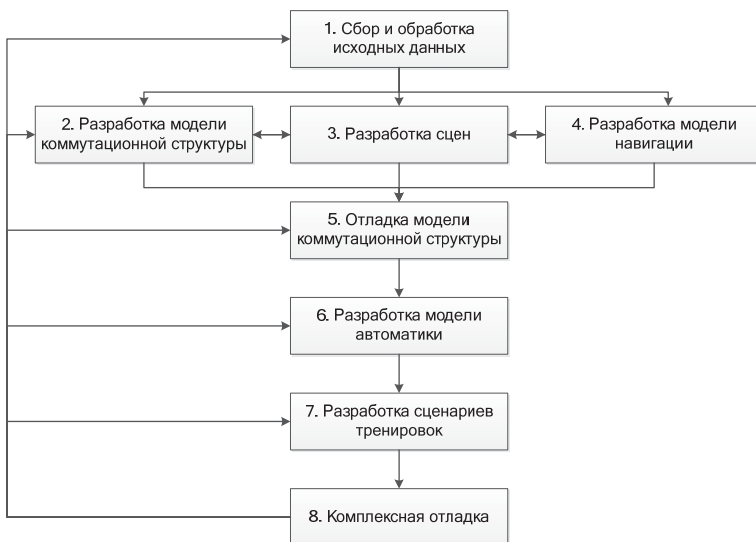


Рис. 1. Блок-схема технологического процесса разработки тренажера

### **Разработка модели коммутационной структуры**

Данные для расчета модели коммутации или, другими словами, коммутационной структуры, тренажера хранятся во внешних структурированных файлах. В качестве формата внешних файлов используется xml. Работа технологов на данном этапе заключается в наполнении этих файлов данными

Преимущество использования формата xml:

- структурированный формат с формальным синтаксисом;
- возможность работы разработчика тренажера с xml-файлом с помощью любого текстового редактора, например стандартной программы «Блокнот»;
- простота программной обработки xml-файлов, наличие в Actionscript3/Flash стандартных библиотечных функций по работе с xml-файлами;
- при необходимости возможность перехода на использования реляционных БД вместо xml-файлов, так как формат xml-файлов строго структурирован.

При запуске тренажера программа загружает информацию о моделях из внешних файлов. Такой подход обеспечивает независимость программного обеспечения тренажера и данных, необходимых для расчета модели. Специалист технолог может, изменив данные настроить или перестроить модель по необходимости внесения изменений в код программного обеспечения тренажера. То есть используя готовое программное обеспечение и разные наборы данных можно разработать несколько тренажеров. Также

выделения данных в отдельные файлы позволяет разрабатывать отдельно от самого тренажера инструментарий для напыления этих файлов.

На первом шаге загружается файл с перечнем объектов, которые будут моделироваться в тренажере. Данные обо всех загруженных объектах формируется в массив.

Пример структуры описывающей один объект модели:

```
<modelElement>
  <identifier>DOB_SR35_B2</identifier>
  <type>breaker</type>
  <operationalName>CP-35 кВ II СШ
                                </operationalName>
  <description>Секційний роз'єднувач 35 кВ II СШ
                                </description>
</modelElement>
```

Тэг <identifier> - определяет идентификатор объекта. Правила построения и использования идентификаторов описаны далее.

Тэг <type> - определяет тип объекта. Возможные типы объектов с описанием приведены в таблице 1:

Таблица 1

Возможные типы объектов

Название типа	Описание
<b>Основные типы объектов</b>	
breaker	разъединитель
switcher	выключатель
busbar	шина
source	источник напряжения
transformer	трансформатор
auxpoint	точка соединения, узел
load	нагрузка
<b>Дополнительные типы объектов</b>	
feeder	фидер, ячейка, состоит из разъединителя, выключателя, разъединителя
voltagegens	трансформатор напряжения, используется для снятия значения напряжения на шине
currentgens	трансформатор тока, используется для снятия значения тока
groundzone	место установки заземления
rvs	разрядник перенапряжения

Тэг <operationalName> - определяет оперативное, диспетчерское наименование объекта.

Тэг <description> - определяет развернутое текстовое описание объекта.

Также для некоторых типов объектов могут использоваться

дополнительные теги: <ratio>, <load>, <znOperationalName> и другие для описания свойств конкретных типов объектов.

### **Идентификатор объекта**

Идентификатор объекта – это уникальная в рамках модели символьная строка. Во время загрузки каждого объекта из внешнего файла идентификатор объекта проверяется на уникальность. Если в модели уже присутствует объект с таким идентификатором – текущий объект не будет загружен, а пользователю будет выдано сообщение об ошибке загрузки.

Используется следующая структура идентификатора объекта:

DOB\_SR35\_B2

(1) (2) (3)

(1) – сокращенное наименование подстанции, на которой находится объект (например: DOB - Добромьль; LV21 – Львов-21);

(2) – тип объекта и класс напряжения (например: SR35 – секционный разъединитель 35 кВ, V110 – выключатель 110 кВ);

(3) – название присоединения (например: B2 – присоединение на 2 секцию шин; T2 – присоединение на трансформатор T2).

Такое представление обеспечивает наглядность и удобство работы технологов, так как идентификаторы соответствуют диспетчерским наименованиям, с которыми технологи привыкли работать.

Данный подход к формированию идентификаторов позволяет с «одного взгляда» определить объект, его тип, класс напряжения, месторасположение. Например:

DOB\_SR35\_B2 – подстанция Добромьль, секционный разъединитель 35 кВ на II секцию шин.

LV1\_T1 – подстанция Львов-1, трансформатор T1.

KR\_BR110\_T1 – подстанция Красне, шинный разъединитель 110 кВ на трансформатор T1.

На втором шаге, после загрузки информации обо всех объектах модели, формируется массив узлов, для которых будет проводиться расчет. Для всех типов объектов, кроме выключателей, разъединителей и трансформаторов, автоматически создается по одному узлу, с тем же идентификатором, что и у объекта. Для выключателей и разъединителей автоматически создается по два узла. Идентификаторы для этих узлов формируется следующим образом: к идентификатору объекта добавляются окончания `_VP` и `_VD`. Например, для разъединителя с идентификатором `DOB_SR35_B2` будет создано два узла `DOB_SR35_B2_VP`, `DOB_SR35_B2_VD`. Аналогично, для каждого трансформатора будет автоматически создано по 3 узла со следующими окончаниями идентификаторов `_VVN`, `_VSN`, `_VNN` (на украинском языке: «вузол високої напруги» - `VVN`, «вузол середньої напруги» - `VSN`, «вузол низької напруги» - `VNN`). Двух- и трехобмоточные трансформаторы никак не различаются, в каждом случае будет создано по 3 узла.

После автоматического формирования массива узлов происходит загрузка файла со связями между узлами, это третий шаг. Файл связей имеет достаточно простую структуру, это просто пары идентификаторов узлов, между которыми есть связь. В теории графов такой подход к хранению структуры называется «Списком ребер». Данный способ представления является наиболее компактным, так как требует наименьшее количество памяти для хранения. Пары идентификаторов определяются тэгами <element1> и <element2>. Пример структуры описывающей связь:

```
<elementConnection>
  <element1>DOB_B35_2</element1>
  <element2>DOB_SR35_B2_VD</element2>
</elementConnection>
```

Данный блок определяет, что существует связь между: DOB\_B35\_2 – секцией шин II 35 кВ подстанции Добромоль и DOB\_SR35\_B2\_VD – вторым узлом секционного разъединителя 35 кВ на секцию шин II подстанции Добромоль.

Еще один пример:

```
<elementConnection>
  <element1>LV21_V110_T2_VD</element1>
  <element2>LV21_T2_VVN</element2>
</elementConnection>
```

Данный блок определяет, что существует связь между: LV21\_V110\_T2\_VD – вторым узлом выключателя 110 кВ на трансформатор T2 подстанции Львов-21 и LV21\_T2\_VVN – узлом высокого напряжения трансформатора T2 подстанции Львов-21.

Четвертый шаг – загрузка начального состояния объектов. Из отдельного файла загружаются начальные состояния объектов. Структура, описывающая начальное состояние объекта, выглядит следующим образом:

```
<modelElement>
  <identifier>DOB_SR35_B2</identifier>
  <conduction>0</conduction>
</modelElement>
```

Тэг <identifier> определяет идентификатор объекта. Далее следуют тэги соответствующие свойствам объекта и необходимые значения. Возможные свойства объектов <conduction> – проводимость (включенное/выключенное положение), <load> – значение нагрузки, <damage> – повреждение оборудования, и другие.

Предполагается возможность создания и загрузки в тренажер неограниченного количества файлов исходного состояния, которые могут моделировать состояния сети, например нормальный режим, разные аварийные ситуации, выведение оборудования в ремонт и т.д.

## Расчет модели

После загрузки всех составляющей модели коммутации, описанных выше и загрузки исходного состояния элементов возможен расчет модели, а именно напряжений в узлах и токов через коммутационные аппараты. Тренажер предполагает подключение и использование известных алгоритмов для расчета модели. Использование алгоритм расчета модели методом решения СЛАУ описано в [3].

## Инструментарий для работы со структурами

Работа в «блокноте» со структурами в xml-формате требует внимательности, так как, любые опечатки могут нарушить синтаксис xml-файла и программа не сможет его корректно обработать. Для упрощения работы технологов, разработчиков тренажера и для уменьшения количества возможных ошибок, был разработан редактор, который позволяет создавать и редактировать xml-файлы с описанием объектов и связей между ними, в соответствии с предложенными структурами. Использование разработанного приложения позволяет избежать синтаксических ошибок и существенно увеличивает скорость разработки коммутационной модели.

Общий вид редактора изображен на рис.2:

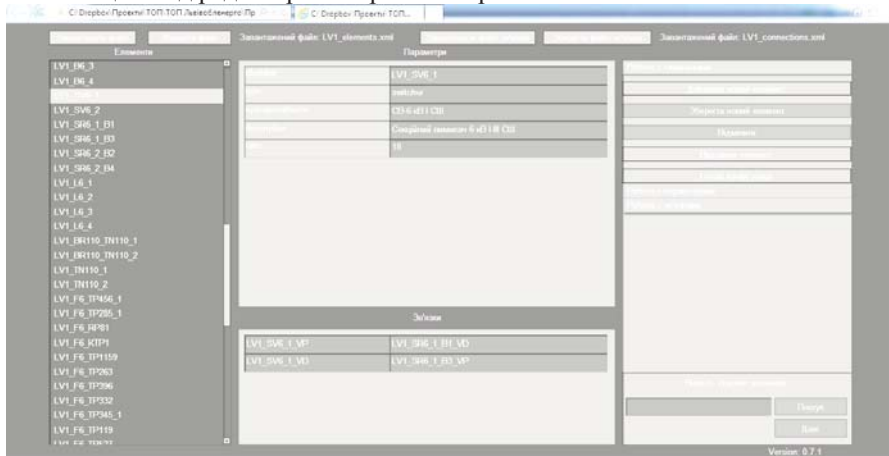


Рис. 2. Общий вид редактора коммутационной структуры

Данный редактор позволяет создавать и редактировать xml-файлы со структурами описанными выше. Основные функции, которые выполняет данное программное обеспечение следующие: просмотр, создание и удаление элементов модели перечисленных типов, просмотр, создание, редактирование и удаление параметров элементов, формирование связей между элементами. Также предусмотрена возможность вставки типовых конфигураций элементов вместе со связями. Например, ячейки ПС, то есть выключателями и двух разъединителей. Для уменьшения количества возможных ошибок в

программе редактора предусмотрены функции проверки правильности введенных данных, такие как проверка максимального количества связей для конкретного элемента, подключение вводов трансформатор и другие.

### **Выводы**

Рассмотрен технологичный процесс разработки тренажера оперативных переключений. Выделены отдельные этапы в процессе разработки тренажера. Предложены структуры данных для формирования и расчета модели коммутации тренажера. Представлен программный инструментарий для формирования данных модели.

1. *Амелин С.В.* Технологии моделирования и макетирования объектов электроэнергетики. // <http://swman.ru/content/blogcategory/29/57/>.
2. Кубарьков Ю.П., Гольдштейн В.Г., Амелин С.В. Задачи моделирования и расчёта режимов электротехнических комплексов и автономных систем электроснабжения // Вестник Самарского государственного аэрокосмического университета. № 1 (25) 2011 – С.162 – 169.
3. *Абрамович Р.П., Бальва А.А., Самойлов В.Д.* Построение модели навигации для компьютерных тренажеров и приложений сценарного типа. // , Электронное моделирование. 2014.Т.36.№1. –С. 97 – 105.
4. *Самойлов В.Д., Абрамович Р.П.* Поиск токов в коммутационных структурах решением СЛАУ. // Электронное моделирование. 2013.Т.35.№1. –С. 95 – 107.

*Поступила 14.09.2016 р.*

УДК 620.9.338.242.4

О.М. Джигун, к.т.н., с.н.с, А.В. Ониськова, м.н.с., ІПМЕ ім. Г.Є. Пухова НАН України, м.Київ

## **ТЕНДЕНЦІЇ РЕФОРМУВАННЯ РИНКІВ ЕЛЕКТРОЕНЕРГІЇ ПРОВІДНИХ КРАЇН СВІТУ**

**Анотація.** Розглянуто шляхи реформування ринків електроенергії в провідних країнах світу.

**Анотация.** Рассмотрены пути реформирования рынков электроэнергии в ведущих странах мира.

**Abstract.** The ways of reforming the electric power markets in the leading countries of the world are considered.

**Ключові слова:** мережа ліній електропередачі, енергоринок, реформа електроенергетики.

**Ключевые слова:** сеть линий электропередачи, энергорынок, реформа