

## СХЕМА ГЕНЕРАЦИИ И ВОССТАНОВЛЕНИЯ ДАННЫХ ПРИ ЛОКАЛИЗАЦИИ ДЕФЕКТА

Моженкова Е.В., Пармонов А.И.

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Пармонов А.И. – к.т.н., доцент

Для локализации дефектов работы корпоративных информационных систем на стороне разработчика предлагается реализовать механизм импорта/экспорта данных клиента, разработав автоматизированную информационную систему «Агрегатор данных». Механизм передачи данных на сторону разработчика будет осуществляться через конфигурационный файл. Предложена структура такого конфигурационного файла для генерации и восстановления данных при локализации дефектов на стороне клиента корпоративной сети.

Для обеспечения взаимодействия банка данных (БД) клиента и разработчика в системе обработки принято использовать конфигурационный файл. Схема выгружаемых данных должна представлять собой перечень полей выгружаемой структуры и их типы. Поля выгрузки могут быть сложно структурированными. В схеме выгружаемых данных могут появляться зависимости, которые усложняют логический разбор данных бизнес-объектов системы.

Состояние БД сохраняется в формате метаданных, поэтому структура конфигурационного файла оформляется в формате XML-документа.

Общее описание процесса обмена данными можно описать следующим образом:

- 1) под обработкой данных БД понимается экспорт/импорт объектов, где каждый объект относится к заранее определенному типу данных и характеризуется набором свойств. В общем случае значения свойств конкретного объекта могут храниться в нескольких таблицах;
- 2) при экспорте данные из базы данных (БД) выгружаются в XML-документ определенной структуры. В одном XML-документе могут содержаться объекты разных типов, при этом возможно существование взаимосвязей между объектами;
- 3) Описание типов данных будет храниться в специальной таблице, чтобы можно было настраивать для каждого типа данных, значения каких свойств объектов экспортировать или импортировать для каждой операции экспорта/импорта создается соответствующий протокол, содержащий краткую информацию об экспортированных/импортированных объектах;
- 4) при выполнении экспорта обеспечивается возможность подхвата для экспортируемых объектов для каждого типа данных могут экспортироваться как все объекты из БД, так и только определены при выполнении импорта данных обеспечивается просмотр содержимого XML-файла, фильтрация импортируемых объектов, возможность замещения существующих в БД объектов, либо создание новых объектов, а также сохранение ссылочной целостности между объектами;
- 5) при необходимости экспортированные/импортированные объекты можно удалить через протоколы экспорта/импорта данных;
- 6) файл, содержащий экспортированные данные, является правильным XML-документом.

Для хранения объектов БД в XML-документе будут использоваться следующие XML-элементы:

- <Data\_Root> – корневой узел XML-документа;
- <Descript> – описание содержимого XML-файла. В свою очередь имеет атрибуты: ExpSet\_Code (код настройки экспорта) и ExpSet\_Name (наименование настройки экспорта, по которой производился экспорт данных). Содержит описание данных, хранящихся в XML-документе;
- <MetaData> – метаописание данных. Используется для хранения описания данных. Содержит элементы Class;
- <Time> – родительский узел для хранения временных интервалов;
- <Filtr> – родительский узел для хранения данных фильтров менеджера сущностей;
- <Data> – родительский узел для хранения объектов. Используется для хранения данных. Без атрибутов (Sobj). Содержит элементы Collection;
- <Class> – класс объектов. Содержит описание класса объектов. В свою очередь имеет атрибуты: name (имя класса, должно быть уникальным для всех элементов Class), caption (наименование/заголовок класса, должно быть уникальным для всех элементов Class) и manager (признак менеджера сущностей). Включает в себя элементы Collection;
- <Collection> – коллекция однородных элементов. Используется для группировки однородных элементов. Имеет такие атрибуты: name (имя коллекции), caption (наименование/заголовок коллекции) и child\_class\_id (тип элементов, которые

могут входить в состав коллекции, должно быть одно из значений: "property", "prop\_value" или наименование класса объектов);

- <Object> – объект БД. Используется для хранения объекта экспорта/импорта. Имеет атрибуты: name (наименование объекта, используется при просмотре содержимого XML-документа), class\_id (имя класса, к которому объект относится) и id (уникальный в пределах класса, к которому относится объект, идентификатор объекта, используется для построения взаимосвязей между объектами). Содержит элементы Collection;

- <property> – свойство. Используется при описании класса объектов. Имеет атрибуты: name (имя свойства), caption (наименование/заголовок свойства), datatype (тип данных) и rft\_classes (список связанных классов объектов);

- <prop\_value> – значение свойства. Используется для хранения значения свойства объекта. Имеет атрибуты: prop\_name (имя свойства), rft\_class (имя класса объекта, на который ссылается свойство).

Структура конфигурационного файла проектировалась с учетом формальной модели информационных потоков и описания модели этих потоков в корпоративной сети [1, 2]. Фрагмент структуры файла представлен на рисунке 1.

```
<?xml version="1.0" encoding="windows-1251" ?>
<Data_Root>
  <Descript ExpSet_Code="1" ExpSet_Name="Настройка для экспорта алгоритмов">
    <![CDATA[описание XML-документа]]>
  </Descript>
  <MetaData>
    <Class name="AlgView" caption="Алгоритм представления">
      <Collection name="Properties" caption="Свойства" child_class_id="property">
        <property name="Name" caption="Наименование" datatype="0" rft_classes="" />
        <property .../>...<property .../>
      </Collection>
      ...
      <Collection .../>...</Collection>
    </Class>
    ...
    <Class .../>...</Class>
  </MetaData>
  <Time>...</Time>
</Filtr>...</Filtr>
  <Data>
    <Collection name="Data.AlgView" caption="Алгоритмы представления">
      <Object name="Период (1)" class_id="AlgView" id="2">
        <Collection name="Prop_Values" caption="Свойства" child_class_id="prop_value">
          <prop_value prop_name="name">
            <![CDATA[Период]]>
          </prop_value>
          ...
          <prop_value>...</prop_value>
        </Collection>...</Collection ...>
        ...
      </Object>
      ...
      <Object .../>...</Object>
    </Collection>
    ...
    <Collection .../>...</Collection>
  </Data>
</Data_Root>
```

Рисунок 1 – Структура конфигурационного файла

Состав элементов конфигурационного файла спроектирован на основе абстракции предметной области и имеет обобщенную структуру. В ходе интеграции с другими информационными системами в корпоративной сети и в результате опытной эксплуатации АИС «Агрегатор данных» возможно изменение конфигурации - введение дополнительных тегов и их атрибутов.

**Список использованных источников:**

1. Моженкова, Е. В. Модель построения общих требований по обработке банков данных в задачах сопровождения систем / Е. В. Моженкова, А.И. Парамонов // Компьютерные системы и сети: материалы 54-й науч. конф. аспирантов, магистрантов и студентов, Минск, 23 – 27 апреля 2018 г. / Белорус. гос. ун-т информатики и радиоэлектроники. – Минск, 2018. – С. 92 – 93.

2. Моженкова, Е. В. Модель представления информации в корпоративных информационных системах / Е. В. Моженкова, А. И. Парамонов // Информационные технологии и системы 2018 (ИТС 2018) = Information Technologies and Systems 2018 (ITS 2018) : материалы межд. науч. конф., Минск, 25 октября 2018 г. / Белорус. гос. ун-т информатики и радиоэлектроники ; редкол. : Л. Ю. Шилин [и др.]. – Минск, 2018. – С. 228 - 229.