

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ПРЕДСТАВЛЕНИЯ ОБЪЕКТОВ ЭНЕРГОСЕТИ В ВИДЕ *CIM XML*

Филиал «Протвино» университета «Дубна»
Кафедра информационных технологий

Работа посвящается изучению *CIM*-моделей, а также созданию программного обеспечения для представления информации конфигурационной базы данных энергообъектов в виде *CIM XML*. Внедрение этого программного обеспечения позволит обмениваться топологической информацией электросети в международном стандарте *CIM* с программным обеспечением различных производителей.

Для осуществления контроля над состоянием электросети и поддержания её работоспособного состояния используются автоматизированные системы диспетчерского управления. Объектный сервер телемеханики (программный комплекс, выполняющий услуги по получению данных и управлению ими) является телемеханическим программным продуктом компании «СИСТЕЛ». Сервер хранит данные в виде энергообъектов, имеющих набор параметров в соответствии с их типом, поддерживает топологическую связанность объектов. Топология позволяет наглядно представить пользователю (диспетчеру) электросеть и её состояние. Описание электрической сети в виде объектов, их атрибутов и отношений хранится в специализированной базе данных *MS SQL Server*.

В последние годы большой интерес вызывает использование спецификации Общей Информационной Модели (*Common Information Model, CIM*) в приложениях для электроэнергетики. *CIM* — открытый стандарт, определяющий представление элементов электрической сети в виде совокупности объектов, их отношений и атрибутов, предназначенный обеспечить унифицированный способ обмена информацией об этих объектах. В упрощенном виде *CIM* можно представить как способ, позволяющий нескольким участникам обмениваться информацией, необходимой для управления их элементами. Упрощение заключается в том, что *CIM* не только определяет представление управляемых элементов и управляющей информации, но и предоставляет возможность управлять ими и контролировать их работу. Управляющее программное обеспечение, созданное с использованием *CIM*, может работать с множеством реализаций этого стандарта без потери данных или сложных перекодировок. Эта модель ориентирована на ее использование диспетчерскими службами предприятий и оперирует коммутационным представлением схемы и операций в ней. Все перспективные стандарты МЭК (61850, 61968, 61970) опираются на *CIM*.

Наиболее удобной формой представления *CIM*-модели является создание документов в формате *XML*. *XML* (*Xtensible Markup Language*) — язык разметки, фактически представляющий собой свод общих синтаксических правил. *XML* — текстовый формат, предназначенный для хранения структурированных данных (взамен существующих файлов баз данных), для обмена информацией между программами, а также для создания на его основе более специализированных языков разметки (например, *XHTML*), иногда называемых словарями.

На сегодняшний день актуальным вопросом является реализация поддержки в *SCADA*-системах возможности обмена описанием электрической сети в формате *CIM XML*. Поставленная задача заключалась в разработке программного обеспечения, осуществляющего выгрузку информации об элементах электрической сети. А также создание демонстрационного материала, содержащего описание специально разработанной базы данных в формате *CIM XML*.

Используется язык программирования *C++*, среда разработки *Microsoft Visual Studio 10.0*. Работа проводилась под управлением операционной системы *Windows 7*.

В базу данных сервера была добавлена таблица *CIMTypeTable*. Каждая из занесенных характеристик, такие как, напряжение на обмотках, на линии, сами объекты, имеют свой индекс и сходное название этой характеристике в терминах *CIM* стандарта. Таблица *CIMTypeTable* соединяется с другими таблицами в базе данных по индексу соответствующего *CIM* объекта. Данная таблица служит для описания объектов классов *CIM* и характерных им параметров. Ниже приведен класс, осуществляющий работу по заполнению объектов сервера из базы. Данные проходят проверку на исключение дубликатов, затем заполняют соответствующие поля объекта.

```
class ZrCimTypeBox
{
public:
```

```

    ZrCimTypeBox(void);
    virtual ~ZrCimTypeBox(void);
private:
    std::vector<ZrCimType*> CimTypeList;
    int CimTypeListSize;
    int CimTypeListLimit;
    PwLocalProtector CimTypeBoxProtector;
public:
    BOOL IsSimTypeIndexExist(long CimTypeIndex);
    void InsertSimTypeList(long CimTypeIndex, MwString CimTypeName);
    MwString GetSimTypeName(long CimTypeIndex);
};

```

Для реализации поставленной задачи произведено разбиение информации, взятой из структур памяти, на отдельные классы, соответствующие классам *CIM*, и имеющие набор параметров, аналогичных стандарту *CIM*.

class WindingClass — класс, реализующий формирование *CIM* представления об обмотках трансформатора. Объект имеет такие параметры:

- *IdWinding* — идентификатор обмотки;
- *UnWinding* — напряжение на обмотке;
- *BaseVoltageUn* — атрибут, ссылка на базовый уровень напряжения трансформаторной ветви;
- *MembOfEquipContainer* — атрибут, ссылка на подстанцию;

Ниже приведен программный код, реализующий заполнение объектов класса *WindingClass*.

```

...
WindingClass* WindingObj = new WindingClass;
data.Format("TransformerWinding_%ld_%s", UnObj->ValueUn, IdObject);
WindingObj->IdWinding = data;
data.Format("%s:VOLT[%ldkV]", AreaBox.GetObjectName(idArea,idObject), UnObj->ValueUn);
WindingObj->NameWinding = data;
WindingObj->UnWinding = UnObj->ValueUn;
data.Format("#_%ld", UnObj->ValueUn);
WindingObj->BaseVoltageUn = data;
...

```

class VoltageLevelClass — в объекты этого класса заносятся данные о напряжениях. Сформирован набор таких параметров:

- *IdVoltageLevel* — идентификатор объекта уровня напряжения;
- *NameVoltageLevel* — имя;
- *BaseVoltage* — атрибут для определения номинального напряжения;
- *Description* — соответствующее имя в *CIM* классификации;
- *HighVoltage* — высший уровень напряжения;
- *LowVoltage* — низший уровень напряжения.

Также разработаны другие классы, сформированные по принципу построения *CIM* моделей энергообъектов:

```

class BayClass;
class BaseVoltageClass;
class BaseObjectsClass;
class ConnectivityNodeClass;
class TerminalClass.

```

При запуске приложения, осуществляется загрузка данных в специально разработанные структуры для хранения объектов.

При формировании *CIM* происходит обращение к вектору областей данных, осуществляется перебор объектов. При получении объекта определяется его класс и тип, затем - его тип в *CIM*-модели. Используется конструкция *switch..case*. Далее для каждого объекта при помощи функций класса, отбираются и формируются нужные параметры, структура *CIM XML* представления. По этим параметрам производится связка с другими классами для получения полной топологии. Анализируется связь объекта в топологии энергосети с другими.

В результате проделанной работы был программно сгенерирован демонстрационный материал в формате *CIM XML*. Часть полученного материала представлена ниже:

```
<?xml version="1.0" encoding="Windows-1251"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:cim="http://iec.ch/TC57/2006/CIM-schema-cim10#">
  <cim:Substation rdf:ID="_Substation_19">
    <cim:IdentifiedObject.aliasName> ПС </cim:IdentifiedObject.aliasName>
    <cim:IdentifiedObject.name> ПС </cim:IdentifiedObject.name>
  </cim:Substation>
  <cim:ConnectivityNode rdf:ID="_CN_1">
    <cim:IdentifiedObject.aliasName>1</cim:IdentifiedObject.aliasName>
    <cim:IdentifiedObject.description>ConnectivityNode</cim:IdentifiedObject.description>
    <cim:ConnectivityNode.MemberOf_EquipmentContainer rdf:resource="#_Bay_35"/>
    <cim:IdentifiedObject.name>1</cim:IdentifiedObject.name>
  </cim:ConnectivityNode>
  <cim:TransformerWinding rdf:ID="TransformerWinding_35_PowerTransformer_1_10">
    <cim:IdentifiedObject.aliasName>трансф 1:VOLT[35kB]</cim:IdentifiedObject.aliasName>
    <cim:ConductingEquipment.BaseVoltage rdf:resource="#_35"/>
    <cim:Equipment.MemberOf_EquipmentContainer rdf:resource="#_Substation_19"/>
    <cim:IdentifiedObject.name>трансф 1:VOLT[35kB]</cim:IdentifiedObject.name>
  </cim:TransformerWinding>
  ...
  <cim:Bay rdf:ID="_Bay_110">
    <cim:IdentifiedObject.aliasName>PowerTransformer_1_10</cim:IdentifiedObject.aliasName>
    <cim:IdentifiedObject.name>PowerTransformer_1_10</cim:IdentifiedObject.name>
  </cim:Bay>
</rdf:RDF>
```

Полученный материал был протестирован свободно распространяемой программой *CIMSpy2.2*. Согласно результатам проверки, базовые принципы формирования *CIM XML* документа были соблюдены.

Библиографический список

1. Шилдт, Г. Самоучитель С++ Г. Шилдт — СПб. : BHV, 1997 — 512 с.
2. Лафоре, Р. Объектно-ориентированное программирование в С++ / Р. Лафоре — СПб. : Питер, 2006 — 928 с.
3. Шеперд, Д. Освой самостоятельно XML за 21 день, 2-е издание / Д. Шеперд — М. : Издательский дом «Вильямс», 2002. — 432 с.
4. Мандрик А. В. «Объектный подход к обработке телемеханической информации» / А. В. Мандрик , М. В. Ухова — V Международная научно-практическая конференция — Протвино : 2011 — с.125-127.
5. http://ru.wikipedia.org/wiki/Common_Information_Model *Common_Information_Model*
6. *Regimov.net*, статья " Введение в МЭК 61970-301 & 61968-11: *Common Information Model* (Общая Информационная Модель) ", http://regimov.net/content/96-im11#Common_Information_Model_.