

# ТОРАЗ NM

## (Модель сети)

643.17480174.00001-01 31-12

ОПИСАНИЕ ПРОГРАММЫ

<i>Инв. №</i>	<i>Подпись и дата</i>	<i>Взам. инв.</i>	<i>Инв. №</i>	<i>Подпись и дата</i>

Москва 2023

## ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

ОС – операционная система

NM – Network Model (модель сети)

CIM – Common Information Model

## АННОТАЦИЯ

В данном программном документе приведено описание приложения TOPAZ NM (Модель сети). Документ содержит сведения о логической структуре и функционировании данного приложения.

## СОДЕРЖАНИЕ

1.	ОБЩИЕ СВЕДЕНИЯ .....	5
2.	ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ .....	5
3.	ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ.....	5
3.1.	Алгоритмы программы .....	5
3.2.	Используемые методы .....	7
3.3.	Структура программы с описанием функций составных частей и связи между ними .....	8
3.4.	Связи программы с другими программами .....	8
4.	ИСПОЛЬЗУЕМЫ ТЕХНИЧЕСКИЕ СРЕДСТВА.....	8
5.	ВЫЗОВ И ЗАГРУЗКА.....	9
5.1.	Способ вызова программы .....	9
5.2.	Входные точки в программу .....	9
6.	ВХОДНЫЕ ДАННЫЕ .....	9
6.1.	Характер, организация и предварительная подготовка входных данных .....	9
7.	ВЫХОДНЫЕ ДАННЫЕ .....	9
7.1.	Характер, организация и предварительная подготовка выходных данных.....	9

## 1. ОБЩИЕ СВЕДЕНИЯ

### 1.1. Обозначение и наименование программы

Наименование программы – ТОРАЗ NM (Модель сети).

### 1.2. Программное обеспечение, необходимое для функционирования программы

Системные программные средства, используемые программой – операционные системы реального времени на основе ядра LINUX.

### 1.3. Языки программирования, на которых написана программа

Программа написана на языке C++.

## 2. ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ

Программа представляет собой объектно-ориентированную CIM-совместимую модель электрической сети (далее – объектная модель), обеспечивающая предоставление производных данных, необходимых для моделирования топологии и состояния сети на основе паспортных данных энергетического и автоматизированного оборудования.

## 3. ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ

### 3.1. Алгоритмы программы

Алгоритм схематично изображен на рисунках ниже.



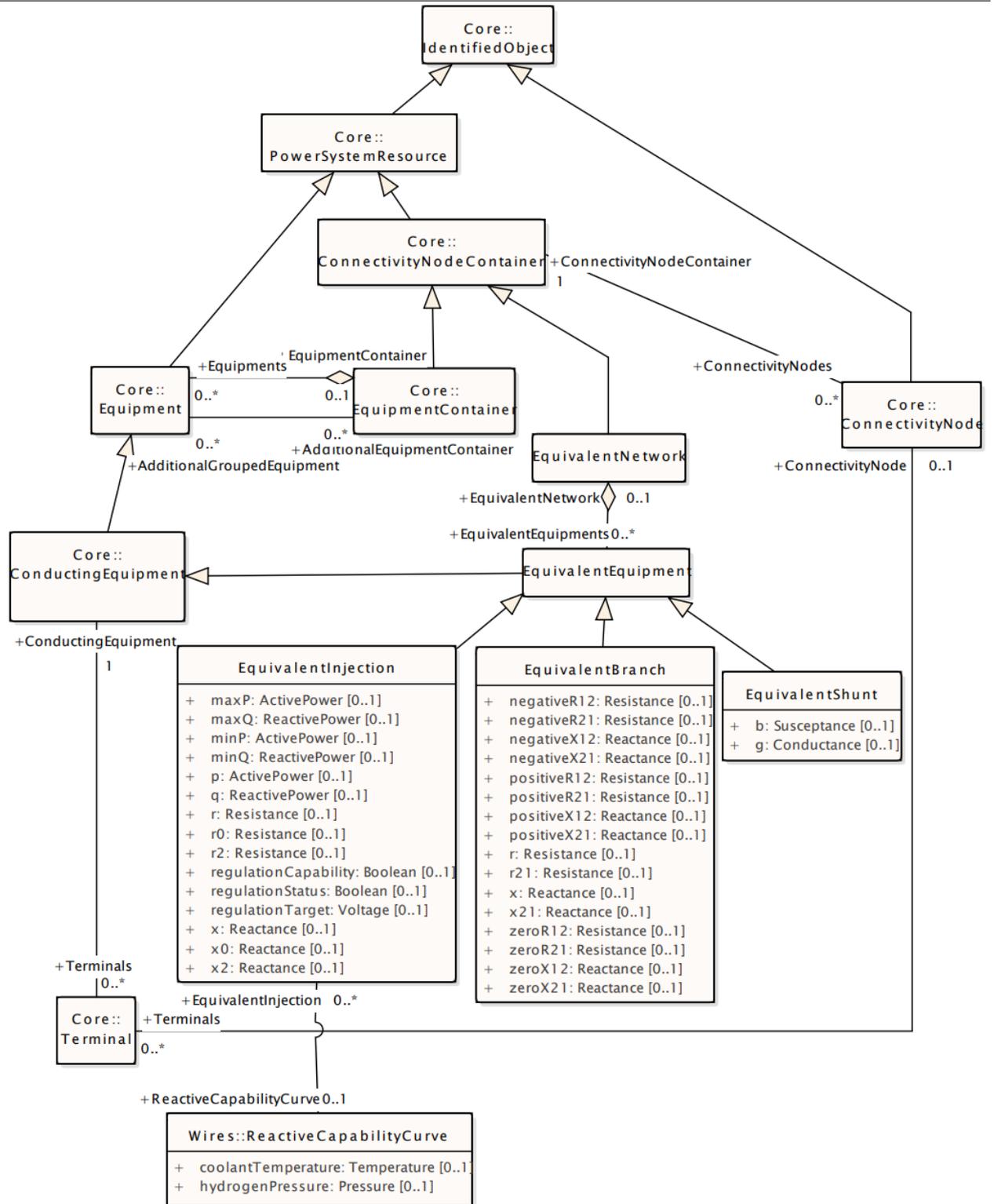


Рис. 2

### 3.2. Используемые методы

- *MTModel::MTModel()* : конструктор основного класса библиотеки объектной модели

- *bool MTModel::SaveHeadModel( const std::string &fileName, clbFcnSave callBackFunc) :* сохранение шапки модели в файл. Принимает в качестве аргументов имя файла и указатель на функцию логирования;
- *bool MTModel::SaveBodyObjectModel( const std::string &fileName, clbFcnSave callBackFunc) :* сохранение тела объектной модели в файл. Принимает в качестве аргументов имя файла и указатель на функцию логирования;
- *bool MTModel::LoadHeadModel( const std::string &fileName, clbFcnLoad callBackFunc) :* загрузка шапки модели из файла. Принимает в качестве аргументов имя файла и указатель на функцию логирования;
- *bool MTModel::LoadBodyObjectModel( const std::string &fileName, MTOBJECTITEM \*objectItem, clbFcnLoad callBackFunc) :* загрузка тела объектной модели из файла. Принимает в качестве аргументов имя файла, указатель на класс подстанции и указатель на функцию логирования;
- *bool MTModel::Check( clbFcnLoad callBackFunc) const:* проверка модели с записью ошибок и предупреждений в логгер; Принимает в качестве аргумента указатель на функцию логирования;
- *bool CIMXML::ExportToSIMXML(QString &cimXmlFileName) :* сохраняет объектную модель в файл в формате CIM-XML;
- *bool CIMXML::ExportValuesToSIMXML(QString &cimXmlFileName) :* сохраняет значения CIM-атрибутов в файл в формате CIM-XML.

### 3.3. Структура программы с описанием функций составных частей и связи между ними

Различные методы класса MTModel вызываются из обработчиков событий, привязанных к различным элементам GUI Model Creator.

*void MainWindow::onExportToCimXml():* точка вызова метода сохранения объектной модели в формат CIM-XML. Вызывает функцию *bool CIMXML::ExportToSIMXML*. Использует указатель на класс MTModel.

*void MainWindow::onExportValuesToCimXml():* точка вызова метода сохранения значения CIM-атрибутов в формат CIM-XML. Вызывает функцию *CIMXML::ExportValuesToSIMXML*.

### 3.4. Связи программы с другими программами

В программе используются элементы программы TOPAZ Model Creator.

## 4. ИСПОЛЬЗУЕМЫЕ ТЕХНИЧЕСКИЕ СРЕДСТВА

Приложение поддерживает аппаратные средства, основанные на АРМ-архитектуре. Необходимые требования к аппаратной части пакет приложений представлены в таблицах ниже.

**Таблица 1 – Требования к серверу доступа к данным**

<b>Наименование параметра</b>	<b>Значение</b>
ЦП	Не менее 4-х ядер, не менее 1,2 ГГц
ОЗУ	Не менее 4 Гб
ПЗУ (системный накопитель)	SSD, не менее 8 Гб
ПЗУ (накопитель БД)	4 x 2.5-inch HDD, емкость носителя определяется количеством узлов в сети и глубиной архивирования, RAID10
ОС	Операционные системы реального времени на основе ядра LINUX
Ethernet	Не менее 2 шт, 1 Гбит/с

**Таблица 2 – Требования к АРМ пользователя**

<b>Наименование параметра</b>	<b>Значение</b>
ЦП	Не менее 4-х ядер, не менее 1,2 ГГц
ОЗУ	Не менее 4 Гб
Видеокарта	Дискретная
ПЗУ (системный накопитель)	SSD, не менее 8 Гб
Диагональ монитора, не менее	27"
Ethernet	Не менее 2 шт, 1 Гбит/с

## 5. ВЫЗОВ И ЗАГРУЗКА

### 5.1. Способ вызова программы

Программа работает в автоматическом режиме. Запуск осуществляется после автозагрузки программы в ОС.

### 5.2. Входные точки в программу

- Класс MTModel;
- События onExportToSIMXML;
- onExportValuesToCimXml

## 6. ВХОДНЫЕ ДАННЫЕ

### 6.1. Характер, организация и предварительная подготовка входных данных

Входными являются пользовательские данные (текстовая или цифровая информация). Входные данные хранятся в формате XML.

## 7. ВЫХОДНЫЕ ДАННЫЕ

### 7.1. Характер, организация и предварительная подготовка выходных данных

Входные данные хранятся в БД системы SCADA, представляют собой XML-файл.