

Н.Ю. Кульман, Е.И. Пугачева

РЕАЛИЗАЦИЯ ВЕКТОРНЫХ ДИАГРАММ ДЛЯ СИСТЕМЫ КОМПЛЕКСНОГО УЧЕТА РЕСУРСОВ ЖКХ в SCADA «СИСТЕЛ»

Филиал «Протвино» университета «Дубна»
Кафедра информационных технологий

Целью данной работы являлась реализация векторных диаграмм для системы комплексного учета ресурсов ЖКХ в SCADA «СИСТЕЛ» [1]. Работа выполнена на базе предприятия ООО «Системы телемеханики и автоматизации» (г. Протвино), созданного в 1992 году с целью внедрения современных информационных технологий в различные отрасли промышленности.

Необходимым условием повышения эффективности использования ресурсов в области промышленной автоматизации является внедрение системы комплексного учета, позволяющей получать в реальном времени информацию о потреблении ресурсов основными производственными участками. Наличие такой информации позволяет проводить оперативную оценку эффективности производства, и своевременно принимать необходимые управленческие решения по оптимизации потребления того или иного носителя. Система позволяет минимизировать расходы за счет внедрения современных технологий автоматизации процессов сбора, обработки, анализа и представления актуальной информации об объектах.

Важнейшими задачами системы комплексного учета ресурсов являются:

- технический и коммерческий учет электроэнергии на объектах ЖКХ и промышленных предприятиях;
- формирование баланса потребления электроэнергии, тепла, воды и газа;
- снижение потерь при транспортировке энергоносителей (по проводам, трубам и т.п.);
- представление данных для их анализа, составление отчетов;
- активное управление ресурсами предприятий – потребителей;
- ведение количественного учета любых ресурсов, которые могут быть измерены в виде набора количественных параметров.

Система комплексного учёта ресурсов должна отвечать выше перечисленным требованиям, а также обладать удобным пользовательским интерфейсом, являющимся одним из важнейших факторов, касающихся удобства работы и во многом определяющим спрос и практичность продукта. Интерфейс системы должен представлять команды как организованный набор наиболее существенных функций, что делает его мощным средством для повышения динамизма в работе различных служб и его адаптации к задачам пользователя.

Одним из главных элементов интерфейса является способ представления информации. В данной системе можно создавать два типа представления: для вывода данных в табличных формах, разработанного с помощью специального класса, и для графического отображения диаграмм, схем, которое было реализовано с помощью библиотеки *GDI+* [2], призванной заменить существующий интерфейс *GDI* [3]. *GDI+* сочетает в себе все достоинства своего предшественника и предоставляет множество новых мощных возможностей, таких как: градиентная закрашка; поддержка прозрачности; режимы улучшения изображения; сплайны; пути; координатные преобразования и многое другое.

В системе комплексного учета ресурсов графическое представление необходимо для вывода векторных диаграмм [4], которые являются наиболее простым и наглядным методом оценки и проверки правильности сборки цепей тока и напряжения устройств. Анализ векторных диаграмм позволяет убедиться в правильности включения приборов учета электроэнергии, проверить настройки оборудования, направление подключенного прибора, определить качество сети и электроэнергию, подаваемую потребителю.

Реализация векторных диаграмм состоит из нескольких этапов.

Во-первых, пользователь должен указать необходимую точку учёта и выбрать время, которое его интересует, для этого была разработана панель «Векторные диаграммы», которая содержит поле для ввода даты, времени и кнопку, отправляющую запрос серверу. В функциональном окне «Сбор данных» пользователь выбирает точку учёта, на ленточной панели указывает дату, время и отправляет запрос (рис. 1).

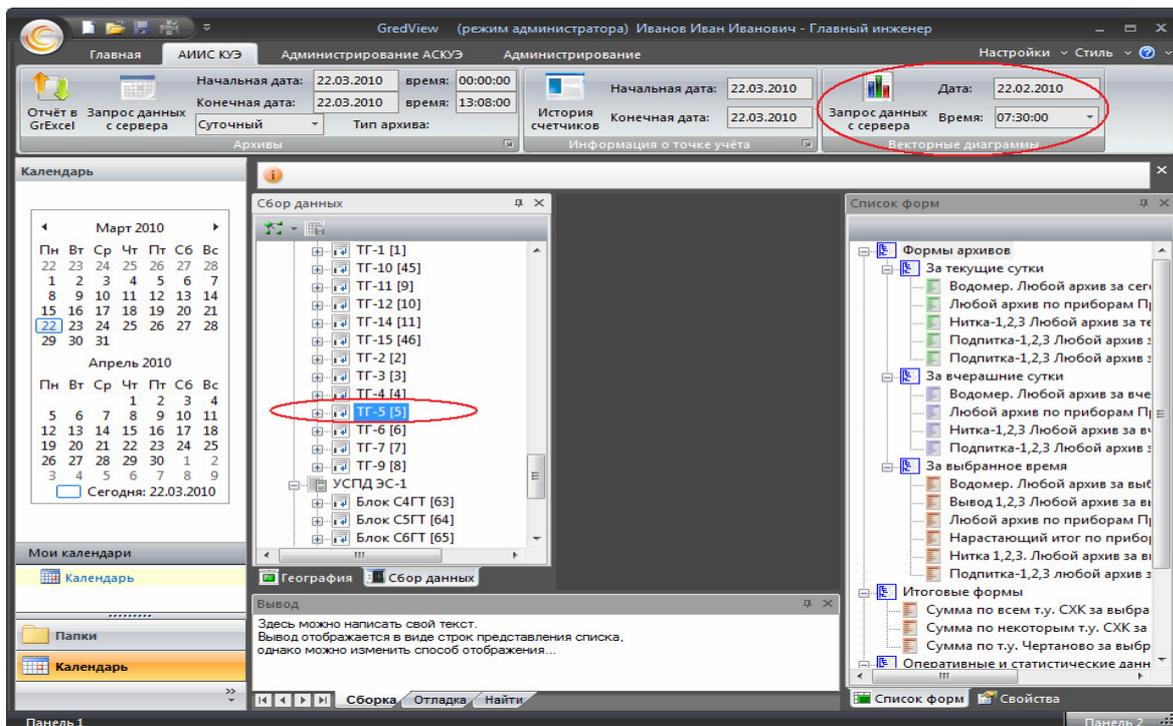


Рис.1 Запрос данных

Далее полученные от сервера данные в виде XML-пакета [5] разбираются. В этом пакете содержатся значения активной, реактивной мощности и значения напряжений по трём фазам. Для вывода самой диаграммы нужно знать значения напряжений, токов и угла отклонения по трем фазам, поэтому происходит расчет по следующим формулам:

$$I_A = \sqrt{\left(\frac{P_A}{U_A}\right)^2 + \left(\frac{Q_A}{U_A}\right)^2} \quad \text{– ток в фазе } A;$$

$$\operatorname{tg} \varphi_A = \frac{Q_A}{P_A} \quad \text{– тангенс угла между током и напряжением в фазе } A;$$

здесь φ_A – фаза (угол), U_A - напряжение фазы A , P_A - активная мощность фазы A , Q_A - реактивная мощность фазы A .

Для вывода векторной диаграммы по одной точке учёта используется режим преобразования координат с переменным масштабом, поэтому при увеличении размера клиентского окна, например, при закрытии каких-либо функциональных окон, изображение с векторной диаграммой тоже увеличивается (рис.2).

Кроме этого, для вывода диаграммы необходимо провести соответствие между полученными по формулам значениями и координатами рисования, а именно, круговая диаграмма строится из трех окружностей, из которых радиус средней соответствует номинальному напряжению в 220 В, а остальные две соответствуют нормально и предельно допустимым значениям отклонения напряжения: $U_{\text{нор}} = \pm 5\%$ и $U_{\text{пред}} = \pm 10\%$. Вектор, соответствующий номинальному напряжению, служит основой для построения векторов со значениями, полученными от сервера. Для этого строится пропорция, включающая нормальное напряжение и вектор в логических единицах. Такое преобразование проводится для напряжений всех трех фаз.

Похожие операции проводятся и для токов, но есть небольшая разница. Из за того, что отрезки, соответствующие напряжениям и токам строятся на одной диаграмме, а измеряются в разных единицах, было принято соотносить значения тока по фазе A с радиусом малой окружности, т.е. вычисленное значение тока по фазе A всегда будет равным вектору, проведенному из центра до малой окружности, а логические значения остальных двух токов находятся уже исходя из этой пропорции.

Вектора напряжений повернуты относительно друг друга на 120° , а вектора токов строятся относительно соответствующих векторов напряжений с учетом углов отклонения, вычисленных по вышеуказанной формуле.

Поворот векторов был реализован с помощью матрицы трансформации - специальной математической структуры, которая обеспечивает простое математическое преобразование из одной системы в другую. По умолчанию центр трансформации имеет координаты 0.0. Для векторной диаграммы центр трансформации по

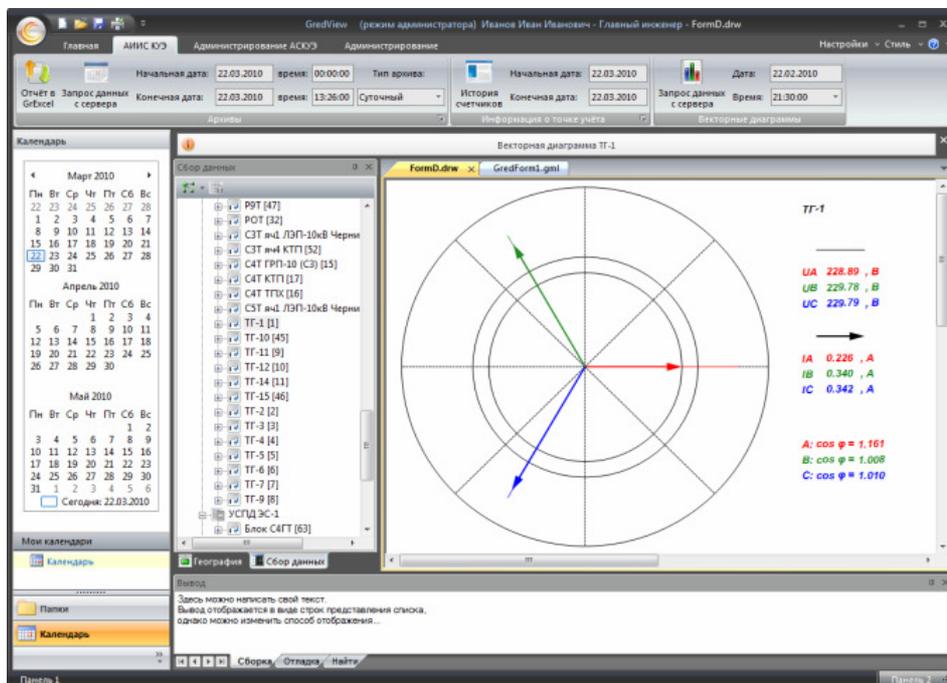
оси абсцисс равен начальной координате поворачиваемой фигуры, а по оси ординат равен центру поворачиваемой фигуры.

Справа от диаграммы находится таблица значений, в которой указываются имя выбранной точки учёта, напряжения, токи и углы отклонения по трем фазам, которые для наглядности подсвечиваются разными цветами.

После получения и преобразования всех необходимых значений происходит открытие самого документа, при этом указывается имя шаблона соответствующего типа, проверяется, не открыт ли уже документ с таким именем, и если открыт, то он закрывается и открывается заново, но с уже новыми данными.

В данной работе также был разработан шаблон для вывода векторных диаграмм сразу по нескольким точкам учёта (рис. 3), реализованный в режиме преобразования координат с постоянным масштабом. Такой набор диаграмм позволяет быстро проводить оценку работы нескольких устройств сразу.

В результате выполнения данной работы был разработан интерфейс для построения векторных диаграмм в системе комплексного учета ресурсов SCADA «СИСТЕЛ», отвечающий поставленным требованиям. В настоящее время это приложение находится в опытной эксплуатации на РТС (районная тепловая станция) «Чертаново» (г.Москва) и на СХК (Сибирский химический комбинат) (г.Северск, Томской обл.)



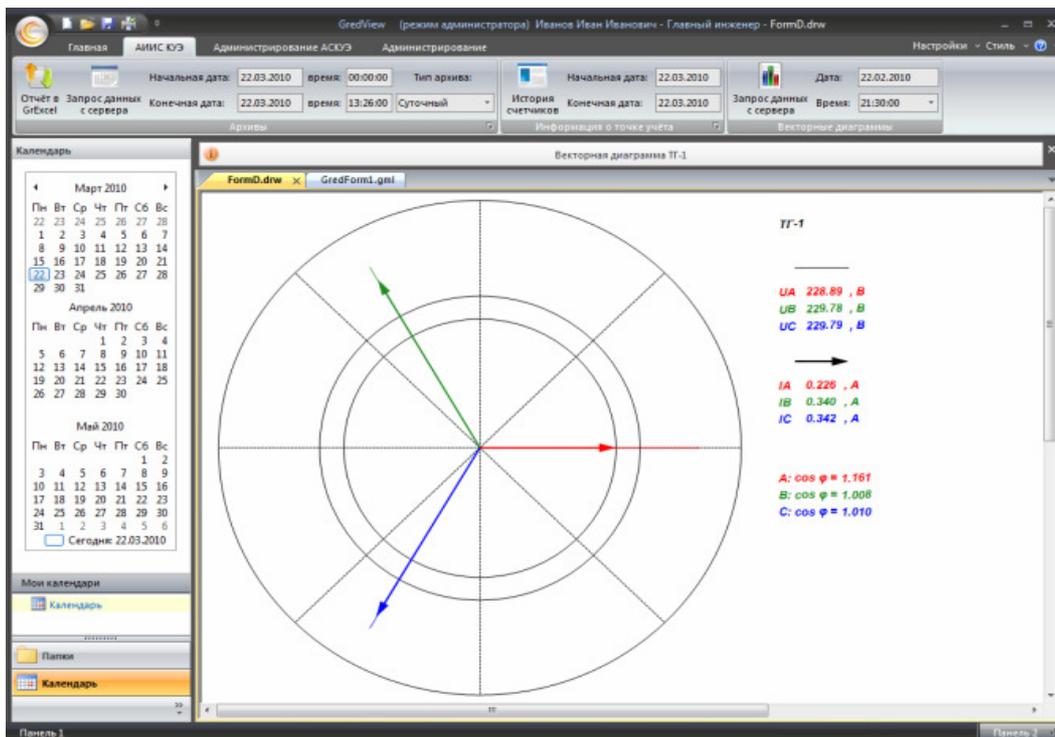


Рис.2 Режим преобразования координат с переменным масштабом

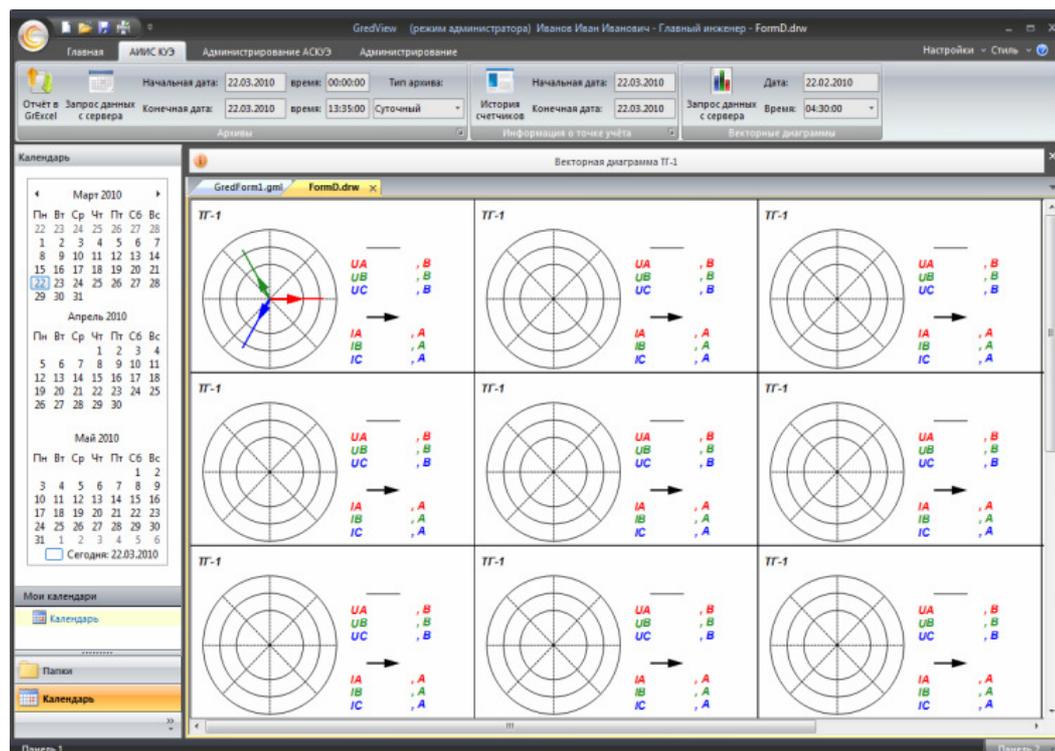


Рис.3 Режим преобразования координат с постоянным масштабом

Библиографический список

1. Рыкованов, С.Н. Оперативный информационный управляющий комплекс «Систел». / С.Н. Рыкованов, Н.Ю. Кульман, В.И. Ухов // Межотраслевой производственно-технический журнал «Автоматизация от А до Я». – 2007, №1. – С. 9-11.
2. Поляков, А. Программирование графики GDI+ и DirectX / А. Поляков, В. Брусенцев. – СПб. : БХВ - Петербург, 2005. – 368 с.
3. Арчер, Т. Microsoft Visual C++ .NET. Библия пользователя / Т. Арчер, Э. Уфйечепел. – М. : Вильямс, 2005. – 1216 с.

4. Кузнецов, Ф.Д. Векторные диаграммы в схемах релейной защиты и автоматики. Практическое пособие / Ф.Д. Кузнецов. – М. : Издательство НЦ ЭНАС, 2009. – 64 с.
5. Шефферд, Джордж. Программирование на Microsoft Visual C++.NET / Джордж Шефферд. – М. : Русская редакция, 2003. – 928 с.