УДК 519.7/6813

МЕТОДЫ РАЗРАБОТКИ ПРОЕКТНО-СИСТЕМНОЙ ДОКУМЕНТАЦИИ НА ОСНОВЕ ПЛАНОВО-УЧЕТНЫХ ЕДИНИЦ

© 2012 О.А. Верушкин

ФНПЦ ОАО "НПО "Марс", г. Ульяновск

Поступила в редакцию 29.06.2012

В статье произведен частичный обзор методов разработки проектно-системной документации на предприятии, изготавливающем автоматизированные системы управления, рассмотрена возможность использования системы документирования DocWIQA.net при разработке проектно-системной документации.
Ключевые слова: документация проектно-системная, схема структурная, система документирования DocWIQA.net, формирование документа, модель дискретно-детерминированная.

Существование любой промышленной продукции немыслимо без технической документации. Любые технические средства, программы или сложные технические системы окружены документами все время: от замысла до утилизации. Невозможно представить себе создание хоть сколько-нибудь сложного технического решения без проектной документации, а применение без эксплуатационной. Информативная, полная, понятная, качественная техническая документация - залог успеха на всех этапах его жизненного цикла. Вместе с тем, разработка технической документации требует специфических знаний и умений, а также значительных трудозатрат.

К проектно-системной документации на корабельные автоматизированные системы управления можно отнести следующие документы:

– схема электрическая структурная;

– технические условия (проект);

– таблица соединений.

Спецификой разработки проектно-системной документации являются жесткие требования ГОСТ, РД, СТП по оформлению, согласованию и сдаче в архив, что обуславливает сложность выбора и применения САПР.

Теперь более подробно рассмотрим процесс создания каждого из выше приведенных документов. Схема электрическая структурная - схема, определяющая основные функциональные части изделия, их назначение и взаимосвязи. Разрабатывается на стадии эскизного проектирования по техническому заданию и протоколам с внешними абонентами и необходима для общего ознакомления с изделием и дальнейшей проработки последующих документов. На данный период выполняется с использованием Microsoft Visio или Microsoft Word в виде условно графических обозначений и связей между ними. Пример выполнения части схемы электрической структурной приведен на рис. 1.

Технические условия (проект) - документ, содержащий требования (совокупность всех показателей, норм, правил и положений) к изделию, его изготовлению, контролю, приемке и поставке, которые нецелесообразно указывать в других конструкторских документах. В связи с тем, что технические условия представляют из себя текстовый документ, то наиболее рационально и целесообразно использовать для его разработки текстовый процессор, например: Microsoft Word. Текстовые процессоры имеют специальные функции, которые предназначены для облегчения ввода текста и представления его в напечатанном виде.

Таблица соединений - таблица, содержащая сведения о расположении технических средств, соединениях, местах подключения и другую информацию. Выпускается взамен схемы электрической соединений в виде самостоятельного документа. В связи с внедрением на предприятии системы документирования DocWIQA.NET, прорабатывается вопрос автоматизированной разработки части документации (в том числе и таблицы соединений) на изделия с использованием данной системы на основе вопросно-ответных протоколов.

Система документирования DocWIQA.NET позволяет осуществлять оперативную проверку проектно-системных решений с использованием типовых интерфейсных прототипов с возможностями выполнения внешней функциональности. Это позволяет:

- осуществить уточнение неясных требований;

- провести анализ осуществимости проекта и отдельных проектных решений;

- осуществить предварительное тестирование на ранних стадиях;

- продемонстрировать отдельные решения заказчикам и членам проектной группы;

- осуществить быструю генерацию шаблонных

Верушкин Олег Александрович, инженер.
E-mail: mars@mv.ru

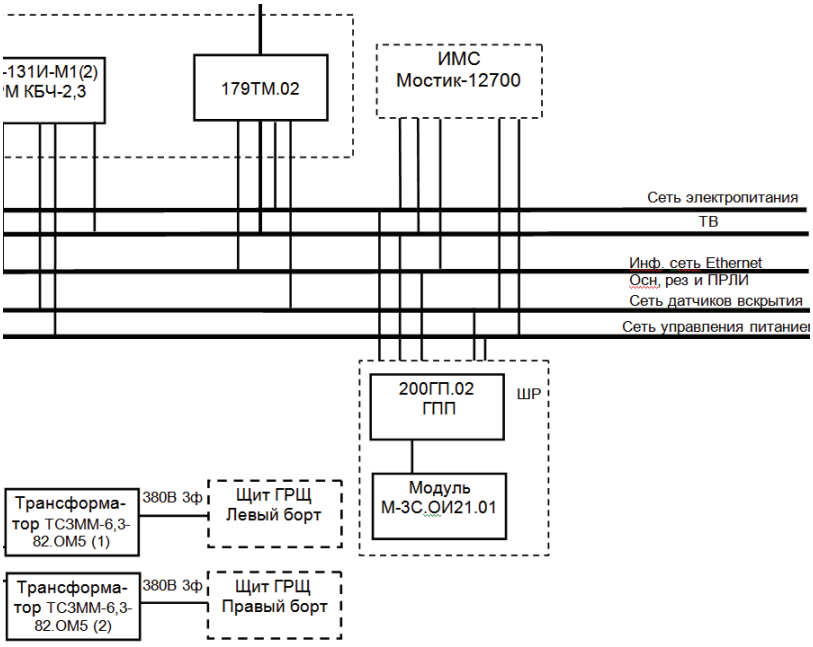


Рис. 1. Выполнение части схемы электрической структурной

решений, которые могут быть в дальнейшем использованы в основной программной реализации;

- помочь в осуществлении выбора одного или нескольких проектных решений.

Комплекс средств автоматизированного составления документов DocWIQA.NET состоит из следующих функциональных модулей:

- подсистема вопросно-ответных шаблонов;
- подсистема составления шаблонов оформления документов;
- подсистема формирования документов;
- подсистема контекстных характеристик (взаимоувязки документов, шаблонов и данных).

Система предоставляет возможности по формированию шаблонов данных документов, шаблонов оформления документов, редактированию реальных данных, автоматическому формированию документов и передачи их во внешние приложения или выводу на печать.

Процесс составления шаблонов и экземпляров документа состоит в следующем:

- формирование шаблонов данных документа (модуль QA - шаблоны) – 1 раз для каждого документа для проектной организации, на этом этапе формируются типовые вопросно-ответные структуры для каждого документа – по сути, определяется содержательная часть типового документа: разделы, главы, абзацы;

- формирование шаблона оформления документа – 1 раз для проектной организации – формирование внешнего вида документа, определение точек вставки содержательных частей в документ, их представление (шрифт, цвет, список и т.д.);

- перенесение шаблонов данных в вопросно-ответный протокол текущего реального проекта, внесение реальных данных – для каждого проектного решения. на этом этапе шаблоны содержательной части переносятся в область данных реального проекта (вопросно-ответный протокол), вместо шаблонных данных вносятся реальные данные конкретного проекта;

- формирование документа в плагине “проектные документы” (для каждого экземпляра документа);

- выбор задачи документирования в вопросно-ответном протоколе;

- выбор шаблона оформления;

- определение имени документа, его описания;

- генерация документа;

- просмотр документа, передача в систему документооборота, печать.

Связующим звеном между данными документа и его представлением выступают контекстные характеристики, которые приписываются содержательным частям документа и местам в шаблонах оформления, куда они должны быть вставлены.

Подсистема контекстной атрибутики предназначена для определения контекстов использования единиц вопросно-ответного протокола в средствах вопросно-ответного моделирования. Подсистема атрибутики A определяется входящими в нее атрибутами ai, представленными в древовидном виде $A=\{a_i\}$, i=1..n, $a_i=\{$name, symname, description, P, $a_k\}$, name – название атрибута, description - описание для пользовательского режима, symname – уникальное символьное имя для использования в автоматическом режиме, $P=\{p_i\}$, i=1..n – множество параметров атрибута, где $p_i=<$pname, ptype> pname – название параметра, ptype – тип параметра (целый, вещественный, строковый и т.д.), a_k – дочерние атрибуты, где k=1..n.

Каждый атрибут самостоятелен и может быть назначен любому объекту системы (вопросно-ответной единице, документу, шаблону, методике и т.д.). При назначении N определяется однозначное соответствие атрибута a_i объекту o_j: $N=<a_i, o_j, value, V>$, где value – значение, приписываемое этому назначению, $V=\{v_g\}$, $v_g=<p_i, pvalue>$, pvalue = значение параметра для созданного назначения.

Подсистема атрибутики используется в двух режимах:

- пользовательский;
- автоматизированный.

В пользовательском режиме атрибуты позволяют члену проектной группы определить, в каком контексте используется объект, что он обозначает. Например, каждой вопросно-ответной единице приписывается множество стандартных атрибутов: время создания, автор, тип (задача, вопрос, ответ, мотив и т.д.), статус. При этом данная единица может использоваться в качестве шага методики, элемента документа, переменной в вопросно-ответной программе и т.д. Чтобы определить контекст использования единицы, ей может быть приписан соответствующий атрибут в пользовательском, "ручном" режиме.

В автоматизированном режиме модули и плагины вопросно-ответного процессора WIQA.NET определяют назначения атрибутов программным способом и используют эти назначения в своей работе.

В автоматизированном режиме атрибутика может быть использована для следующих целей:

- хранения служебной информации для плагинов, например: тип переменной, название которой хранится в тексте вопросно-ответной единицы; адрес вопросно-ответной единицы (идентификатор в базе данных), на которую ссылается QA-единица, которой приписан данный атрибут; с помощью этого варианта использования в QA-дереве становится возможным хранить графовые структуры;
- в системе документирования с помощью атрибутов могут быть помечены те вопросно-ответные единицы, которые могут быть использованы в качестве данных для вставки в шаблон документа;
- в подсистеме прецедентов атрибуты указывают те вопросно-ответные единицы, которые могут быть использованы в качестве входных данных для выбора прецедентом места его использования;
- атрибутика также может быть использована для реализации средств объектно - вопросно-ответных преобразований;

Подсистема контекстной атрибутики предназначена для реализации средств объектно – вопросно-ответного преобразования (OQAM, Object-Question-Answer Mapping). Под механизмами OQAM понимаются схожие механизмы с объектно-реляционным преобразованием (ORM, Object-Relational Mapping), реализованным в технологиях LINQ To SQL, Entity Framework, Hibernate, NHibernate: автоматическое преобразование сущностей базы данных в объекты языка программирования (C#, Java) и генерация описания классов этих объектов на основе схемы базы данных.

В случае OQAM реализуются схожие механизмы, но применительно не к базам данных, а к средствам вопросно-ответного моделирования:

- экземпляры сущностей инкапсулированы во фрагментах вопросно-ответного протокола;
- необходима полуавтоматическая, ручная генерация описания классов, схемы данных.

Объектно-вопросно-ответные преобразования в самом общем виде предназначены для облегчения программного доступа к данным вопросно-ответного протокола, расширения семантики отдельных экземпляров объектов плагинов, в т.ч. обеспечение взаимодействия отдельных объектов путем установления связей между ними. Эти возможности могут быть успешно применены при реализации средств вопросно-ответного программирования, документирования, работы с метрическими показателями, ссылками на ресурсы, прецедентами и др.

Главным преимуществом средств OQAM по сравнению с аналогичными средствами в NetWIQA (Delphi) является возможность для конечного пользователя (а не системного программиста) определять новые классы в модели предметной области, изменять существующие, создавать экземпляры этих классов из разных источников, управлять ими, создавать собственную логику работы с ними.

Например: создадим класс "Документ", который содержит обязательные свойства "Название", "Автор", "Дата составления", "Уровень секретности". От этого класса может наследоваться другой класс "Техническое задание", который помимо свойств базового класса может содержать новые свойства: "Введение", "Основание для разработки", "Назначение разработки", "Требования к программе" и т.д. в соответствии с ЕСКД. Экземпляр класса "Техническое задание" – это конкретный заполненный документ. Причем данные для заполнения хранятся в вопросно-ответном протоколе, а описание классов – в подсистеме контекстной атрибутики. Пользователь в визуальном режиме определяет внешний вид документа, определяя места документа для включения в них свойств элементов. Сама система осуществляет автоматическую генерацию документа на основании определенных экземпляров документов, описании представления.

В подсистеме контекстных атрибутов реализованы средства для распределения атрибутов по приложениям. Т.е. каждому атрибуту должны быть поставлены в соответствие приложения (плагины), которые его используют. Также должны быть средства для контроля сущностей на уровне базы данных, которым могут быть приписаны дополнительные атрибуты.

Конкретный атрибут может быть назначен произвольной таблице (сущности) в базе данных, для этого для каждой такой таблицы динамически создается новая таблица для хранения значений атрибутов и значений их свойств, хранимые процедуры для доступа и контроля назначений.

Все описанное окружение может быть удалено с использованием визуальных средств.

В настоящее время доступны следующие возможности:

- древовидная структура атрибутов, пользовательский визуальный контроль для создания, назначения атрибутов;
- функционал по назначению, выборке, изменению атрибутов отдельным сущностям программы (необходимы визуальные средства);
- пользовательский контроль по назначению атрибутов qa-единицам и просмотра списка уже назначенных атрибутов;
- пользовательский контроль по назначению атрибутов qa-шаблонам и просмотра списка уже назначенных атрибутов;
- средства импорта/экспорта атрибутов в xml – файл;
- средства поиска сущностей, которым назначен определенный атрибут.

На данный момент определен перечень входных и выходных документов и их атрибутов (рис. 2).

Особенности математического аппарата при дискретно-детерминированном подходе разработки модели процесса формирования проектно-системной документации рассмотрим на примере использования теории автоматов. Теория автоматов — это раздел теоретической кибернетики, в котором изучаются математические модели — автоматы. На основе этой теории система документирования DocWIQA.NET представляется в виде абстрактно конечного автомата, перерабатывающего дискретную информацию и меняющего свои внутренние состояния лишь в допустимые моменты времени.

Представим автомат как некоторое устройство (черный ящик), на которое подаются входные сигналы и снимаются выходные и которое может иметь некоторые внутренние состояния. Конечным автоматом называется автомат, у которого множество внутренних состояний и входных сигналов (а, следовательно, и множество выходных сигналов) являются конечными множествами.

Абстрактно конечный автомат (англ. finite automata) можно представить как математическую схему (F-схему), характеризующуюся шестью элементами: конечным множеством X входных сигналов (входным алфавитом, к которому можно отнести следующие данные – перечень приборов, внешних абонентов, кабелей с соответствующими атрибутами); конечным множеством Y выходных сигналов (выходным алфавитом – это получаемая на выходе проектно-системная документация); конечным множеством Z внутренних состояний (внутренним алфавитом или алфавитом состояний нашей системы); начальным состоянием z_0, $z_0 \in Z$; функцией переходов состояний $\varphi(z, x)$; функцией выходов $\psi(z, x)$. Автомат, задаваемый F-схемой: $F=<Z, X, Y, \varphi, \psi, z_0>$,— функционирует в дискретном автоматном времени, моментами которого являются такты, т. е. примыкающие друг к другу интервалы времени, каждому из которых соответствуют постоянные значения входного и выходного сигна-

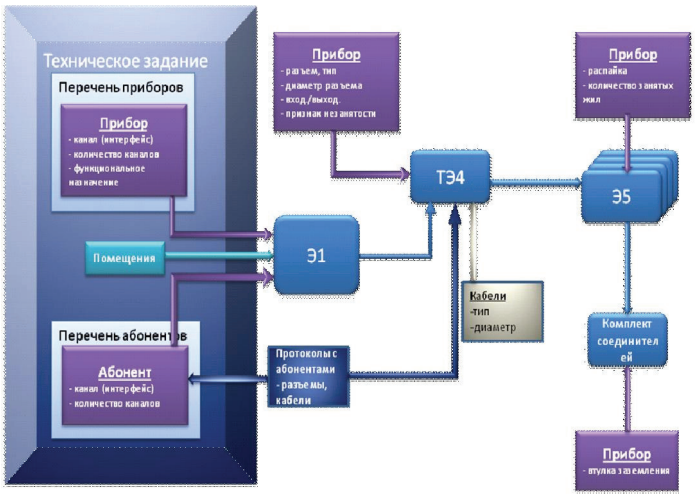


Рис. 2. Модель последовательности формирования документации

лов и внутренние состояния. Обозначим состояние, а также входной и выходной сигналы, соответствующие t-му такту при t=0, 1, 2, ..., через z(t), x(t), y(t). При этом, по условию, $z(0)=z_0$, а $z(t) \in Z$, $x(t) \in X$, $y(t) \in Y$.

Абстрактный конечный автомат имеет один входной и один выходной каналы. В каждый момент t=0, 1, 2, ... дискретного времени F-автомат находится в определенном состоянии z(t) из множества Z состояний автомата, причем в начальный момент времени t=0 он всегда находится в начальном состоянии z(0)=z0. В момент t, будучи в состоянии z(t), автомат способен воспринять на входном канале сигнал $x(t) \in X$ и выдать на выходном канале сигнал $y(t) = \psi [z(t), x(t)]$, переходя в состояние $z(t+1) = \varphi [z(t), x(t)]$, $z(t) \in Z$, $y(t) \in Y$. Абстрактный конечный автомат реализует некоторое отображение множества слов входного алфавита X на множество слов выходного алфавита Y. Другими словами, если на вход конечного автомата, установленного в начальное состояние z0, подавать в некоторой последовательности буквы входного алфавита x(0), x(1), x(2),..., т. е. входное слово, то на выходе автомата будут последовательно появляться буквы выходного алфавита y(0), y(1), y(2), ..., образуя выходное слово.

Таким образом, работа конечного автомата происходит по следующей схеме: в каждом t-м такте на вход автомата, находящегося в состоянии z(t), подается некоторый сигнал x(t), на который он реагирует переходом в (t+1)-м такте в новое состояние z(t + 1) и выдачей некоторого выходного сигнала. Сказанное выше можно описать следующими уравнениями: для F-автомата первого рода, называемого также автоматом Мили,

$$z(t+1) = \varphi [z(t), x(t)], t = 0, 1, 2, ...; \quad (1)$$

$$y(t) = \psi [z(t), x(t)], t = 0, 1,2, ...; \quad (2)$$

Таким образом, уравнения (1), (2) полностью задают F-автомат детерминированной системы S с входным поступающим дискретным сигналом X.

По характеру отсчета дискретного времени конечные автоматы делятся на синхронные и асинхронные. Данная модель относится к асинхронному F-автомату. Асинхронный F-автомат считывает входной сигнал непрерывно, и поэтому, реагируя на достаточно длинный входной сигнал постоянной величины x, он может, как следует из (1), (2), несколько раз изменять состояние, выдавая соответствующее число выходных сигналов, пока не перейдет в устойчивое, которое уже не может быть изменено данным входным сигналом.

Простейший табличный способ задания конечного автомата основан на использовании таблиц переходов и выходов, строки которых соответствуют входным сигналам автомата, а столбцы — его состояниям. При этом обычно первый слева столбец соответствует начальному состоянию z_0. На пересечении i-й строки и k-го столбца таблицы переходов помещается соответствующее значение $\varphi (z_k, x_i)$ функции переходов, а в таблице выходов — соответствующее значение $\psi (z_k, x_i)$ функции выходов.

Описание работы F-автомата Мили таблицей переходов φ и выходов ψ иллюстрируется в табл. 1.

Пример табличного способа задания F-автомата Мили F1 с тремя состояниями, двумя входными и двумя выходными сигналами приведен в табл. 2.

При другом способе задания конечного автомата используется понятие направленного графа. Граф автомата представляет собой набор вершин, соответствующих различным состояниям автомата и соединяющих вершины дуг графа, соответствующих тем или иным переходам автомата. Если входной сигнал $x_к$ вызывает переход из состояния z_i, в состояние z_j, то на графе автомата дуга, соединяющая вершину z_i, с вершиной z_j, обозначается $x_к$. Для того чтобы задать функцию выходов, дуги графа необходимо отметить соответствующими выходными сигналами. Для автоматов Мили эта разметка производится так: если входной сигнал $x_к$ действует на состояние z_i, то, согласно сказанному, получается дуга, исходящая из z_i и помеченная $x_к$; эту дугу дополнительно отмечают выходным сигналом $y = \psi (z_i, x_k)$ (рис. 3).

При решении задач моделирования систем часто более удобной формой является матричное задание конечного автомата. При этом

Таблица 1. Описание работы F-автомата Мили таблицей переходов φ и выходов ψ

x_i	z_k			
	z_0	z_1	...	z_k
Переходы				
x_1	$\varphi(z_0, x_1)$	$\varphi(z_1, x_1)$	...	$\varphi(z_k, x_1)$
x_2	$\varphi(z_0, x_2)$	$\varphi(z_1, x_2)$	...	...
...	...	...	...	$\varphi(z_k, x_2)$
x_I	$\varphi(z_0, x_I)$	$\varphi(z_1, x_I)$	...	$\varphi(z_k, x_I)$
Выходы				
x_1	$\psi(z_0, x_1)$	$\psi(z_1, x_1)$	...	$\psi(z_k, x_1)$
x_2	$\psi(z_0, x_2)$	$\psi(z_1, x_2)$	...	$\psi(z_k, x_2)$
...	...	...	...	...
x_I	$\psi(z_0, x_I)$	$\psi(z_1, x_I)$	...	$\psi(z_k, x_I)$

Таблица 2. Табличный способа задания F-автомата Мили F1 с тремя состояниями, двумя входными и двумя выходными сигналами

x_i	z_k		
	z_0	z_1	z_2
Переходы			
x_1	z_2	z_0	z_0
x_2	z_0	z_2	z_1
Выходы			
x_1	y_1	y_1	y_2
x_2	y_1	y_2	y_1

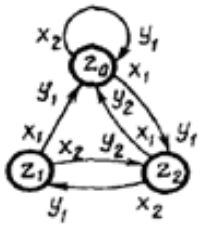


Рис. 3. Граф F-автомата Мили с тремя состояниями, двумя входными и двумя выходными сигналами

матрица соединений автомата есть квадратная матрица $C=\|c_{ij}\|$ (3), строки которой соответствуют исходным состояниям, а столбцы – состояниям перехода. Элемент $c_{ij}=x_k/y_s$, стоящий на пересечении i-й строки и j-го столбца, в случае автомата Мили соответствует входному сигналу x_k, вызывающему переход из состояния z_i, в состояние z_j, и выходному сигналу y_s, выдаваемому при этом переходе. Для автомата Мили F1, рассмотренного выше, матрица соединений имеет вид:

$$C_1 = \begin{Vmatrix} x_2 y_1 & - & x_1 y_1 \\ x_1 y_1 & - & x_2 / y_2 \\ x_1 / y_2 & x_2 / y_1 & - \end{Vmatrix}. \qquad (3)$$

Если переход из состояния z_i в состояние z_j происходит под действием нескольких сигналов, элемент матрицы c_{ij} представляет собой множество пар “вход-выход” для этого перехода, соединенных знаком дизъюнкции.

Для F-автомата Мура элемент c_{ij} равен множеству входных сигналов на переходе (z_i, z_j), а выход описывается вектором выходов (4), 1-я компонента которого — выходной сигнал, отмечающий состояние z_i.

$$\vec{y} = \begin{Vmatrix} \psi(z_0) \\ \psi(z_1) \\ \dots \\ \psi(z_k) \\ \dots \\ \psi(z_K) \end{Vmatrix}. \qquad (4)$$

Алгоритм формирования таблицы соединений можно представить графологически (рис. 4).

Таким образом, вид конечного документа таблицы соединений следует представить также графологически в виде таблицы соединений по номеру кабеля (рис. 5).

В дальнейшем будут разработаны алгоритмы формирования других документов и созданы шаблоны документов.

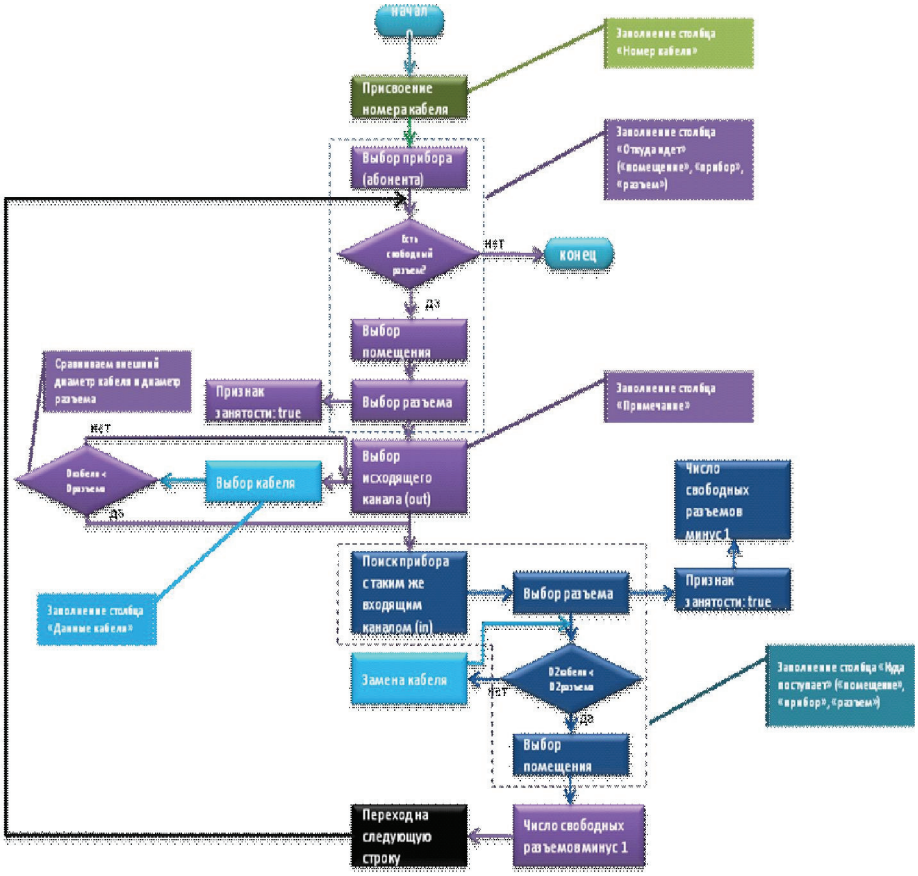
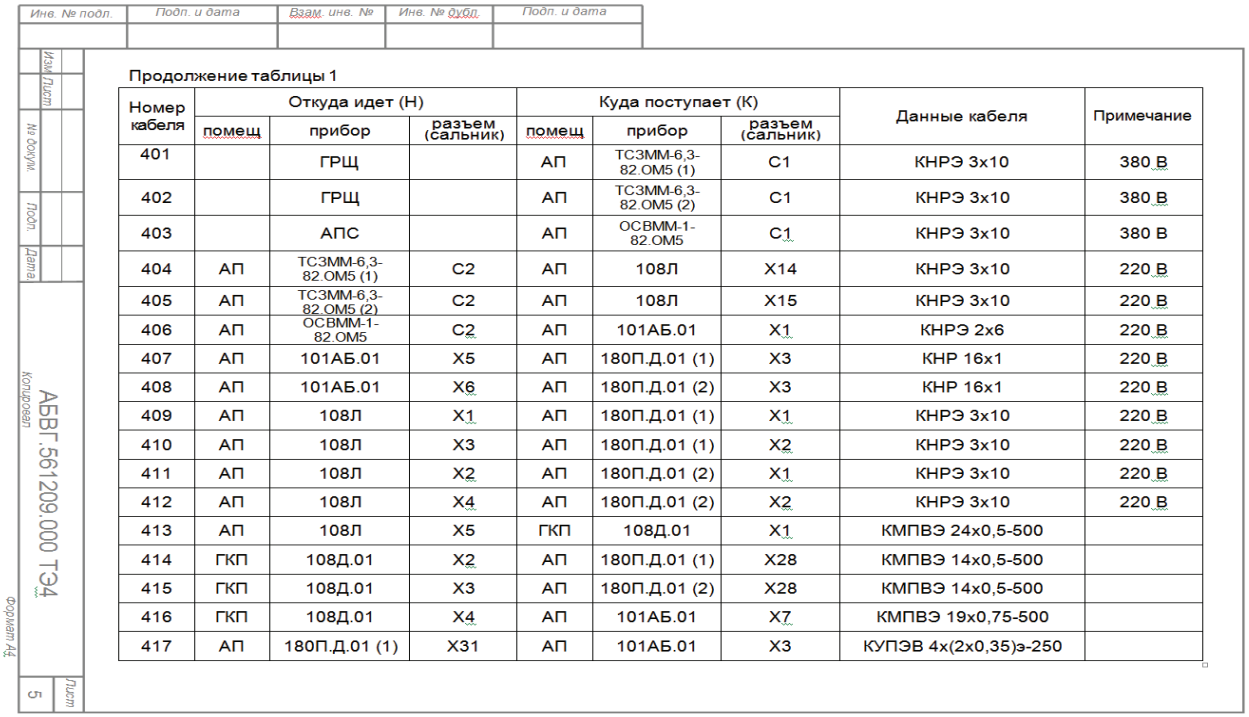


Рис. 4. Алгоритм формирования таблицы соединений



Продолжение таблицы 1

Номер кабеля	Откуда идет (Н)			Куда поступает (К)			Данные кабеля	Примечание
	помещ	прибор	разъем (сальник)	помещ	прибор	разъем (сальник)		
401		ГРЩ		АП	ТСЗММ-6,3-82.ОМ5 (1)	С1	КНРЭ 3х10	380 В
402		ГРЩ		АП	ТСЗММ-6,3-82.ОМ5 (2)	С1	КНРЭ 3х10	380 В
403		АПС		АП	ОСВММ-1-82.ОМ5	С1	КНРЭ 3х10	380 В
404	АП	ТСЗММ-6,3-82.ОМ5 (1)	С2	АП	108Л	Х14	КНРЭ 3х10	220 В
405	АП	ТСЗММ-6,3-82.ОМ5 (2)	С2	АП	108Л	Х15	КНРЭ 3х10	220 В
406	АП	ОСВММ-1-82.ОМ5	С2	АП	101АБ.01	Х1	КНРЭ 2х6	220 В
407	АП	101АБ.01	Х5	АП	180П.Д.01 (1)	Х3	КНР 16х1	220 В
408	АП	101АБ.01	Х6	АП	180П.Д.01 (2)	Х3	КНР 16х1	220 В
409	АП	108Л	Х1	АП	180П.Д.01 (1)	Х1	КНРЭ 3х10	220 В
410	АП	108Л	Х3	АП	180П.Д.01 (1)	Х2	КНРЭ 3х10	220 В
411	АП	108Л	Х2	АП	180П.Д.01 (2)	Х1	КНРЭ 3х10	220 В
412	АП	108Л	Х4	АП	180П.Д.01 (2)	Х2	КНРЭ 3х10	220 В
413	АП	108Л	Х5	ГКП	108Д.01	Х1	КМПВЭ 24х0,5-500	
414	ГКП	108Д.01	Х2	АП	180П.Д.01 (1)	Х28	КМПВЭ 14х0,5-500	
415	ГКП	108Д.01	Х3	АП	180П.Д.01 (2)	Х28	КМПВЭ 14х0,5-500	
416	ГКП	108Д.01	Х4	АП	101АБ.01	Х7	КМПВЭ 19х0,75-500	
417	АП	180П.Д.01 (1)	Х31	АП	101АБ.01	Х3	КУПЭВ 4х(2х0,35)э-250	

Рис. 5. Внешний вид таблицы соединений

СПИСОК ЛИТЕРАТУРЫ

1. *Советов Б.Я., Яковлев С.А.* Моделирование систем: учеб. для вузов. 3-е изд., переработанное и дополненное.М.: Высшая школа, 2001. 343 с.
2. *Соснин П. И.* Вопросно-ответное программирование человеко-компьютерной деятельности. Ульяновск: УлГТУ, 2010. 240 с.
3. ГОСТ 2.701-2008 ЕСКД. Схемы виды и типы. Общие требования к выполнению
4. ГОСТ 2.114-95 ЕСКД. Технические условия

METHODS DEVELOPMENT PROJECT-SYSTEM DOCUMENTATION BY USING PLANE-ACCOUNTING UNIT

© 2012 O.A. Verushkin

Federal Research and Production Center of JSC “NPO” Mars “, Ulyanovsk

The article made a partial review of the methods of design and development of system documentation for companies that make automated control systems, the possibility of using the system DocWIQA.net documenting the development of design and system documentation.

Keywords: design and documentation system, the circuit structure, the system documentation DocWIQA.net, document generation, model discrete deterministic

Oleg Verushkin, Engineer. E-mail: mars@mv.ru