



МЕТОДЫ УСКОРЕНИЯ СОЗДАНИЯ ПРОЕКТА В SCADA-СИСТЕМЕ iFix

А.Я. Коростелев, В.В. Грибов, А.А. Койда (ООО "ПраймГруп")

В статье рассматривается проблема ускорения создания проектов в SCADA-системах и пути снижения трудоемкости разработки экранных форм. На примере SCADA-системы iFix приводятся практические приемы повышения эффективности работы и производительности труда программистов при проектировании АСУТП. Описанные методы целесообразно применять в т.ч. при реализации АСУТП в нефтегазовой и химической отраслях.

При сравнительном анализе SCADA-систем, присутствующих на современном рынке, обычно учитывают такие факторы, как поддержка внешних интерфейсов, сетевые возможности, быстродействие, встроенные языки программирования, цена и др. При выборе SCADA-системы разработчик оценивает возможность реализации всех требуемых функций и обеспечения заданных характеристик. Часто этим требованиям удовлетворяет целый ряд SCADA-систем, причем, со сходными ценовыми показателями.

Однако, при этом часто забывают про еще один немаловажный фактор – трудоемкость создания проекта в данной SCADA-системе. При более детальном рассмотрении оказывается, что для реализации одной и той же функции в разных SCADA-системах требуется затратить различное количество времени и усилий. Иногда то, что в одной SCADA-системе осуществляется стандартными средствами путем двух щелчков мышью, то в другой системе может потребоваться значительные усилия (например, написания dll-библиотек или ActiveX-объектов). Хотя формально для решения поставленной задачи подходят обе системы.

Сокращение времени работы над проектом, а также легкость и удобство внесения в него изменений, позволяют разработчику больше внимания уделять потребностям и желаниям заказчика, не отвлекаясь на рутинную и однообразную работу. Благодаря грамотному использованию средств быстрого и легкого создания проекта в SCADA-системе, появляется возможность повысить производительность труда, уменьшить число программистов, занятых в проекте, в одни и те же сроки реализовывать большее число проектов.

Даже при наличии у SCADA-системы средств, позволяющих автоматизировать процесс создания и редактирования проекта, разработчики не всегда их используют. Зачастую это связано с нежеланием разработчиков разбираться в тонкостях программирования, отсутствием уверенности в том, что затраты времени на изучение этих возможностей в дальнейшем окупятся. Например, может возникнуть необходимость изменить цвет некоего графического объекта на нескольких десятках экранных форм. Эту задачу можно решить вручную, выполняя в течение нескольких часов однообразные операции. Есть и дру-

гой способ, – написать программу, которая выполнит ту же задачу за несколько секунд. Конечно, написание такой программы отнимает некоторое время, но, как показывает практика, затраченное время очень быстро окупается (особенно для тех разработчиков, которые работают над многими проектами). Помимо этого, ту же программу можно путем небольших изменений адаптировать для решения любой похожей задачи.

Применение методов, описываемых в данной статье, особенно оправдано для крупных проектов, содержащих большое число однотипных элементов и насыщенные сложные экранные формы. Такие проекты характерны для нефтегазовой и химической отраслей промышленности. Описываемые методы применяются при создании АСУТП таких объектов, как установки подготовки нефти и газа, нефтеперекачивающие и компрессорные станции, магистральные трубопроводы, нефтебазы, нефте- и газоперерабатывающие и химические производства и др.

Целью данной статьи стало желание обратить внимание специалистов на проблему ускорения создания проектов в SCADA-системах, а также привести несколько реально используемых приемов при работе со SCADA-системой iFix (выпускаемой компанией Intellution, которая сейчас приобретена корпорацией GE Fanuc).

В систему iFix встроен язык программирования Visual Basic for Applications (VBA), который может использоваться как в режиме выполнения, так и в режиме конфигурирования.

Применение описываемых методов возможно и в других SCADA-пакетах, поддерживающих VBA в режиме конфигурирования (например, WinCC, фирмы Siemens), хотя программная реализация будет, конечно, отличаться.

Далее приводится описание некоторых полезных возможностей VBA в iFix.

Разработка графических форм и их привязка к переменным

iFix позволяет программным путем создавать экранные формы (в iFix они называются рисунками – Picture), располагать на них графические объекты,

Название	Значение	Минимум	Максимум
Уровень (рез. п. 31)	0.00	0.00	100.00
Плотность (рез. п. 31)	788.35	690.00	900.00
Температура (рез. п. 31)	2.15	-40.00	50.00
Уровень подтоварной воды (рез. п. 31)	68.54	10.00	135.00
Уровень в рез. п. 31	9.37	0.00	20.00
Плотность (рез. п. 32)	788.35	690.00	900.00
Температура (рез. п. 32)	2.15	-40.00	50.00
Уровень подтоварной воды (рез. п. 32)	68.54	10.00	135.00
Уровень в рез. п. 32	9.37	0.00	20.00
Температура (рез. п. 33)	788.35	690.00	900.00
Уровень подтоварной воды (рез. п. 33)	68.54	10.00	135.00
Уровень в рез. п. 33	9.37	0.00	20.00
Плотность (рез. п. 34)	788.35	690.00	900.00
Уровень подтоварной воды (рез. п. 34)	68.54	10.00	135.00
Уровень в рез. п. 34	9.37	0.00	20.00
Плотность (рез. п. 35)	788.35	690.00	900.00
Температура (рез. п. 35)	2.15	-40.00	50.00
Уровень подтоварной воды (рез. п. 35)	68.54	10.00	135.00
Уровень в рез. п. 35	9.37	0.00	20.00
Плотность (рез. п. 36)	788.35	690.00	900.00
Температура (рез. п. 36)	2.15	-40.00	50.00
Уровень подтоварной воды (рез. п. 36)	68.54	10.00	135.00
Уровень в рез. п. 36	9.37	0.00	20.00

Рис. 1

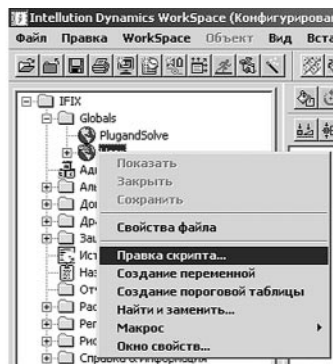


Рис. 2

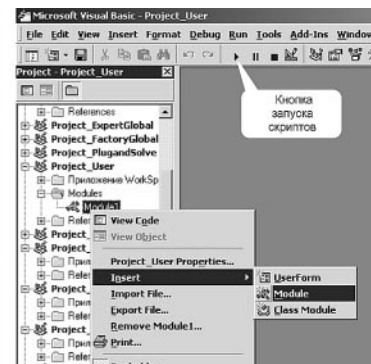


Рис. 3

менять их свойства, привязывать эти свойства к значениям тегов БД.

Обычно, все экранные формы создаются разработчиком с использованием встроенного графического редактора и затем нужные объекты анимируются путем привязки тегов БД к свойствам объекта с помощью обычных средств системы.

Есть также и другой способ создания экранных форм – программный. Этот способ в целом, конечно, сложнее и требует определенных затрат времени. Однако, его все же имеет смысл применять в тех случаях, когда создание экранной формы вручную потребует очень много времени и рутинной работы, например, если требуется создать список переменных в табличной форме, и число переменных составляет несколько сотен.

Примерный вид такой экранной формы показан на рис.1.

Данная экранная форма содержит статические объекты – линии (разделители строк и столбцов таблицы), прямоугольник (фон таблицы), текстовые элементы (заголовки столбцов), а также динамические элементы – datalink (названия, значения, границы шкал параметров), непосредственно отображающие поля тегов БД iFix.

Для того, чтобы создавать экранные формы программным путем нужно знать всего несколько основных приемов: собственно создание пустого рисунка, создание на рисунке статических и динамических объектов, изменение свойств уже существующих объектов. Рассмотрим эти приемы каждый по-отдельности.

Создание пустого рисунка

Скрипт, который будет создавать рисунок, должен выполняться в режиме конфигурирования. Можно разместить этот скрипт в отдельном модуле файла User.fgx. Это можно сделать следующим образом: в системном дереве раскрыть пункт "Globals", щелкнуть правой кнопкой мыши на пункте "User", в выпадающем меню выбрать "Правка скрипта" (рис. 2), затем в редакторе VBE щелкнуть правой кнопкой мыши на пункте "Project_User", выбрать команду "Insert -> Module" (рис. 3).

После этого будет создан новый модуль, в котором можно создавать свои собственные скрипты. Запус-

кать их можно с помощью кнопки "▶" прямо из редактора VBE.

Можно также создать свою панель инструментов, со специальной кнопкой, которая будет запускать требуемый скрипт (этот вариант немного сложнее).

Для создания рисунка используется функция GeneratePicture. В качестве параметра ее передается структурная переменная типа PictureInfo, содержащая свойства, которыми будет обладать созданный рисунок. В случае успешного завершения функции она возвращает значение True, в противном случае False.

На листинге 1 приведен фрагмент программы, который осуществляет создание рисунка с именем "BigTable".

Структурная переменная aPicInfo заполняется свойствами создаваемого рисунка (расположение на экране, размеры, цвет фона, имя и т.д.). Затем вызывается функция GeneratePicture, которая и создает рисунок с указанными свойствами. В случае невозможности создания рисунка (например, если рисунок с таким именем уже существует) выдается сообщение об ошибке.

Создание на рисунке статических объектов

Для создания объектов в iFix используется метод BuildObject. Как параметр этому методу передается строковая константа, указывающая тип создаваемого объекта. Список возможных значений этого параметра можно посмотреть в системе помощи iFix. Метод возвращает ссылку на созданный объект.

На листинге 2 приведен фрагмент программы, выполняющий создание прямоугольника, линии и текста на рисунке "BigTable", созданным в предыдущем примере. Перед началом работы программы этот рисунок должен быть открыт (либо вручную, либо с использованием процедуры OpenPicture).

Для получения ссылки на объект редактируемого рисунка используется метод FindObject объекта System. Как параметр указывается имя искомого объекта. Заранее необходимо открыть этот рисунок, иначе он не будет найден.

Статические объекты создаются с помощью метода BuildObject. Как параметр указывается тип создаваемого объекта. Далее вновь созданному объекту присваиваются нужные свойства.

```

' Получаем ссылку на объект рисунка
Set oPic = System.FindObject("BigTable")
' Создаем прямоугольник
Set oRect = oPic.BuildObject("Rect")
oRect.HorizontalPosition = 5 ' Горизонтальная координата
oRect.Width = 90 ' Ширина
oRect.VerticalPosition = 5 ' Вертикальная координата
oRect.Height = 65 ' Высота
oRect.ForegroundColor = RGB(255, 255, 255)
' Цвет
oRect.EdgeWidth = 3 ' Ширина кромки
' Создаем линию
Set oLine = oPic.BuildObject("Line")
oline.StartX = 5 ' Горизонтальная координата
' начальной точки
oline.EndX = 95 ' Горизонтальная координата
' конечной точки
oline.StartY = 8 ' Вертикальная координата
' начальной точки
oline.EndY = 8 ' Вертикальная координата
' конечной точки
' Создаем текст
Set oText = oPic.BuildObject("Text")
oText.Caption = "Название" ' Содержание текста
oText.ScalesWidth = False ' Возможность изменения ширины
oText.AutoSize = False ' Автоматическое изменение размера
oText.HorizontalPosition = 5 ' Горизонтальная координата
oText.VerticalPosition = 5.5 ' Вертикальная координата
oText.FONTSIZE = 14 ' Размер шрифта
oText.ForegroundColor = 0 ' Цвет
oText.Alignment = 1 ' Выравнивание
oText.Width = 45 ' Ширина

```

Листинг 1

Создание на рисунке динамических объектов

Затем необходимо создать динамические элементы экранной формы, привязанные к переменным БД iFix.

Рассмотрим создание динамических объектов на примере объекта "Связь данные" (или "Datalink"). С помощью рассмотренного выше метода BuildObject создается сам объект Datalink и подчиненный ему объект Format, через который осуществляется связь с тегом БД. Объект Format имеет свойство Source, которое должно содержать полное имя тега. Таким образом, объект Format получает данные из тега БД, преобразует их в необходимую форму, и передает в объект Datalink. Для соединения объектов Datalink и Format используется метод Connect. Структура соединения показана на рис. 4.

На листинге 3 приведен фрагмент подпрограммы, выполняющий эти действия.

Сначала создается объект "Datalink" и ему присваиваются необходимые свойства. Затем в нем создается объект Format для связи с тегом БД. Полное имя тега указывается в свойстве Source объекта Format. Непосредственно соединение осуществляется методом Connect объекта Datalink. В качестве первого параметра указывается свойство объекта Datalink, которое мы хотим анимировать (это свойство Caption), а в качестве второго параметра — источник данных для анимации (это свойство OutputValue объекта Format).

```

Dim aPicInfo As PictureInfo
' Описываем свойства создаваемого рисунка
aPicInfo.lfTopPct = 0 ' Вертикальная координата
aPicInfo.lfLeftPct = 0 ' Горизонтальная координата
aPicInfo.lfHeightPct = 75 ' Высота
aPicInfo.lfWidthPct = 100 ' Ширина
aPicInfo.lbkColor = RGB(200, 200, 200)
' Цвет
aPicInfo.szName = "BigTable" ' Имя
aPicInfo.bPixels = False ' Размерность, в которой задаются
' координаты (пиксели или проценты)
aPicInfo.bTitlebar = False ' Наличие строки заголовка
aPicInfo.bSystemMenu = False ' Наличие системного меню
aPicInfo.bResizable = False ' Возможность изменения размеров
aPicInfo.bAlwaysOnTop = False ' Признак "всегда сверху"
aPicInfo.bRuntimeVisible = True ' Видимость в режиме исполнения
' Создаем рисунок
If GeneratePicture(aPicInfo) = False Then
MsgBox "Can't create picture"
Exit Sub
End If

```

Листинг 2

```

' Получаем ссылку на объект рисунка
Set oPic = System.FindObject("BigTable")
' Создаем datalink
Set oDatalink = oPic.BuildObject("Datalink")
' Присваиваем ему требуемые свойства
oDatalink.Name = "DatalinkDesc" ' Имя
oDatalink.VerticalPosition = 8 ' Вертикальная координата
oDatalink.HorizontalPosition = 7 ' Горизонтальная координата
oDatalink.Caption = "description" ' Содержание текста (в режиме
' конфигурирования)
oDatalink.ForegroundColor = 0 ' Цвет
oDatalink.FontStyle = 0 ' Размер шрифта
oDatalink.Alignment = 0 ' Выравнивание
oDatalink.IsSelectable = True ' Выбираемый
' Соединяем его с переменной через объект Format
Set oFormat = oDatalink.BuildObject("Format")
oFormat.Source = "Fix32.IFIX.AI_000.A_CV"
oDatalink.Connect "Caption", oFormat.FullyQualifiedName + ".OutputValue",
IStatus

```

Листинг 3

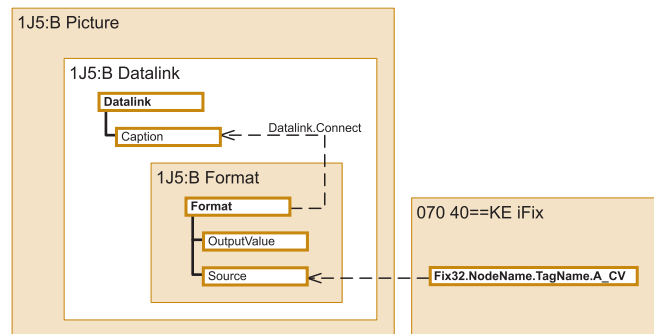


Рис. 4

Изменение свойств существующих объектов

Для изменения свойств уже существующего объекта необходимо получить ссылку на него, и затем назначить свойствам новые значения. Ссылку на объект можно получить двумя путями — с использованием метода FindObject родительского объекта или

Первый вариант:
 Set oPic = System.FindObject("BigTable")
 Set oText = oPic.FindObject("Text1")
 oText.HorizontalPosition = oText.HorizontalPosition + 5

Второй вариант:
 Set oPic = System.FindObject("BigTable")
 Set oText = FindLocalObject(oPic, "Text")
 oText.HorizontalPosition = oText.HorizontalPosition + 5

Листинг 4

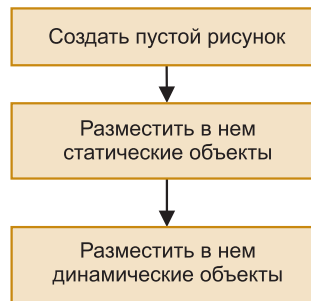


Рис. 5

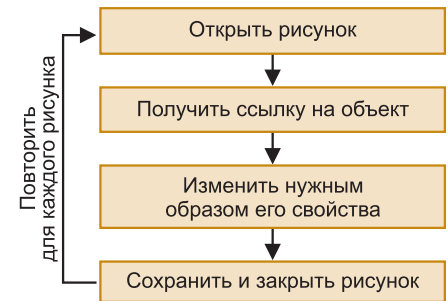


Рис. 6

процедуры FindLocalObject. Метод FindObject требует в качестве параметра полное имя объекта, на который мы хотим получить ссылку, а для процедуры FindLocalObject имя можно указывать неполностью, достаточно указать лишь начало имени объекта. Процедура FindLocalObject требует еще один параметр – ссылку на родительский объект, в котором будет осуществляться поиск.

Сначала получается ссылка на объект редактируемого рисунка, затем ссылка на объект, свойства которого мы хотим изменить. Далее свойствам объекта присваиваются новые значения. Например, в данном случае текст смещается на 5 единиц вправо (листинг 4).

Заключение

Таким образом, создание экранной формы, подобной той, что показано на рис. 1, сводится к последовательному применению описанных выше методов. Упрощенно процедура создания такой экранной формы показана на рис. 5. После создания пустого рисунка на нем размещаются текст заголовка, фон таблицы, рамка, линии-разделители строк и столб-

цов. Для заполнения таблицы динамическими объектами (datalink) удобно организовать цикл по строкам таблицы. Имена тегов, к которым осуществляется привязка, можно, например, считывать из текстового файла.

Если требуется изменить характеристики какого-либо объекта, присутствующего на многих экранных формах, то это можно сделать применив процедуру, показанную на рис. 6.

Умело комбинируя описанные выше приемы можно значительно уменьшить время создания некоторых типов экранных форм. Более того, таким методом можно легко создавать экранные формы, разработать которые вручную практически невозможно из-за слишком больших затрат времени и труда.

Опыт авторов показывает, что применение процедур автоматического изменения экранных форм не только сокращает время разработки проекта, но и позволяет быстро его адаптировать по требованию заказчика, а также улучшать эргономические свойства визуализации ТП в зависимости от опыта и привычек операторов-технологов.

Коростелев Александр Яковлевич – инженер-программист, **Грибов Виктор Валерьевич** – руководитель группы программирования, **Койда Александр Алексеевич** – директор департамента АСУ и энергетики.

Контактный телефон (095)725-44-32 E-mail:info@primegroup.ru Http://www.primegroup.ru

В компании "АЗР Автомобиль" внедрена ERP/EAM-система IFS Applications

Компания КФС объявила о завершении проекта по внедрению системы IFS Applications в компании "АЗР Автомобиль – звезда Руси" (www.mb-asg.com), крупнейшем дилере DaimlerChrysler в России. Создано решение "IFS Автобизнес", готовое к тиражированию и планируемое к сертификации в DaimlerChrysler.

"АЗР Автомобиль – звезда Руси" (АЗР) – компания со 100% иностранным капиталом и руководствуется в своей деятельности стандартами концерна DaimlerChrysler. Помимо продаж автомобилей Mercedes-Benz, в сферу деятельности АЗР входит сервисно-техническое обслуживание, тюнинг, реализация запчастей и аксессуаров. Компания имеет крупное многопрофильное предприятие в Москве, предприятия в г. Балашихе и Ярославле.

До выбора IFS Applications компания использовала ряд разрозненных систем автоматизации бухучета и сервисного обслуживания. Отсутствие интеграции и моральное старение использовавшихся систем поставили на повестку дня вопрос об их замене на комплексную интегрированную систему. Процесс выбора решения, наиболее полно отвечающего требованиям заказчика, потребовал рассмотрения не только представ-

ных в то время на рынке специализированных решений, но и ERP-систем, обладающих широкими функциональными возможностями и ярко выраженным фокусом на управление техобслуживанием и ремонтами (MRO, EAM).

АЗР остановил свой выбор на системе IFS Applications, в первую очередь, в силу большей открытости и гибкости, так как изначально было очевидно, что именно эти качества окажутся определяющими в процессе адаптации решения к требованиям отрасли. В то же время IFS Applications уже не один год успешно использовалась рядом автодилеров, сервисных и тюнинговых центров за рубежом – в частности, шведскими дилерами Volvo.

Переход к использованию в АЗР новых программных средств автоматизации осуществлялся поэтапно. Первым крупным шагом стало внедрение финансового решения IFS Applications, что позволило автоматизировать расчеты с покупателями и поставщиками, управление финансовыми потоками и бухгалтер. На втором этапе была выполнена автоматизация внутреннего склада, в процессе которой как на прототипе отработывались наиболее важные решения по управлению складским хозяйством и снабжением.

Контактный телефон (095) 787-70-40.