

УДК 681.3.068:62-523

## ПРИНЦИПЫ ИНТЕГРАЦИИ КОМПОНЕНТОВ ЭЛЕКТРОАВТОМАТИКИ НА ПРИМЕРЕ ГРАФИЧЕСКОГО РЕДАКТОРА 3D-СЦЕНЫ ВИЗУАЛИЗАЦИИ

Г.М.Мартинов, Н.В.Козак

*Представлены принципы по интеграции компонентов прикладной составляющей электроавтоматики. Целью интеграции является снижение стоимости разработки и повышение качества, за счет применения готовых решений. Принципы интеграции проиллюстрированы на примере встраивания графического редактора в программный комплекс CoDeSys.*

### Введение

Наиболее динамично развивается и растёт в последнее время, область человеко-машинного интерфейса (HMI) систем управления. Развитие в области технологий аппаратного и программного обеспечения привело к совершенствованию и повышению точности устройств отображения и систем, которыми оно управляет. Продуманная графика и анимация, а также способность выполнять более сложные функции, расширенные возможности соединения и беспроводной связи стимулируют разработку программного обеспечения и развитие в этой области [1].

Приложения систем управления, исполняемые на персональных компьютерах, в том числе и промышленных, и реализующие интерфейс с пользователями решают задачи, которые носят прикладной характер. Типичными прикладными задачами для систем управления электроавтоматикой можно представить следующие: «разработка управляющих программ», «эмуляции исполнения и отладки управляющих программ», «выполнения управляющих программ», «моделирования объекта управления», «конфигурирования PLC» и другие в соответствии с предъявляемыми требованиями.

В технологиях разработки систем управления установились тенденции применения модульного и компонентного подходов разработки и тенденции программной реализации контроллера электроавтоматики (SoftPLC) [2, 3].

Это предполагает использование стандартных технологий разработки, таких как COM, DCOM, ActiveX, применение преимуществ управляемого кода, в том числе и расширенных возможностей языка C# платформы .NET[4].

Актуальной становится решение задачи интеграции прикладных компонентов в системах управления электроавтоматикой. Принципы интеграции проиллюстрированы на примере встраивания графического редактора 3D-сцены визуализации технологических процессов в программный комплекс CoDeSys.

### Принципы интеграции прикладных программных компонентов электроавтоматики Декомпозиция программных компонентов электроавтоматики

На базе функциональности прикладной составляющей электроавтоматики были выделены две группы программных компонентов. Первая группа - это компоненты окружения, а вторая - прикладные компоненты [5].

Компоненты окружения предоставляют главное окно приложения и размещаемые в нём области: меню, панели инструментов, документов, строк состояния, плавающие окна и механизмы управления этими окнами.

Прикладные компоненты реализуют задачи управления электроавтоматикой и осуществляют функциональное наполнение приложения. Они используют предоставляемые сервисы и интерфейсы компонентов окружения для реализации содержимого плавающих окон, команд в меню и панелях инструментов, панелей отображения статуса в строке состояния и т.п.

Для анализа набора и структуры прикладных компонентов электроавтоматики используют матрицу компонентов. Матрицу компонентов строят в виде пересечения пользовательских задач по горизонтали с логическими уровнями виртуальной структуры по вертика-

ли. Таким образом, столбцы полученной матрицы определяют многообразие пользовательских задач системы управления электроавтоматикой, а строки - способ компонентной реализации программного обеспечения.

### Принципы интеграции

Задача интеграции сводится к связыванию в единую систему модулей готовых программных решений (рис. 1). Связывающим звеном здесь выступает специальный модуль, разработанный для передачи и трансформации данных между перечисленными модулями – это компоненты интеграции. В результате, в единой системе функционируют готовые решения в области пользовательского интерфейса, в области разработки УП для PLC и в области серверной части электроавтоматики, которая непосредственно осуществляет управление объектом.

Требования к компонентам пользовательского интерфейса меняются с выпуском каждой новой версии операционной системы Windows, по этому их следует приобретать у фирм специализирующиеся на их выпуске. Компоненты средств разработки управляющих программ PLC реализуют редакторы языков стандарта МЭК 61131-3, их компиляторы, средства отладки, выполнения программ, средства ведения проекта системы, конфигурирования оборудования и т.п. Бесспорным лидером на рынке является система CoDeSys. Компоненты аппаратных средств PLC реализуют работу с контроллерами электроавтоматики. На сегодняшний день это могут быть изделия любого производителя на рынке.

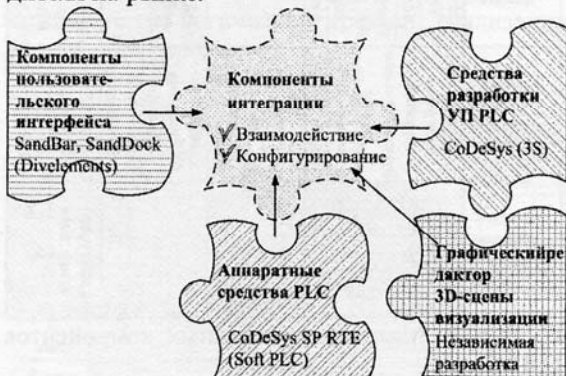


Рис. 1. Связывание в единую систему компонентов прикладной составляющей электроавтоматики

В систему электроавтоматики встраивается независимая программная разработка – графический редактор 3D-сцены визуализации, для моделирования процесса управления.

В основе принципов интеграции, лежит использование программных интерфейсов, компонентов прикладной составляющей электро-

автоматики. Интерфейсы интеграции, как составная часть программных интерфейсов, во-первых, служат для обеспечения функционирования интегрируемых компонентов в составе приложений прикладной составляющей (производят загрузку/выгрузку библиотек компонентов, управление приоритетами потоков исполнения, размещение элементов управления в интерфейсе пользователя и т.п.). Во-вторых, обеспечивают взаимодействие между интегрируемыми компонентами (формализуют и специфицируют функциональность способов их взаимодействия).

Принципы интеграции компонентов предполагают спецификацию, разработку и использование набора программных интерфейсов. Принцип взаимодействия компонентов интеграции с интегрируемыми компонентами и компонентами прикладной составляющей электроавтоматики, посредством использования и предоставления программных интерфейсов, представлен на рисунке (рис. 2).



Рис. 2. Пример использования программных интерфейсов в компонентах интеграции

Программные интерфейсы интегрируемых компонентов и компонентов прикладной составляющей не должны изменяться в ходе интеграции (на рисунке они заштрихованы). Интерфейс компонентов интеграции реализуется с учётом механизмов взаимодействия в прикладных компонентах и использованием интерфейса интегрируемых компонентов.

За основу интерфейсов принимаются типы данных, классы, структуры, делегаты, перечисления и COM-интерфейсы, реализуемых прикладных компонентов. Таким образом, исключается необходимость использования типов и интерфейсов интегрируемых компонентов в механизмах взаимодействия с прочими компонентами прикладной составляющей, они инкапсулируются в компонентах интеграции.

Разделяем два способа предоставления интерфейсов интеграции. Первый, реализация интерфейсов интеграции осуществляется на стороне компонентов интеграции для обёртывания

под них функций и данных из интегрируемых компонентов. Второй, интерфейсы интеграции реализуются как сервисы, предоставляющие свои функции и события интегрируемым компонентам. Во втором случае в компонентах интеграции осуществляется вызов предоставляемых функций и реакция на события.

Обобщенная схема шагов интеграции, с используемыми и получаемыми данными этого процесса, представлена ниже:

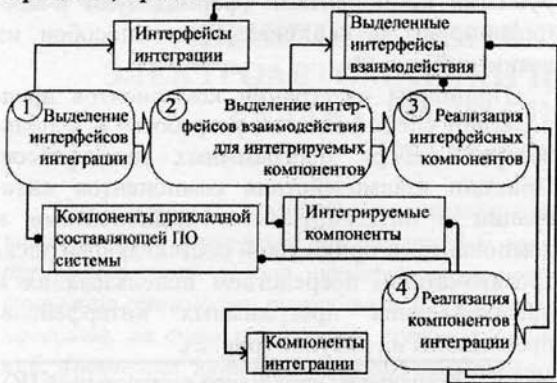


Рис. 3. Обобщенная схема шагов интеграции:  $\Rightarrow$  - последовательность деятельности процесса;  $\curvearrowright$  - получение новых данных процесса в ходе шага;  $\leftarrow$  - использование данных в ходе шага

На первом шаге интерфейсы интеграции выделяют из программных интерфейсов компонентов прикладной составляющей. Далее производится выделение программных интерфейсов необходимых для взаимодействия с другими прикладными задачами. Это делается в соответствии с логическими уровнями реализации компонентов.

На основе выделенных интерфейсов взаимодействия реализуются интерфейсные компоненты. И, наконец, реализуются сами компоненты интеграции на основе полученных интерфейсов взаимодействия и интеграции.

Взаимодействие между интегрируемыми компонентами осуществляется посредством выделенных программных интерфейсов взаимодействия и, интеграции.

Представленные принципы и последовательность интеграции компонентов были использованы для интеграции компонентов графического редактора сцены визуализации в состав прикладных компонентов электроавтоматики программного комплекса CoDeSys (разработчик 3S).

**Интеграция компонентов графического редактора 3D-сцены визуализации**

Разработка и отладка сложных управляющих программ электроавтоматики немислимо

без моделирования управляемого объекта. Трех мерное моделирование более полно отображает объекты управления в рамках технологического процесса.

Для построения технологического окружения процесса и самих объектов управления был разработан прототип графического редактора 3D-сцены визуализации техпроцесса.

**Программный комплекс электроавтоматики CoDeSys**

Широкий круг поддерживаемого аппаратного обеспечения, инструментальные средства разработки управляющих программ всех языков стандарта МЭК, а также четкая модульная организация в решении пользовательских задач, превратили системы CoDeSys в стандарт де-факто в реализации клиентской части электроавтоматики [6].

Матрица прикладных компонентов[5] для системы CoDeSys представлена на рис. 4. Компоненты обладают узкой специализацией, что повышает гибкость настройки пользовательских конфигураций системы.

Логические уровни	Пользовательские задачи системы управления						
	Разработка управляющих программ	Отладка управляющих программ	Моделирование объекта управления	Конфигурирование PLC			
Визуальное представление	Редакторы ST, FBD, SFC, CFC языков	Панели команд редакторов	Редактор отслеживания значений	Сцена визуализации	Редактор визуализации	Редактор конфигурирования задания	Редактор сетевых соединений
Управление данными	Объекты ST, FBD, SFC	Компьютер	Автоопределение переменных	Команда визуализации	Менеджер сети	Команда сети	
Хранение и предоставление данных	Объекты БД программ, функций, функциональных блоков	Генератор кода хэб, 106 пресетов и др.	Языковая модель	Объекты БД наблюдения	Библиотека визуальных объектов	Объекты БД визуализации	Объекты БД PLC, Network List
Взаимодействие с аппаратурой	Сервер разработки «ENI Server»			Эмулятор PLC		Конфигурирование задачи	

Рис. 4. Матрица прикладных компонентов CoDeSys

В системе CoDeSys соблюдено разграничение компонентной реализаций пользовательских задач по логическим уровням. Уровень хранения и предоставления данных для большинства задач реализуется объектами базы данных системы. Доступ к их данным осуществляют компоненты, реализованные по типу менеджеров, например, менеджер объектов или менеджер языковых моделей, которые располагаются на уровне управления данными.

Программные средства системы CoDeSys, были дополнены механизмом интеграции компонентов сторонних производителей. Далее проиллюстрирована интеграция графического редактора 3D-сцены визуализации.

**Интеграция компонентов графического редактора**

Графический редактор 3D-сцены визуализации предоставляет пользователю средства трех мерного представления объекта управления и его окружения. Исполнительные звенья механизмов, их приводы и объекты окружения сцены размещаются и компоуются пользователем при построении модели технологического окружения процесса. Далее производится подключения переменных управляющей программы к соответствующим полям в сцене (включение выключение двигателя, координаты исполнительного звена, шаги переключения барабана инструментов и т.п.). В режиме исполнения управляющей программы пользователь имеет возможность для выбора позиций слежения за техпроцессом, для просмотра значений переменных процесса, для остановки выполнения программы по условию достижения определённого значения переменной и некоторые другие функции.

Интеграция компонентов графического редактора 3D-сцены визуализации объекта управления осуществляется для задачи Моделирования объекта управления. На рис. 5 в виде фрагмента матрицы представлены компоненты графического редактора (сплошной линией) и компоненты интеграции (пунктиром), расположенные на логических уровнях реализации задачи моделирования.

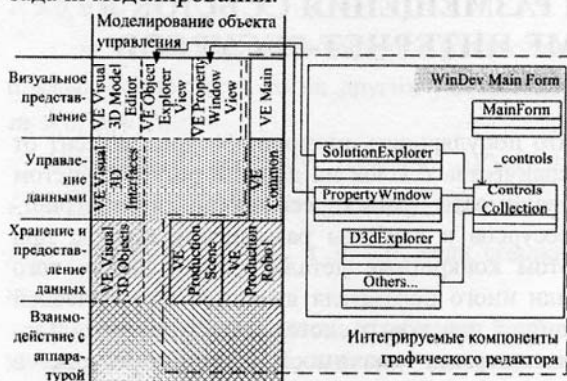


Рис. 5. Создание компонентов интеграции для компонентов графического редактора 3D-сцены визуализации

Главный элемент управления графический редактор, реализуется в компоненте VE Main. Он обертывается при интеграции в компонент редактора VE Visual 3D Model Editor. Для хранения и представления данных в структуре

проекта CoDeSys на уровне хранения и предоставления данных реализуется компонент VE Visual 3D Objects. Этот компонент так же является компонентом интеграции для компонентов VE Production Scene (предоставляющего информацию об объектах интерьера сцены) и VE Production Robot (предоставляющего информацию об объектах частей роботов).

Элементы управления инструментария графического редактора, до интеграции были реализованы в общем компоненте главного окна приложения Win Dev Main Form в виде классов Solution Explorer для окна библиотеки 3D элементов сцены и PropertyWindow окна свойств для этих элементов при расположении в сцене. Для каждого класса окна инструментария были реализованы соответствующие компоненты интеграции на уровне визуального представления VE Object Explorer View и VE Property Window View соответственно.

Программные интерфейсы взаимодействия компонентов графического редактора, служащие для управления данными, вынесены в компоненте интеграции VE Visual 3D Interfaces.

Проиллюстрированный пример графического редактора 3D-сцены визуализации (рис. 6) позволяет производить отладку управляющих программ, разработанных в среде CoDeSys, на модели робота манипулятора в некотором технологическом окружении. Т.о. процесс разработки и отладки управляющих программ не требует привлечения оборудования производства и может осуществляться параллельно с процессом подготовки и построения производства. Возможен вариант симуляции технологического процесса во время работы системы, на основе данных полученных с датчиков технологического процесса.

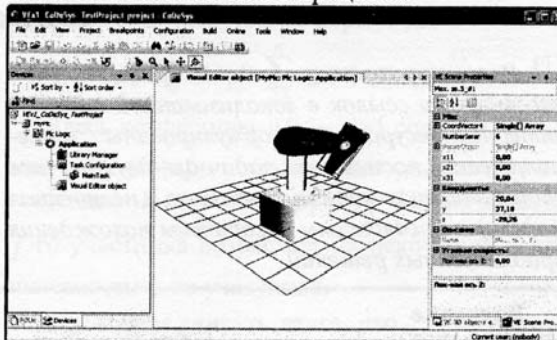


Рис. 6. Результат интеграции компонентов графического редактора

**Заключение**

Для создания конкурентно способного продукта система управления должна охватывать широкий круг прикладных задач. Также системе нужна возможность компоноваться только

нужными для пользователя задачами. Стремление вовремя выпустить продукт на рынке автоматизации предполагает использование готовых и отлаженных решений сторонних производителей, а также наличие механизмов интеграции в состав системы. Сложность интеграции заключается в разработке правильной архитектуры прикладной составляющей, в формализации её описания и в отсутствие единой методологии для самого процесса.

Декомпозиция программных компонентов прикладной составляющей электроавтоматики позволяет сформировать обобщенный взгляд на структуру компонентов. Инструмент матрицы компонентов применяется как для анализа структуры компонентов системы, так и для описания процесса интеграции набора компонентов в отдельных задачах.

Принципы интеграции основываются на спецификации интерфейсов и разработке компонентов интеграции. Применение этих принципов позволяет снизить себестоимость разработки, за счет применения готовых решений, повысить качество создаваемого программного обеспечения, путём использования отлаженных компонентов, и сократить сроки выпуска продукта на рынок.

Встраивание графического редактора 3D-сцены визуализации в системе CoDeSys предоставляет разработчику управляющих программ электроавтоматики визуальные средства для их отладки, а оператору возможность си-

муляции объекта управления на экран системы.

#### **Список использованных источников**

1. Катцель Д. Инвестиции в HMI отражают расширяющийся рынок// Control Engineering Россия, 2006. №1. С. 43-49.
2. Сосонкин В.Л., Мартинов Г.М. Системы числового программного управления. – М.: Логос, 2005.– 296 с
3. Сосонкин В.Л., Мартинов Г.М., Перепелкина М.М. Концепция числового программного управления мехатронными системами: управление электроавтоматикой станков с ЧПУ по типу виртуальных контроллеров SoftPLC// Приборы и системы. Управление, контроль, диагностика. 2003. №7. С. 5-10.
4. Знакомство с Microsoft .NET. - М.: ИТД "Русская Редакция", 2001. - 240 с.
5. Мартинов Г.М. Козак Н.В. Декомпозиция и синтез программных компонентов электроавтоматики// Приборы и системы. Управление, контроль, диагностика. 2006. №12. С. 4-11.
6. Петров И.В. Программируемые контроллеры. Стандартные языки и приемы прикладного проектирования. – М.: СОЛОН-Пресс, 2004. 256 с.

**Московский государственный  
технологический университет "СТАНКИН",  
г.Москва**