

## Практические аспекты реализации модулей открытой системы ЧПУ

к.т.н. **МАРТИНОВА Л.И.**  
д.т.н. **МАРТИНОВ Г.М.**  
ФГУП НИИАЭ, Москва



**Рассмотрена технология применения теоретических и методологических аспектов построения систем ЧПУ к конкретным задачам в системе управления.**

**Проанализирован опыт, накопленный при создании новой отечественной системы ЧПУ широкого назначения, построенной на базе персонального компьютера, обладающей открытой архитектурой и модульным программным обеспечением.**

**Ф**ункциональные возможности современных систем ЧПУ позволяют, помимо управления станком, управлять и его окружением. В число окружения входят: система хранения инструментов, робот загрузки, тележка подачи заготовок и т.д. В автомобильной промышленности системам ЧПУ отведены более сложные задачи – управление целыми участками технологических конвейеров. Несмотря на конкретную конфигурацию системы ЧПУ и ее назначение, в ней всегда присутствуют основные задачи управления: геометрическая, логическая, терминальная и диагностическая, управляющие, соответственно, формообразованием изделия, управлением электроавтоматикой, пользовательским интерфейсом и диагностированием системы [1-4].

Накопленный практический опыт («know-how») является одним из основных критериев успехов или неудач на рынке производителей систем ЧПУ.

**Реализация геометрической задачи**

Геометрическая задача состоит из трех крупных модулей: интерпретатора управляющих программ, интерполятора, модуля управления следящими приводами. Последний модуль сильно зависит от типа следящих приво-

дов и способа замыкания позиционных контуров, поэтому его рассмотрение опущено.

Наилучший вариант реализации интерпретатора состоит в его построении по типу ISO-процессора [5], поскольку такое решение обеспечивает наибольшее быстродействие и гибкость PCNC-системы (Personal Computer Numerical Control – Персональная система ЧПУ) в отношении системы команд, то есть версии языка управляющих программ ISO-7bit. Выделение «независимого» и «зависимого» уровней в объектной реализации ISO-процессора позволило выделить программные компоненты, одинаковые для всех версий языка ISO-7bit.

Интерпретация кадров управляющей программы построена на основе конвейера и выполняется за семь шагов (рис. 1). На завершающей стадии данные поступают в кольцевой буфер (позволяющий анализировать на совместимость группу соседних кадров с эквидистантной коррекцией), а окончательный результат интерпретации представлен в виде IPD-кода (InterPolator Data – код данных интерполятора).

Современные требования определяют новую (открытую) архитектуру интерполятора, в которой четко обозначены отдельные блоки. Открытый интерполятор допускает свободное наращивание алгоритмов интерполяции и произвольную их комбинацию

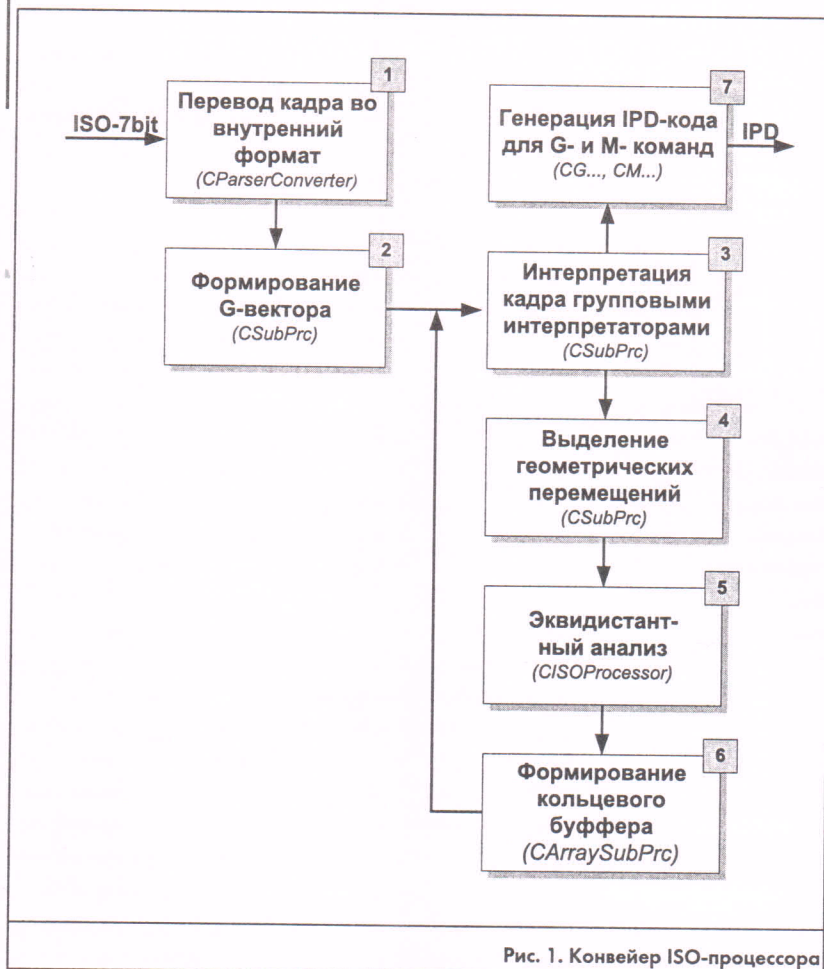


Рис. 1. Конвейер ISO-процессора

при воспроизведении сложных траекторий в многокоординатном пространстве (в том числе и с использованием сплайнов). Модуль интерполяции подключен к общей объектно-ориентированной магистрали системы ЧПУ. Интерфейс интерполятора обеспечивает: прием параметров интерполяции и оперативных сигналов управления интерполятором, выдачу данных о состоянии модуля интерполяции.

Схема интерполятора показана на рис. 2 в виде некоторого набора блоков, собственной внутренней шины и администратора. Кадры управляющей программы поступают на вход транслятора в IPD-формате, пре-

образуются во внутренний формат интерполятора, обрабатываются в блоке опережающего просмотра кадров Look Ahead (с целью сглаживания скорости подачи). Транслятор формирует сообщения, в которых упакованы параметры интерполяции. Сообщения адресуются к определенным блокам интерполятора и могут быть главными и дополнительными.

Внутренняя шина интерполятора является «шиной быстрых процессов» и связывает между собой все блоки. Она реализована на базе объектно-ориентированного подхода и соединена с основной объектно-ориентированной шиной системы ЧПУ с помо-

щью администратора. Блоки, участвующие в обработке текущего кадра, назначаются с помощью специального кода. Этот код инициализируется в трансляторе и передается в администратор. Схема кодирования обеспечивает гибкость и открытость интерполятора. Возникает возможность построения администратора, который инвариантен к составу и количеству блоков интерполятора.

Общая схема работы интерполятора выглядит следующим образом. После предоставления кванта процессорного времени администратор (построенный по схеме микропрограммного автомата) посылает запрос транслятору на получение кодов блоков интерполятора, которые должны быть запущены.

#### Реализация логической задачи управления

Применение визуальных средств программирования дает оператору инструмент для графической диалоговой разработки программы управления электроавтоматикой, а также порождает исполняемые C++ коды без компиляции в промежуточный язык. При этом изменяется (в сторону существенно большей эффективности) сама структура математического обеспечения системы ПСС (Personal Computer Controller, персональный программируемый контроллер).

Описание циклов электроавтоматики использует формализм иерархических графов. Методика описания цикла электроавтоматики включает такие этапы: этап разработки «первичного автомата», то есть автомата верхнего уровня иерархии, являюще-

гося по сути диспетчером режимов; этап разработки режима нерегулярных ситуаций (внутреннего режима), который сохраняет корректность состояния управляемого объекта при любых переключениях основных режимов, а также гарантирует неизменное состояние объекта, если цикл пассивен; этап выделения параллельно работающих автоматов, действующих в рамках цикла; этап разработки автоматов нижнего уровня иерархии.

интерфейса программиста, которая предлагает набор графических примитивов: простую вершину-состояние, сложную вершину-состояние, дугу, узел дуги. Свойства примитивов (имена, типы вершин-состояний и др.) задают в диалоговом режиме на «странице свойств» (property page). Функции визуального проектирования обеспечивают: многоуровневое вложение графов с работой на каждом уровне в отдельном окне; выполне-

другой; документирование проекта и генерацию отчетов; генерацию исходного кода для последующей компиляции; верификацию графа на уровне проектирования, моделирование и отладку циклов. Применение инструмента визуального проектирования многократно повышает производительность разработчика, позволяет создавать сложные циклы электроавтоматики, реализация которых без инструментальной поддержки проблематична.

Жизненный цикл логической задачи управления предполагает программирование, интерпретацию программы и ее исполнение. Современная тенденция состоит в упрощении первой фазы за счет визуального программирования, включая инструментальную поддержку; в объектно-ориентированной реализации второй фазы.

#### Реализация терминальной задачи

«Наполнение» терминальной задачи определяет привлекательность и конкурентоспособность системы ЧПУ на рынке. Свойства открытой системы ЧПУ развиты настолько, насколько терминальная задача поддается конфигурации и расширению. Наиболее важными разделами терминальной задачи служат: интерпретатор диалога оператора в Windows-интерфейсе, редактор управляющих программ в коде ISO-7bit, редактор-отладчик управляющих программ на языке высокого уровня.

Проектирование MMI-приложения включает: создание скелета приложения, реализацию экранов, разработку интерпрета-

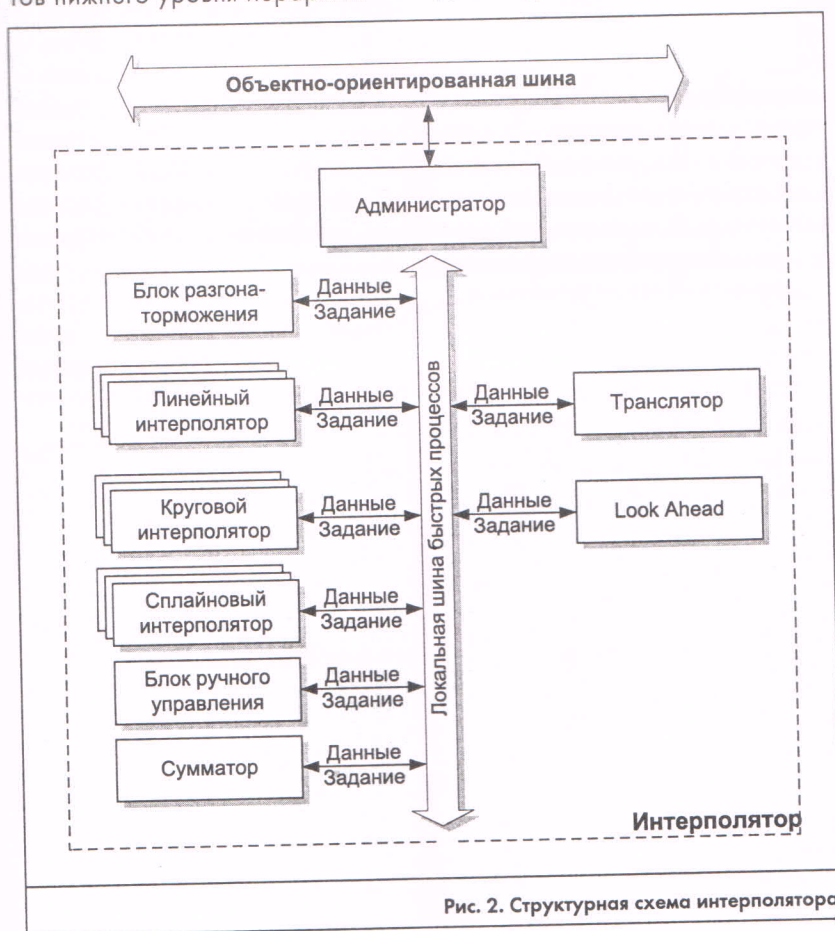


Рис. 2. Структурная схема интерполятора

Однотипность скелета исполняемого кода циклов позволила разработать инструментальную систему визуального проектирования, генерирующую исполняемые C++ исходные файлы. Конкретный граф вводят с панели

интерфейса программиста, которая предлагает набор графических примитивов: простую вершину-состояние, сложную вершину-состояние, дугу, узел дуги. Свойства примитивов (имена, типы вершин-состояний и др.) задают в диалоговом режиме на «странице свойств» (property page). Функции визуального проектирования обеспечивают: многоуровневое вложение графов с работой на каждом уровне в отдельном окне; выполне-

тора диалога; организацию информационных сессий с другими модулями системы управления.

В числе функций диалога можно обозначить: получение текущей информации о процессе управления; тестирование системы и объекта; редактирование и моделирование управляющей программы; ручной ввод и управление обработкой данных; ввод программы и автоматическое управление; управление наладочными операциями. Диалог устанавливает допустимые переходы между состояниями MMI-приложения, в рамках которых и воспроизводятся необходимые функции.

В качестве формальной модели описания диалога предложен иерархический граф состояний, вершины которого отражают устойчивые состояния MMI после нажатия оператором той или иной клавиши панели оператора, а дуги нагружены именами функциональной клавиатуры или других клавишей.

Иерархические графы удобны для описания многорежимных многоуровневых диалогов и позволяют проектировать диалог «шаг за шагом», от укрупненного выбора режимов к детальному определению поддерживаемых функций. При этом возможно описывать процесс управления не только с использованием функциональной клавиатуры, но и с помощью меню, дерева навигации и т. д.

В рамках инструментальной системы визуального проектирования задание вводят непосредственно в виде иерархического графа. В диалоговом режиме устанавливают имена состояний и свойства переходов. Глубина вложения назначается разработ-

чиком по его усмотрению, и это позволяет концентрировать внимание на текущих фрагментах диалога.

Практика показывает, что сгенерированные САМ-станции управляющей программы начинают работать на станке, как правило, после третьей редакции. При этом исправления, которые нужно внести в управляющую программу (УП), как правило, незначительные и их удобно выполнить, прежде всего, на самом станке.

Во-первых, к редактору управляющих программ предъявляют стандартные требования, характерные для текстового редактора. Во-вторых, существуют специфические требования: перенумерация после изъятия-включения кадров, изменение масштаба и размерности, вывод активных G-функций (G-вектора) на основе предыстории кадра, синтаксический и семантический контроль, диалоговый (графический) ввод кадра и параметров стандартных циклов (файлы графической помощи находятся в составе конфигурационного файла), создание управляющих программ в режиме обучения.

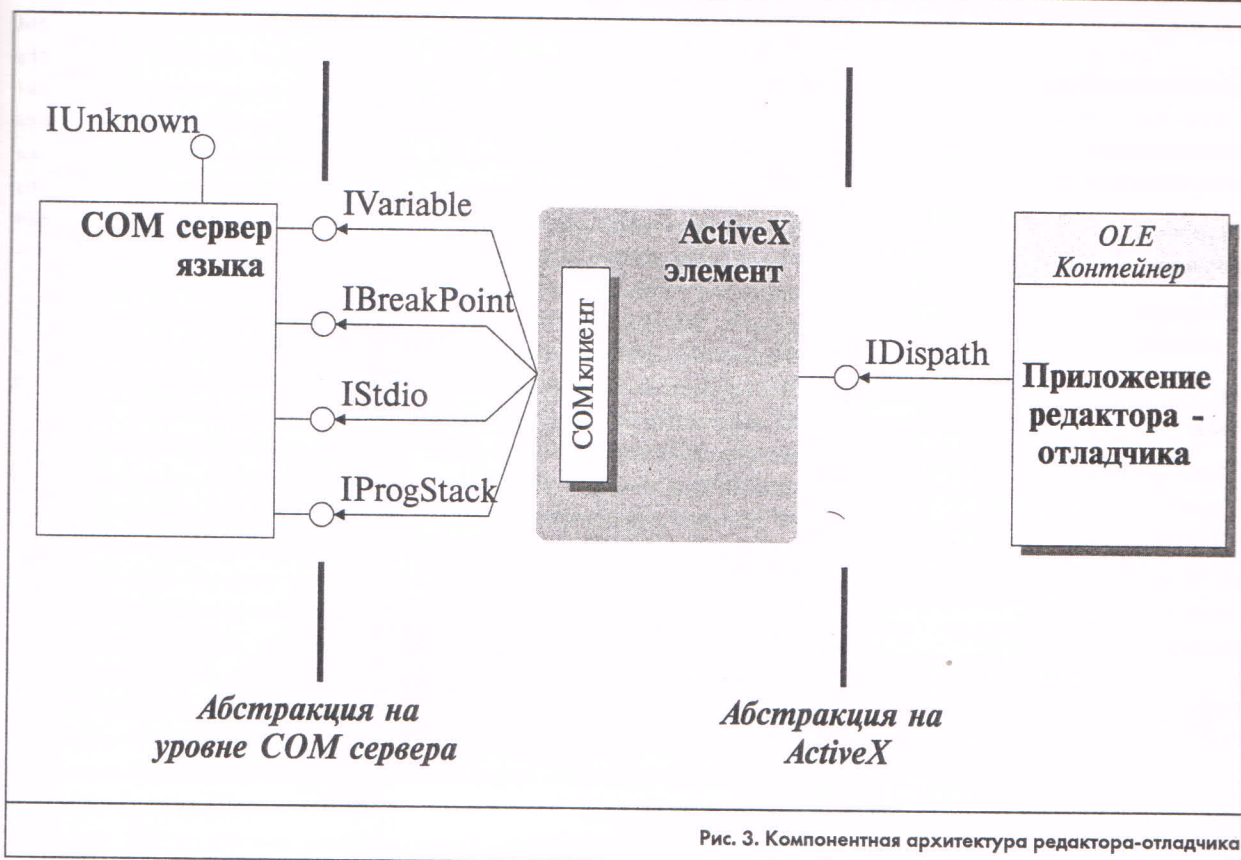
Средства отладки программ включают: пространственное графическое моделирование траектории инструмента с различием (по цвету, типу и толщине линий) быстрых и рабочих перемещений; активное использование точек останова, используемых, в том числе, для выделения фрагментов графического изображения; масштабирование графического изображения; поддержку различных режимов изображения; моделирование оставшейся части программы по отношению к текущей позиции

станка. Подобные возможности требуют включения в состав редактора некоторого ядра и дополнительных подсистем: интерпретатора управляющих программ (для любых версий кода ISO-7bit), имитатора интерполятора для рисования траекторий и сервера обучения.

Конфигуратор формализует код ISO7-bit путем выделения в нем нескольких уровней абстракции. На первом уровне определяется система команд (G-функций) и параметры каждой команды. Следующий уровень разбивает систему команд на группы по функциональному назначению G-функций и формирует G-вектор активных команд. Последний уровень абстракции назначает списки разделителей, комментариев, имен осей и адресов, имен G-функций. Подобным способом удается формализовать любую версию кода ISO 7-bit и соответствующим способом сконфигурировать редактор.

Редактор управляющих программ имеет архитектуру, открытую для конечных пользователей, открытую для разработчиков самого редактора, открытую для внешних приложений. Для конечных пользователей это, прежде всего, означает возможность конфигурации на различные версии языка ISO-7bit с помощью конфигурационного файла, имеющего текстовый формат. Далее, существует возможность конфигурировать интерфейс пользователя, включая систему контекстных подсказок и систему помощи, используя текстовый файл инициализации и соответствующие динамические библиотеки ресурсов.

Разработчикам редактора предлагается архитектура, от



крытая для интеграции и компоновки. Так, редактор может быть интегрирован в существующий интерфейс системы ЧПУ или работать в качестве независимого приложения в технологическом отделе подготовки управляющих программ.

Архитектура, открытая для внешних приложений, поддерживается интерфейсом OLE IDataObject, с помощью которого осуществляется передача данных через clipboard, – стандартный Windows-механизм для обмена данными между приложениями.

Независимо от версии, структуры всех языков управляющих программ высокого уровня однотипны: имеется основная программа и некоторый набор вызываемых подпрограмм. В теле программы представлен список пе-

ременных, которые по ходу выполнения программы меняют значения. Процесс выполнения сопровождается информационными сообщениями, предупреждениями, сообщениями об ошибках.

Базовые окна постоянно присутствуют на экране, окна со вспомогательной информацией (например, списком точек останова) реализованы как всплывающие. Основные и вспомогательные окна редактора-отладчика образуют ActiveX управляющий элемент.

Архитектура редактора-отладчика включает COM-сервер, ActiveX – управляющий элемент, приложение (рис. 3). В этой архитектуре выделены два абстрактных уровня. На первом уровне поддерживается работа с различными языками управляющих программ, причем для каж-

дого языка необходимо разработать его собственный COM-сервер. Любой COM-сервер, однако, должен располагать неизменным набором интерфейсов для работы с переменными, точками останова, буферизованными файлами и сообщениями. В этом случае ActiveX – управляющий элемент со всеми своими основными и вспомогательными окнами способен работать с любой версией языка высокого уровня управляющих программ.

Второй уровень абстракции развязывает ActiveX – управляющий элемент и механизм управления им. Это позволяет использовать ActiveX или в составе терминальной задачи системы ЧПУ, или в отдельном приложении на персональном компьютере.

Терминальная задача относится к числу наиболее сложных и

наиболее ответственных разделов системы ЧПУ при управлении мехатронными системами. Ее «скелетом» служит интерпретатор диалога оператора в Windows-интерфейсе, для разработки которого использована формальная методика, поддержанная оригинальной инструментальной системой. Для редактирования, отладки и моделирования управляющих программ предложены два конфигурируемых приложения для управляющих программ низкого и высокого уровня соответственно.

**Реализация диагностической задачи управления**

Наиболее совершенные системы ЧПУ располагают отдельным режимом диагностики, который реализован в виде программно-аппаратного комплекса и ориен-

тирован на тестирование и глубокое исследование логической и геометрической задач управления. Диагностика, как правило, выполняется «вне реального времени», что означает: измерения сохраняются в памяти, а затем анализируются. Подсистема диагностики способна конфигурировать измерения, считывать измеряемые сигналы, запоминать результаты измерений вместе с результатами конфигурации измерений, распечатывать осциллограммы измерений, считывать файлы с результатами измерений и результатами конфигурации измерений, выполнять разнообразные операции над измеренными сигналами. Для диагностики логической задачи управления служит Логический анализатор, а для диагностики геометрической задачи предназначен Осциллограф.

Под виртуальным прибором диагностики будем понимать элемент, предоставляющий результаты диагностических испытаний и создающий внешний образ измерительного устройства, например, Логического анализатора или Осциллографа.

В обобщенном виде распределенная архитектура подсистемы диагностики представлена на рис. 4. Соединение с физическим устройством (контроллером ввода-вывода, контроллером приводов и т. д.) осуществляется с помощью интерфейсной функции этого устройства. СОМ-серверы маскируют особенности физических устройств, но организуют доступ к устройствам по общему СОМ-интерфейсу. Уровень виртуальных приборов предлагает средства интерактивного конфигурирования и визуализации измерений. На прикладном уровне эти приборы встроены в приложение с доступом к приборам через OLE-интерфейс.

Электроавтоматика мехатронных систем достаточно сложна и требует высококвалифицированных специалистов при наладке и запуске оборудования в эксплуатацию. Подобные специалисты находятся обычно в удаленных сервисных бюро и занимаются дистанционным анализом входных и выходных сигналов программируемого контроллера с помощью все того же виртуального прибора, располагая конфигурацией и результатами измерений.

Оптимальная настройка регуляторов следящих приводов подачи невозможна без тщательного анализа их динамических характеристик с помощью Осциллографа подсистемы диагности-

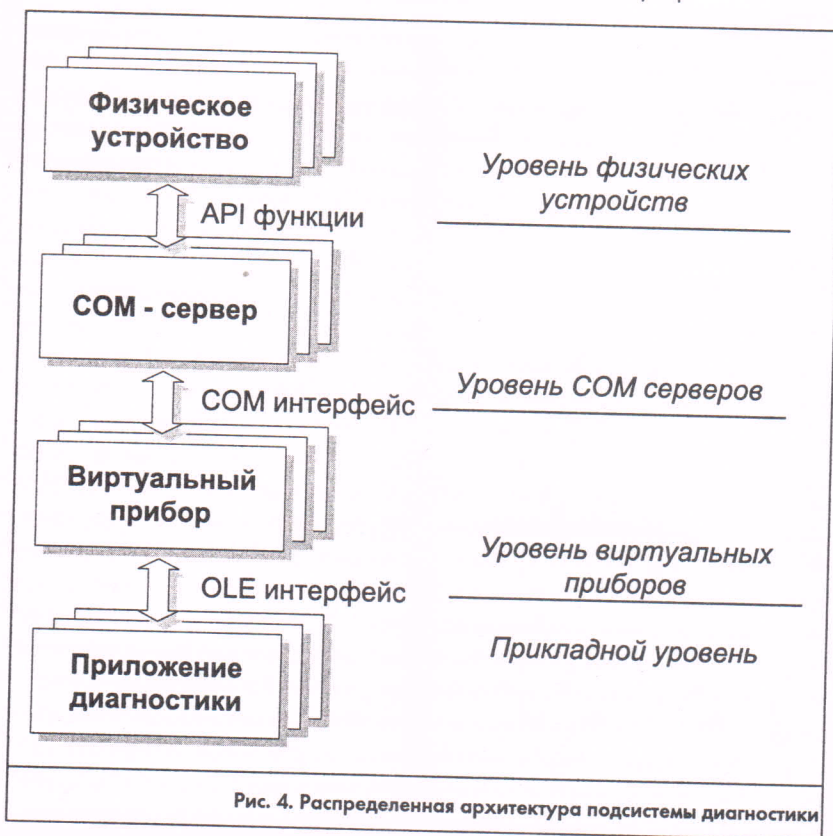


Рис. 4. Распределенная архитектура подсистемы диагностики

ки. Особенность распределенной архитектуры Осциллографа состоит в использовании «процесс-СОМ-сервера», в котором собраны все операции над сигналами, независимо от устройства – источника этих сигналов. В числе возможных операций над сигналами: масштабирование, сдвиг, практически любые математические вычисления. Помимо стандартных процедур конфигурации и отображения измерения, Осциллограф позволяет строить с помощью процесс-сервера амплитудно-частотные и фазо-частотные характеристики следующих приводов.

Современные системы управления располагают свободными ресурсами вычислительной мощности, которые должны быть использованы наиболее эффективно. В этом смысле наибольший интерес представляет создание и развитие подсистемы диагностики. В первую очередь следует диагностировать логическую и геометрическую задачи управления. Концепция виртуальных приборов, построенных по типу ActiveX – элементов, позволяет использовать разработанные средства диагностики в самых различных приложениях, которые представляют наибольший интерес для конечных пользователей. Особенности СОМ-подхода и СОМ-технологии таковы, что разработанные диагностические системы могут быть использованы в любых устройствах ЧПУ.

### Заключение

Условное разделение архитектуры интерпретатора на платформозависимый и платформонезависимый уровни позволяет выделить программные компо-

ненты, одинаковые для всех версий языка УП ISO-7bit. Архитектура построения интерполятора по типу микропроцессора инвариантна по отношению к функции интерполяции и обеспечивает параллельную работу при управлении формообразованием сложнопрофильных деталей на многокоординатных станках. Удачный выбор входных форматов интерполятора является необходимым условием для обеспечения свободного наращивания алгоритмов интерполяции и произвольной их комбинации в многокоординатном пространстве.

Описание циклов (во всех режимах, включая режим нерегулярных ситуаций) в виде иерархических графов с дополнительными средствами описания свойств (properties) состояний, дуг и узлов, предшествует применению инструментальных средств визуального программирования циклов электроавтоматики.

Коды, реализующие диалог с оператором в терминальной задаче, образуют вполне обособленный модуль, выполняющий специфичные функции и имеющий свой интерфейс. Специфика состоит, например, в проверке доступности предполагаемого перехода. Чрезвычайно сложные графы, описывающие интерпретатор диалога, становятся обозримыми при использовании концепции иерархических графов. Построение и отладка сложных интерпретаторов диалога радикально упрощается при использовании инструментальных средств разработки.

Подсистема диагностики имеет многоуровневую структуру, где виртуальные приборы обособливают верхний уровень программ-

ного обеспечения, на котором строятся пользовательские приложения. Реализация виртуальных приборов как ActiveX – элементов позволяет встраивать их в любой контейнер, функционирующий в среде MS Windows NT. Спецификация основных интерфейсов подсистемы диагностики позволяет компоновать программное обеспечение диагностики для разных систем ЧПУ. При этом меняется только реализация СОМ-сервера диагностики.

### ЛИТЕРАТУРА:

1. Сосонкин В.Л., Мартинов Г.М. Концепция числового программного управления мехатронными системами: архитектура систем типа PCNC // Мехатроника, 2000, №1, с. 26-29.
2. Мартинов Г.М. Виртуальные приборы диагностики в системе ЧПУ // Информатика-машиностроение. 1998, №4, с. 8-12.
3. Мартинов Г.М. Открытая система ЧПУ на базе общей магистрали // Автомобильная промышленность, 1997, №4, с. 31-34
4. Мартинова Л.И., Мартинов Г.М., Персональная система ЧПУ (PCNC) широкого профиля // Электрооборудование автомобилей и тракторов. Сборник научных трудов – 74. М.: НИИАЭ. 1998, с.82-88.
5. Мартинов Г.М. Реализация ISO-процессора в системы CNC Andronic фирмы ANDRON (ФРГ) //Сб. тезисов Международной научно-технической конференции «Электротехнические системы транспортных средств и робототехнических производств». – Суздаль, 1995, с. 21-22.