

ВИРТУАЛЬНЫЕ ПРИБОРЫ КОМПЛЕМЕНТАРНОГО ТИПА, АНАЛИЗАТОРЫ ПРОТОКОЛОВ И "ВИЗУАЛЬНЫЕ УМОЛЧАНИЯ"

М.И.Перцовский, М.Г.Евтихов

("Лаборатория автоматизированных систем (АС)")

На страницах журнала "Автоматизация в промышленности" (№10, 2004) была опубликована статья "Автоматизированные приборные комплексы и виртуальные приборы (ВП): из опыта внедрений "Лаборатории автоматизированных систем (АС)". Авторы были приятно удивлены неожиданно большим потоком звонков и писем читателей журнала. Мы благодарим всех, откликнувшихся на нашу публикацию, за интересную и плодотворную дискуссию. Общую заинтересованность этой темой показал и "Круглый стол" на VII Международной конференции "Цифровая обработка сигналов и ее применение". Отвечая многочисленным пожеланиям читателей, авторы предлагают новую статью, в которой постарались максимально обобщить обсуждаемую тему и дать ответы на наиболее типичные вопросы - философские и сугубо практические. В этой статье мы решились, следуя разговорному стилю завязавшихся дискуссий, приоткрыть некоторые подходы и практические "ноу-хау", используемые в "Лаборатории автоматизированных систем (АС)" при реализации ВП, автоматизированных приборных комплексов и систем лабораторной и промышленной автоматизации.

Статья касается более подробного разделения ВП на типы, а также идей их программирования. Изложение иллюстрируется реализацией ВП, дополняющего возможности осциллографа С9-8, а также обсуждается ПО TstKop - утилита, предназначенная для сопровождения драйвера контроллера канала общего пользования (КОП). Сами по себе представленные идеи элементарны, но они касаются основополагающих принципов программирования ВП и предполагают дальнейшее развитие. Статья может быть полезна аналитикам при разработке проектов систем автоматизации измерений, программистам, проектирующим ВП, и всем интересующимся проблемой ВП.

Виртуальные приборы: что это такое и с чем это едят...

ВП возникли из стремления заменить дорогостоящие измерительные приборы относительно недорогими вставными платами для компьютера, а элементы управления приборов реализовать программно в виде элементов на экранных формах [1, 2]. Такие комплексы из вставных плат, компьютера и программного продукта являются альтернативой реальным приборам, поэтому их можно назвать "конкурирующими ВП". Идея создания ВП лежит в основе популярного продукта LabView фирмы National Instrument. В LabView имеется много интересных решений, она представляет собой мощное средство изготовления ВП, основанное на оригинальной системе визуального программирования [2]. В отечественных разработках, проводившихся в Инженерном центре АСНИ в 80-90 гг., ВП рассматривались как средства создания установок для проведения физических экспериментов [1, 3]. Оказалось, что существует значительная потребность не только в конкурирующих, но и в ВП несколько иного рода. ВП, не заменяющие реальные приборы, а работающие совместно с ними и дополняющие их функциональность, можно назвать дополнительными или комплементарными.

Рассмотрим в качестве примера ВП, позволяющий управлять посредством компьютера осциллографом С9-8 и накапливать данные в виде файлов на машинных носителях. Осциллограф С9-8 — громоздкий прибор еще советского производства, но достаточно популярный и в настоящее время. Первый вариант программы, управляющей С9-8, был разработан в "Лаборатории авто-

матизированных систем (АС)" в середине 90-х годов, постепенно совершенствуется и используется в качестве типичного примера при анализе новых идей, связанных с ВП и применением КОП [1]. На рис. 1 приведена экранная форма, имитирующая внешний вид С9-8. С помощью кнопок здесь можно менять режимы работы осциллографа. Результаты измерений и режимы работы сохраняются в файлах в текстовом виде с дополнительными комментариями, облегчающими их интерпретацию. Для большинства пользователей этот простой стиль использования прибора оказывается вполне достаточным. Обратим внимание на существенную особенность рассматриваемого ВП: он является не конкурирующим, а комплементарным (дополняющим), используется только совместно с реальным С9-8 и позволяет "сделать так, чтобы результаты измерений оказались в ПК".

ВП обладают способностью, которой остро недостает многим реальным приборам — сохранять результаты измерений на машинных носителях в виде файлов. Далее эти файлы можно и подвергать обработке другими программными средствами, и пересылать коллегам, и сохранять на CD для долговременного хранения.

В теории проектирования вычислительных комплексов термином "архитектура" обозначают экономически обоснованный выбор между программной и аппаратной реализацией функций вычислительной системы как между альтернативами, требующими различных издержек [4]. Понятие "архитектура" позволяет рассматривать не только конкурирующие и компле-



Рис. 1

ментарные ВП, но и такие, у которых вообще нет аппаратной части. Однако практическое значение последних трудно переоценить. Они образуют пограничную область между ВП и обширным миром моделирования и автоматизированного проектирования. Их можно назвать моделирующими ВП. Про некоторые исследовательские программы можно сказать, что они реализуют функциональность приборов, поэтому могут быть прототипами и моделями при разработке реальных устройств. Если такие программы не содержат продуманного дизайна управляющих элементов прибора, то еще не являются моделирующими ВП, но приближаются к ним. При разработке конкурирующего или комплементарного прибора, как правило, приходится строить несколько параллельных версий программной части прибора. Если такая версия предполагает отказ от использования аппаратной части, то ее можно рассматривать как ВП моделирующего типа. Более того, удобство использования любого ВП резко возрастает, если он может использоваться как моделирующий, т.е. в условиях, когда его аппаратная часть выходит из строя или временно отсутствует. Естественная для ВП способность функционировать как моделирующий прибор (т.е. без использования аппаратной части) является еще одним важным преимуществом, которого остро недостает реальным приборам. Благодаря этой способности ВП могут служить средствами моделирования, рекламы, обучения и сопровождения, связанными с различными системами автоматизации. ВП конкурирующего типа с точки зрения архитектуры является еще не до конца вырожденным в абстрактный прибор случае, когда его аппаратная часть сведена к минимуму. ВП комплементарного типа при определенных условиях оказывается решением более близким к оптимальному в экономическом смысле. Например, идея "повязать с компьютером через КОП С9-8, оставшийся со старых времен," для многих исследовательских организаций более реалистична, чем приобрести современный прибор такого же класса и вновь оказаться перед проблемой переноса результатов измерений в компьютер. Если делить типы ВП по принципу конкурирующий — дополняющий, то следует учитывать не два, а три типа деления: моделирующие, конкурирующие и комплементарные ВП.

Типы виртуальных приборов — новый взгляд на старые проблемы

Типы ВП можно различать не только по принципу конкурирующий — дополняющий, но и по совершенно другим основаниям. Автономным ВП, являющимся отдельными программными продуктами, можно противопоставить ВП, входящие в системы автоматизации. Стоимость аппаратной части на один прибор мало зависит от числа приборов в системе, тогда как стоимость программной части на один прибор достаточно быстро убывает с увеличением числа однотипных приборов. В ситуации, когда с помощью

SCADA создается сложная система, объединяющая десятки и сотни приборов, естественен выбор однотипных приборов и сведение к минимуму их аппаратной части. SCADA и автономные ВП имеют близкие, но различающиеся области применения. SCADA уместны при автоматизации крупных производственных процессов, системы типа LabView [2] и ACTest [5] применяются для достаточно просто формализуемых лабораторных измерений. Однако система ACTest не ориентирована только на создание автономных ВП и оказывается несколько более практичной при решении технологических задач. Для автономных ВП комплементарного типа и систем из нескольких приборов, не базирующихся на SCADA, также остается достаточно обширная ниша, включающая плохо формализованные экспериментальные исследования, которые тоже имеют свои особенности. Последовательность действий при таких измерениях может многократно повторяться, но при повторениях существенны нерегулярные и неповторяющиеся участки, приводящие к плохо предсказуемым последствиям. Поэтому оказываются важны не только редактируемые сценарии измерений, но также и автоматизированное протоколирование действительно произошедших событий.

Анализаторы протоколов

Рассмотрим программные элементы особого типа — анализаторы протоколов с точки зрения использования их в ВП. Под анализатором протоколов понимается специализированный отладчик, ориентированный на отладку особым образом ограниченного сценария проведения эксперимента. Необходимость в таких отладчиках возрастает при усложнении эксперимента, но может появляться уже в достаточно простых случаях. Если процесс измерения оказывается очень быстрым или наоборот продолжительным во времени, то появляется необходимость "сделать так, чтобы само работало", возникает потребность в средствах программного управления в составе ВП, т.е. некоторой разновидности скрипта и средств отладки для него. Далее, говоря о скриптах (командных файлах), не будем учитывать тонкие различия, отличающие их от макросов и будем считать макросы и скрипты синонимами. Введение скриптов видимо составляет необходимый этап усложнения и развития многих программных продуктов. Однако необходимость в скриптах проявляется не сразу, а в тех случаях, когда использование программного продукта приобретает характер экспериментального исследования. Проблема введения скриптов решена фирмой Microsoft с помощью возможности использования со всеми продуктами пакета Office и языка Basic for Applications. SCADA-системы и БД предоставляют пользователям специализированные объектно-ориентированные языки. Разработка таких специализированных языков требует значительных усилий и затрат, которые не были бы оправданными при разра-

ботке автономных ВП. Поэтому возникает вопрос: не следует ли для ВП (и аналогичных им программных продуктов) сознательно ограничить возможности скрипта как языка программирования? В первоначальной форме макросы, появившиеся в старых версиях MsOffice, 1С, Lexicon и многих других продуктах, были весьма просты. Практичность продукта АС-Тест также связана с достаточно простым языком формирования сценариев, не требующим от пользователя программистской квалификации. В данной работе предлагается пройти в упрощении еще дальше: удалить из языка скрипта все средства формирования абстракций вплоть до переменных, а вместо них попытаться ввести какие-то иные возможности. Речь идет о том, чтобы реализовать для случая плохо формализованных экспериментов идею Э. Дейкстры о последовательной дисциплине программирования в ее первоначальной форме, высказанную им в "Заметках по структурному программированию" и гарантирующую понимание программы: "Отказавшись от большого разнообразия блочных схем и ограничившись тремя типами операторов управления, мы, тем самым, следуем некоторой последовательной дисциплине" [6]. Э. Дейкстра сохранял три типа операторов управления, поскольку ориентировал свою "дисциплину" на программистов. Ориентируясь на пользователей-экспериментаторов, нет смысла сохранять разделение операторов на типы. Предопределенные типы операторов должен вводить программист, сопровождающий систему (не исключено, что и сам пользователь возьмет на себя эту роль). Возможно, что простой линейный язык скрипта придется как-то усложнить, но эти усложнения должны выражать принципы экспериментирования, но не принцип универсального программного управления. Первой такой дополнительной возможностью, на наш взгляд, является использование протокола работы программного средства как программы для исполнителя скрипта. Если автоматически формируемый протокол работы можно корректировать и использовать как текст нового скрипта, получается анализатор протоколов — программное средство, с помощью которого можно накапливать опыт экспериментирования и добиваться при проведении эксперимента выполнения такого существенного, но иногда трудно достижимого требования, как воспроизводимость результатов.

На рис. 2 приведена экранная форма с анализатором протокола измерений для С9-8. Программа С9-8

предусматривает ведение протокола измерений в отдельном файле, отличающемся от файлов результатов измерений. Протокол одного измерения заключен между строками BEGIN и END, в него включены настройки осциллографа и пути к файлам результатов. Файл протокола вызывается в анализатор протокола, где его можно просмотреть, отредактировать, сохранить в отредактированном виде и воспроизвести измерения повторно. Воспроизвести можно как отдельные выбранные измерения, так и всю последовательность.

"Визуальные умолчания" — "ноу-хау" в концепции виртуальных приборов

Билл Гейтс в своей популярной работе "Дорога в будущее" проанализировал связь между рядом философских, психологических и социальных проблем, с одной стороны, и тенденциями развития ПК и программирования, с другой. Отметим два его замечания. Первое — об усилиях программистов сделать компьютеры более дружественными людям, о том, что еще сохранившиеся элементы неудобства в использовании компьютеров — повод к дальнейшей работе программистов. Второе — о том, что компьютер располагает к экспериментированию, поэтому склонность человека к экспериментированию лежит в основе его дружелюбности к компьютеру. Вместе с тем существуют софизмы, противопоставляющие экспериментирование и дружелюбность. "Эксперимент, эпатаж, провокация" стали чуть ли

не синонимами в искусстве. В программировании имеется механизм, весьма развитый и эффективный, пригодный, на наш взгляд, для правильного разрешения целого ряда парадоксов, связанных с социальными и моральными проблемами — это механизм умолчания (defaults). В наиболее гибкой форме умолчания реализуются как свойства и поля программных объектов. Устраняя из скриптов средства формирования абстракций вплоть до переменных, мы начинаем балансировать на грани разумного, так как удаляем наиболее привычные для программистов способы введения умолчаний. Поэтому следует обратить особое внимание на альтернативные и более архаичные подходы к умолчаниям, не связанные с классами объектов. Во многих программах типичным способом задания значений параметров по умолчанию являются вкладки, на которых можно поменять настройки программы и сохранить их в системном реестре. Главное в этом способе, на наш взгляд, смена экранных форм.



Рис. 2. Анализатор протокола С9-8. Нажатие кнопки "Выполнить" вызывает повторное выполнение выделенного измерения с автоматической настройкой параметров осциллографа и записью результатов в указанный файл. Кнопки "Вперед", "Назад" позволяют смещаться по протоколу без повторного измерения. Одновременно с выполняемыми измерениями может автоматически формироваться новый протокол

Осознав смену форм как способ введения умолчаний, можно пойти дальше и использовать с той же целью изменение размера видимой части одной экранной формы или наложение нескольких экранных форм друг на друга. Назовем умолчания, вводимые с помощью изменения размеров и смены экранных форм, визуальными умолчаниями. Визуальные умолчания – новое сочетание старых терминов, оно давно известно, широко распространено и реализуется с помощью интерфейса SDI программных продуктов. Это новое сочетание терминов подчеркивает, что при введении скриптов в программные средства следует представить пользовательский интерфейс в виде нескольких слоев. Каждый слой должен быть ориентирован на различные умонастроения пользователя, причем анализатор протоколов не должен оказываться на первом месте. Думаем, что при поиске принципов визуального выражения смены умонастроений, возникающих у экспериментирующего пользователя, программистам будут не бесполезны эстетические образцы этого выражения, созданные поэтами авангарда (например Каммингсом Э.Э. и Ниной Искренко), так же, как учет опыта Microsoft в использовании макросов. Целесообразно перемещение средств, поддерживающих программирование на скрипте, в глубину программы, куда доберется лишь достаточно опытный пользователь, и ограничение этих средств непосредственными задачами экспериментирования. В продуктах Microsoft Office вызов Basic for Applications происходит через Alt-F11, после чего происходит смена экранных форм, и это хороший и правильный пример. Однако появление в макросах MsOffice развитого языка программирования привело к появлению макровирусов и стало ощутимой угрозой безопасности информации. Устраняя из скриптов средства формирования абстракций вплоть до переменных, мы блокируем возможность написания вирусов на скрипте. Однако это блокирование еще не означает решения всех проблем безопасности информации. Содержательный эксперимент не может не породить какого-то риска, а искусство экспериментирования включает умение снижать этот риск. Если протокол можно редактировать, то его легко можно и испортить, а найти внесенную ошибку трудно. Автоматическая генерация правильных протоколов и возможность их копирования сокращают остроту принципиально неустранимой проблемы безопасности информации при проведении экспериментов.

Вернемся к программе С9-8. Пользователь достаточно быстро запоминает связь кнопок с функцио-

нальными клавишами F1...F12, поэтому часть форм с кнопками (рис. 1) становятся ненужными. Ставшая излишней часть экранной формы может быть скрыта путем изменения размера формы или же путем наложения форм (рис. 3), освободив место на дисплее для раскрытия иных возможностей программы и взаимодействия с ней других программных средств, что и является примером применения визуального умолчания.

Зачем все это нужно – от теории к практике

Вышеизложенные идеи о комплементарных ВП, анализаторах протоколов и визуальных умолчаниях были применены при построении программного продукта, не являющегося ВП, но связанного с необходимостью экспериментального исследования, полезного для сопровождения и разработки систем автоматизации на основе КОП. Приборы, подключаемые к компьютеру через интерфейс КОП, часто имеют уникальное программное управление, близкое к программированию в кодах. Причем описания этих уникальных систем программирования не отличаются подробностью. Поэтому перед началом работы над созданием системы автоматизации с использованием таких приборов оказывается необходимым экспериментальное исследование реальных возможностей и работоспособности приборов. Ситуация, как правило, усложняется и другими трудностями, тем, что программное управление самого КОП не просто, пользователь сом-

невается в возможностях своего компьютера, в работоспособности платы и своего прибора, в правильности установки драйвера и, наконец, начинает сомневаться в своей собственной программистской квалификации. Для решения этих проблем мы предлагаем программный пакет TstKop, который помогает выпутаться из этой сложной и драматической ситуации, часто возникающей у пользователей в первые дни после покупки контроллера КОП. После запуска программы TstKop появляется экранная форма, обеспечивающая детектирование контроллера КОП и поиск приборов, подключенных к нему (рис. 4). Эта минимальная проверка фиксирует работоспособность платы, компьютера, математического обеспечения и способность всего комплекса средств функционировать совместно. Когда КОП и приборы обнаружены, изменение размера окна позволяет раскрыть дополнительные возможности программы. На рис. 5 слева, внизу экрана сосредоточены кнопки, позволяющие подготовить интерфейс КОП к работе, выбрать прибор, пересылать в при-

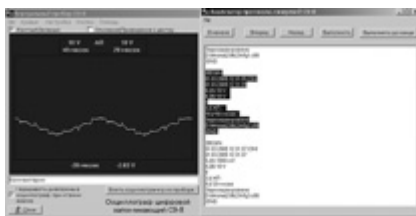


Рис. 3. Визуальное умолчание как наложение анализатора протоколов на неиспользуемую часть исходной формы ВП



Рис. 4. Основное назначение программы TstKop – определение наличия платы КОП и подключенных к ней приборов

бор и принимать из прибора информацию по одному байту. Пересылаемая информация отображается в окне "Протокол". Имеется специальное окно для отображения информации, принятой из прибора. Возможности переслать и принять небольшое число байт часто оказывается достаточным для того, чтобы понять целесообразность предполагаемой разработки системы автоматизации и сделать общее заключение о работоспособности приборов. Более подробное исследование прибора, уточнение назначения и функционирования его команд, может быть проведено с помощью анализатора протоколов (рис. 6). Содержимое протокола записанной в прибор информации можно перенести в анализатор через буфер промежуточного хранения. Строка анализатора протокола соответствует одной команде. Кроме пересылки байта в прибор и комментариев, в число команд включены чтение строк из прибора, чтение из прибора заданного числа байт, чтение байта состояния прибора. Этот набор команд далеко не исчерпывает всех возможностей КОП, но достаточен, чтобы разобраться с работой систем программирования большого числа приборов и выяснить факторы, влияющие на их работоспособность.

Исторически, программе TstKop предшествовала другая программа, позволяющая экспериментировать на уровне обращений к функциям драйвера КОП. По первоначальному замыслу необходимость экспериментов должна была бы стимулировать пользователя к освоению функций драйвера. В действительности число обращений пользователей в фирму с вопросами о КОП уменьшилось в несколько раз, после того как необходимость эксперимента с приборами и изучение функций драйвера перестали смешивать. Эксперименты с Koptst, как правило, предшествуют этапу привлечения профессиональных программистов к разработке будущей системы автоматизации, и функции обучения на этом этапе оказываются не вполне уместными. Чтобы помочь программистам в освоении функций, в поставку драйвера были включены примеры на Delphi, Builder, MsC++, MsVB, которые служат стартом для начала работы над созданием ВП, интерфейсных модулей для SCADA-систем и независимых от SCADA простых систем автоматизации лабораторных измерений. В случае сбоев и возникающих неясностях при эксплуатации и разработке систем авто-

матизации измерений на основе КОП программу TstKop с анализатором протоколов можно использовать как средство сопровождения, помогающее разобратся в трудных ситуациях.

При анализе протоколов часто встречается ошибка, приводящая к неожиданным катастрофическим результатам, но удивительно простая по своей сути. Она состоит в том, что исчезают одна или несколько строк, как правило, в начале протокола. Подпрограммы, предполагающие жестко заданный формат данных, в результате такой ошибки неожиданно начинают работать неверно. Если же возможность ошибки осознана, то удается найти приемы автоматической корректировки или блокирования последствий ошибки. Существует большое число приемов, позволяющих зафиксировать ошибки сдвига потока информации и другие сбои: контрольные суммы, мейндр, коды с коррекцией ошибок. Однако применению этого богатого арсенала предшествует драматичный момент осознания самого наличия такой проблемы, ее существования в реальной жизни и влияния на работу конкретной системы.

При переносе данных из протокола в анализатор пользователь может нечаянно пропустить несколько строк, но при достаточной предусмотрительности такой ошибки легко избежать. Более тяжелые последствия возникают, когда поток информации сдвигается из-за ненадежной работы аппаратуры или математического обеспечения, "нестыкков" между компонентами системы. Сдвиг на несколько бит вообще воспринимается как полная шифрация информации. К счастью в КОП (как и во многих других интерфейсах РС) информация передается целыми байтами или словами. Если кадр

информационного потока состоит из слов и в нем теряется или появляется лишний байт, получается эффект, схожий с шифрацией. Сдвиг информации на несколько байт в потоке данных естественен для старых приборов (таких, как С9-8), однако сдвиг может появиться где угодно. Всякий канал передачи данных характеризуется ограниченным количеством безотказно переданной информации. Анализатор протокола позволяет зафиксировать отличие потока информации от ожидаемого и заметить сдвиг в самом начале потока. Понимание неизбежности сбоев типа "сдвига потока" при достаточно больших объемах передачи информации позволяет предусмотреть необходимые тесты и модификации алгоритмов.



Рис. 5. Пересылка в выбранный прибор информации и протоколы переданных в прибор и принятые из прибора байт



Рис. 6. Успешно выполненные команды анализатора протокола переносятся в новый протокол

Заключение

ВП естественным образом обладают способностью (которой остро недостает многим реальным приборам) записать результаты изменений в виде стандартных файлов, которые можно подвергать последующей обработке другими программными средствами, пересылать коллегам, сохранять на CD для долговременного хранения. При делении видов ВП по принципу конкурирующий – дополняющий следует учитывать не два, а три члена деления: моделирующие, конкурирующие и комплементарные ВП. При проведении лабораторных измерений, приближающихся к экспериментальным исследованиям, целесообразно использовать в ВП визуальные умолчания, учитывающие изменение умонастроений пользователя в процессе исследования. В процессе проведения плохо формализованного экспериментального исследования целесообразно применение анализаторов протоколов, которые позволяют зафиксировать, видоизменить и автоматически воспроизводить процессы измерений и настройки приборов, но не позволяют реализовать вирусы. Анализаторы протоколов помогают установить соответствие реального и ожидаемого процессов измере-

ния, выявить возможные расхождения и ошибки, полезны в качестве средства сопровождения SCADA-систем и независимых от SCADA достаточно простых систем автоматизации измерений.

Список литературы

1. *Перцовский М.И., Воробьев Е.А., Евтихов М.Г.* Автоматизированные приборные комплексы и виртуальные приборы: Из опыта внедрений "Лаборатории автоматизированных систем (АС)" // Автоматизация в промышленности. 2004. №10.
2. Компания National Instruments. Технология виртуальных приборов – революционный подход к системам измерений и автоматизации // Там же. 2004. №10.
3. *Pertcovsky M., Vorobiov Ye, Kiryukhin S., Kutsevich N.* Integreted Environment for Acquisition, Processing and Control by Digital Data // The Internation Computer Systems and NetWorks, ESONE Committee, 1986. P.52.
4. *Байцгер Б.* Архитектура вычислительных комплексов, т.1. М.: Мир, 1974.
5. *Ртищев А.В.* Инструментальный интегрированный комплекс АСТест // Приборы и системы, управление, контроль, диагностика. 2002. №2.
6. *Дал У., Дейкстра Э., Хоор К.* Структурное программирование. М.: Мир, 1975.

*Перцовский Михаил Изидорович – кан. физ.-мат. наук, директор,
Евтихов Михаил Георгиевич – канд. физ.-мат. наук, ведущий специалист
"Лаборатории автоматизированных систем (АС)".*

*Контактный телефон (095) 730-36-32 (многоканальный).
E-mail: office@actech.ru, http://www.actech.ru*

МЭРА
MEASURING TESTING CONTROLLING AUTOMATION

**12-Я МЕЖДУНАРОДНАЯ
СПЕЦИАЛИЗИРОВАННАЯ ВЫСТАВКА
ИЗМЕРИТЕЛЬНОЙ ТЕХНИКИ И АВТОМАТИКИ**

СМЕРА

31 ОКТЯБРЯ – 3 НОЯБРЯ 2005 Г.

**Москва, Экспоцентр,
павильон "ФОРУМ"**

Информационная поддержка:

Тел.: (095)105-6561/62, факс: (095)248-07-34
e-mail: mera@meraexpo.ru
www.meraexpo.ru

Организаторы:

- Федеральное агентство по техническому регулированию и метрологии
- МНТО приборостроителей и метрологов
- ЕВРОЭКСПО
- Гань & Сибирян

При поддержке:

- Министерство промышленности и энергетики РФ
- ISA - The Instrumentation, Systems and Automation Society

При содействии:

- ЗАО "Экспоцентр"

Информационная поддержка:

- ПРИБОРЫ
- ТЕХНИКА
- ИЗМЕРЕНИЯ
- CONTROL ENGINEERING