

# Archestra System Platform

Простой способ интеграции систем автоматизации и информационных систем на производстве



Wonderware®  
Russia



Системная платформа **Wonderware Orchestra System Platform** предоставляет широкий набор инструментальных средств для автоматизации производственных процессов и формирования отчетности: сервер промышленных приложений **Application Server** для быстрой эффективной разработки и управления приложениями; реляционную базу данных реального времени **Historian**; информационный web-портал **Information Server** для распределения информации в сети и формирования интерфейсов пользователя. Готовые интерфейсы обеспечивают стыковку с оборудованием от

большинства известных поставщиков и изготовителей средств автоматизации.

Системная Платформа обеспечивает интеграцию приложений, связь с устройствами, архивирование производственных данных и быстрый доступ к ним, обработку алармов и событий, безопасность, централизованную диагностику и администрирование и т.п., а также позволяет производить поэтапное безрисковое внедрение решений для наращивания функционала существующих систем.

## InTouch

Свыше 800.000 пакетов InTouch установлены и работают на более чем 150.000 промышленных объектах по всему миру. InTouch обеспечивает интерфейс ко всем основным контроллерам для систем автоматизации - Rockwell, Siemens, Schneider, Unitronics и др., а также используется в системах автоматизации таких известных фирм, как ABB, Honeywell, Foxboro и других поставщиков DCS. Wonderware – единственное стандартное программное обеспечение мониторинга и формирования отчетности для всех основных систем автоматизации.



Дополнительная информация представлена на сайте [www.wonderware.ru](http://www.wonderware.ru)

В офисах компании Klinkmann Вы можете заказать демо-версию, а также 30-дневную лицензию на использование программного обеспечения Wonderware

Для визуализации процесса системная платформа Wonderware Orchestra System Platform использует удостоенное многочисленных наград программное обеспечение человеко-машинного интерфейса Wonderware InTouch.

Wonderware®  
Russia

[www.wonderware.ru](http://www.wonderware.ru)

**Санкт-Петербург**  
тел. +7 812 327 3752  
[info@wonderware.ru](mailto:info@wonderware.ru)

**Москва**  
тел. +7 495 641 1616  
[info@wonderware.ru](mailto:info@wonderware.ru)

**Екатеринбург**  
тел. +7 343 287 1919  
[info@wonderware.ru](mailto:info@wonderware.ru)

**Самара**  
тел. +7 846 273 95 85  
[info@wonderware.ru](mailto:info@wonderware.ru)

**Київ**  
тел. +38 044 495 33 40  
[info@wonderware.com.ua](mailto:info@wonderware.com.ua)

**Минск**  
тел. +375 17 2000 876  
[info@wonderware.ru](mailto:info@wonderware.ru)

Компания ЗАО "Клинкманн СПб" является официальным авторизованным дистрибьютором Wonderware на российском рынке.

# Преимущества объектно ориентированных архитектур для SCADA и систем диспетчерского управления



В статье раскрыты особенности и преимущества объектно ориентированной архитектуры, перечислены ее отличия от традиционной архитектуры на основе тегов. С помощью расчетов показано, сколько времени, труда и других ресурсов она позволяет сэкономить при построении систем промышленной автоматизации.

ЗАО «Клинкманн СПб», г. Санкт-Петербург

Используемые на большинстве современных производств системы SCADA и HMI-приложения диспетчерского управления построены на базе традиционной архитектуры с применением тегов. Однако сегодня уже широко распространена и хорошо технологически отработана объектно ориентированная архитектура, позволяющая на 80% сократить затраты по сравнению с системами на основе тегов. Расчет затрат будет пояснен и продемонстрирован ниже. Кроме того, преимуществом объектно ориентированной архитектуры является простота внедрения передовых решений на территориально распределенных предприятиях, высокая согласованность информации о производственных характеристиках, а также расширенные возможности масштабирования и обслуживания.

Объектно ориентированная архитектура программного обеспечения издавна применялась в профессиональных компьютерных системах. Сегодня она используется в приложениях управления процессами и SCADA, обеспечивая значительное сокращение затрат и преимущество при эксплуатации. В статье мы рассмотрим, что же представляет собой объектно ориентированная архитектура, как она помогает при разработке приложений SCADA и HMI, а также покажем, насколько она экономичнее по сравнению с традиционными системами на базе тегов.

## Сравнение систем, основанных на тегах, и объектно ориентированных систем

Системы на основе тегов  
С момента появления систем человеко-машинного интерфейса

(HMI) и диспетчерского управления на базе ПК доступ к данным процессов, создание скриптов, аварийная сигнализация и анализ данных осуществлялись на основе концепции тегов. В этих системах используется «плоский» список тегов со встроенными иерархией, взаимосвязями или взаимозависимостями.

Масштабные изменения баз данных теговых систем обычно выполняются извне по отношению к приложению, зачастую с применением текстового файла или таких инструментов, как Microsoft Excel®. После внесения изменений их импортируют в базу данных приложения. Повторное использование результатов разработок, как правило, осуществляется путем динамических операций или отношений клиент – сервер. При этом система создает общую графику, содержащую скрипты для переключения тегов во время исполнения приложения. Вследствие плоской структуры приложения пользователю затем приходится изменять каждый тег в системе и анализировать результаты воздействия изменений на остальную часть приложения.

Обслуживание приложений, основанных на тегах, как правило, включает в себя выполнение операций анализа и обновления отдельных тегов, то есть является очень трудоемким. Поэтому системные изменения занимают много времени и зачастую требуют привлечения внешнего персонала, что ограничивает возможности усовершенствования теговых систем.

Объектно ориентированные системы  
Истоки концепции объектно ориентированной архитектуры ле-

жат в сфере информационных технологий (ИТ). Изначально ее целью было создание инструментов, освобождающих разработчика от выполнения рутинных, однообразных задач программирования и позволяющих максимально использовать ранее созданный код с помощью стандартных программных объектов.

Естественно, эти инструменты не вполне подходят для применения в производственной среде. Во-первых, системные интеграторы и инженеры на производстве обычно не являются программистами. Во-вторых, между ИТ-системами и приложениями производственной автоматизации имеются принципиальные архитектурные различия. Например, в ИТ-приложениях доступ к базам данных обычно осуществляется через недетерминированные интерфейсы на основе форм, с помощью которых реализуются такие задачи, как дистанционное банковское обслуживание, бизнес-отчетность, управление кадрами, финансовая отчетность или отображение статической информации. В отличие от них системы диспетчерского управления, оперативного управления производством (MES) и интеллектуальные системы предприятия обеспечивают получение текущих данных о производственных процессах, выполняющих сложные вычисления для определения потоков и производственных показателей, выводят информацию на операторские дисплеи, формируют отчеты, а также сохраняют эти данные в регистраторах хронологии процессов или производственных базах данных.

Эти две области применения достаточно сильно различаются, поэто-

Таблица 1. Сравнение основанных на тегах и объектно ориентированных систем

	Объектно ориентированная архитектура		Архитектура, основанная на тегах	
	Разработка	Исполнение	Разработка	Исполнение
Структура приложений	Иерархическая – объекты создаются с применением объектно ориентированной методологии	Иерархическая – компоненты соответствуют физическим устройствам и могут осуществлять координацию с компонентами в различных компьютерах.	Иерархическая – графический контент иногда создается с применением объектно ориентированного подхода	Плоская – монолитные экземпляры ПО исполняются на одной/нескольких машинах в качестве отдельных «приложений»
Разработка графики	Осуществляется в последнюю очередь	–	Осуществляется в первую очередь	–
Скрипты	Разрабатываются в шаблонах объектов, затем внедряются в конкретное исполняемое приложение	–	Разрабатываются отдельно с последующей привязкой к графическому интерфейсу	–
Соответствие стандартам	Строгое	–	Не строгое	–
Внесение изменений в приложения	Наследуются из шаблонов объектов	Возможны распределение, замена или усовершенствование объектов	На основе графики или с применением таких средств, как Excel	Требуется перекомпилирование приложения
Представление данных	Логические конструкции, такие как физические устройства (например, клапаны и насосы) или логические устройства (например, петли PID или вычисления), представляются в виде объектов	–	С помощью тегов	–

му для промышленности требуются специально разработанные объектно ориентированные приложения. В системной платформе ArchestrA® System Platform применяется объектно ориентированная архитектура ArchestrA, предназначенная непосредственно для промышленных заказчиков, занимающихся разработкой, администрированием и обслуживанием систем управления.

В таблице 1 сравниваются объектно ориентированные и основанные на тегах системы.

**Повышение согласованности и применение проверенных решений**

В объектно ориентированных приложениях SCADA прикладные объекты содержат в себе характеристики или параметры, связанные с представляемыми ими материальными ресурсами. Например, объект «клапан» (рис. 1) может содержать в себе все события, аварийные сигналы, средства безопасности, вычисления, сбор данных, интеграции, коммуникации и скрипты, связанные с данным ресурсом.

Объекты не просто представляют производственное оборудование. Они могут включать в себя вычисления, методы доступа к базе данных, ключевые показатели эффективности (KPI), события мониторинга состояний ERP, процедуры мобильных операторов, операции технологических процедур и задачи MES. Все эти объекты можно стандартизировать и использовать во всех управляющих приложениях для соблюдения согласованности при разработке и эксплуатации системы.

Так, в управляющем приложении можно создать стандартизованный объект задания на работу и затем добавить его к любому производственному ресурсу, допустим, насосу, обеспечив согласованный и стандартизированный подход к выдаче заявок на выполнение работ.

Производства, управляемые современными системами контроля, обладают рядом общих особенностей:

- ▶ промышленными приборами и оборудованием;
- ▶ эксплуатационными процедурами;
- ▶ измерением параметров процессов;
- ▶ вычислениями;
- ▶ графическими операторскими дисплеями.

Объектно ориентированная архитектура позволяет реализовать ти-

повой подход к разработке систем, при котором функционал системы (например, перечисленные выше характеристики) можно встраивать в шаблоны объектов, дублируемые и объединяемые для формирования законченной системы управления.

Ключевым преимуществом объектно ориентированного подхода является концепция шаблонов объектов. На рис. 2 можно увидеть, каким образом шаблоны объектов обеспечивают быструю разработку системы и распространение изменений.

В первом ряду показано реплицирование шаблона объекта, представляющего диафрагменный клапан, и всех присущих ему характеристик. Реплицирование – это процесс создания исполняемых экземпляров, или компонентов, из шаблонов объектов. В следу-



Рис. 1. Шаблон объекта содержит важную информацию об аварийных сигналах, событиях, безопасности, хронологии, SCADA, сценариях, входах и выходах

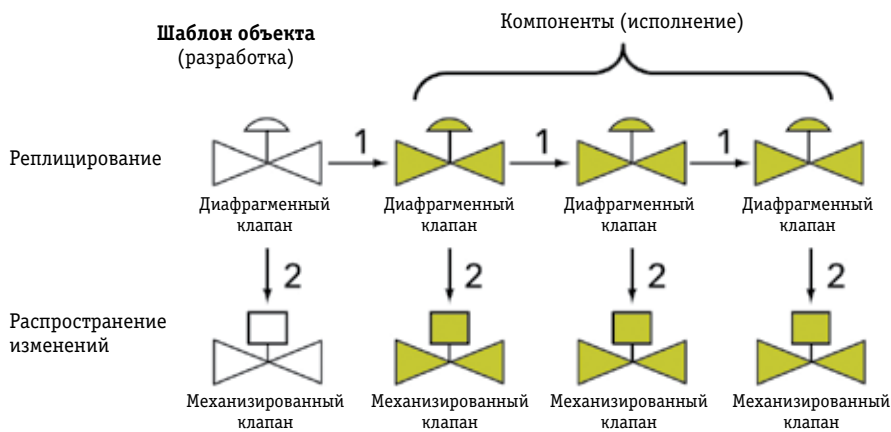


Рис. 2. Реплицирование объектов и распространение изменений

в ряду показано распространение изменений характеристик клапана (замена ручной активации на механическую) на все исполняемые экземпляры объекта «клапан».

Это отношение «родитель – потомок» является ключевым преимуществом объектно ориентированного подхода. Изменения автоматически распространяются на все исполняемые экземпляры шаблона объекта, включая любое количество физически рассредоточенных управляющих приложений. Это избавляет от необходимости выезжать на все площадки, для того чтобы внести изменения в сотни или даже тысячи стандартных ресурсов, например, клапанов. В результате:

- ▶ создание приложений оптимизируется за счет использования шаблонов объектов для автоматической генерации исполняемых компонентов;
- ▶ упрощается внесение изменений в проект за счет распространения изменений шаблона объекта на все компоненты;
- ▶ текущая модификация и модернизация систем упрощается и удешевляется благодаря автоматизированному реплицированию и распространению изменений.

#### Многочисленное использование результатов разработки

Объектно ориентированный подход, применяемый в системе ArchestrA System Platform, значительно упрощает разработку и обслуживание управляющих приложений. Интегрированная среда разработки (IDE – Integrated Deve-

lopment Environment) этого программного обеспечения позволяет использовать для создания объектов и манипулирования ими простые приемы Windows – перетаскивание (drag-and-drop), выбор щелчком (click-to-select) или заполнение текстовых полей (fill-in-the-text box).

В большинстве случаев такой подход оказывается более простым, чем построчное редактирование скриптов. При этом также сокращается количество синтаксических ошибок и ошибок времени выполнения, поскольку IDE заставляет соблюдать определенные системные правила. Кроме того, пользователи могут многократно использовать однажды разработанные шаблоны объектов в различных приложениях, что обеспечивает максимальную окупаемость разработок.

#### Объектно ориентированная графика HMI

Термин «объектно ориентированная графика» употребляется в сфере SCADA/HMI с начала 1990-х. С ее помощью пользователи могут создавать символы и затем реплицировать их в масштабах всего приложения HMI. При последующем редактировании этих символов изменения одновременно распространяются на все аналогичные символы, что упрощает работу.

Это очень полезная возможность, но приложениям SCADA/HMI требуется не только графика. Больше всего труда и времени при разработке приложений диспетчерского управления отнимают следующие функции:

- ▶ мониторинг аварийных сигналов;

- ▶ скрипты анимации;
- ▶ скрипты обеспечения безопасности;
- ▶ управляющие скрипты;
- ▶ хранение хронологических данных;
- ▶ интеграция со сторонними приложениями и базами данных;
- ▶ обнаружение событий;
- ▶ расчеты потоков и перемещения;
- ▶ интеграция оборудования;
- ▶ технологические процедуры.

Чтобы в полной мере реализовать преимущества объектно ориентированной архитектуры в системе SCADA/HMI, необходимо включить все эти функции или возможности в шаблоны объектов, в том числе графические элементы.

#### Преимущества объектно ориентированной архитектуры при разработке

##### Архитектура на основе тегов

С момента появления программного обеспечения HMI и SCADA на базе ПК пользователи создавали графические элементы операторских дисплеев и связывали их с тегами, представляющими адреса в ПЛК или в системе управления. Перечисленные ниже шаги описывают типичный процесс развития традиционных приложений SCADA на основе тегов:

1. Для разработки используется отдельный компьютер.
2. Для приложения создаются графические элементы и экраны операторских дисплеев.
3. Определения тегов импортируются из ПЛК или настраиваются вручную.
4. Для каждого тега определяются скрипты аварийных сигналов и обнаружения событий.
5. Тегам и связанным с ними входам и выходам присваиваются ссылки на графические элементы.
6. Создаются графические скрипты или ссылки для анимации.
7. Для активации новых функциональных возможностей при модернизации системы требуется закрывать приложения и вносить изменения во множество скриптов и ссылок на базу данных тегов. После этого приложение приходится переустанавливать на каждой операторской рабочей станции.

Объектно ориентированная архитектура

Объектно ориентированная архитектура, связанная с управляющими приложениями и приложениями SCADA/HMI, впервые была создана компанией Wonderware®. ArchestrA System Platform и ее основной инструмент IDE (интегрированная среда разработки) в корне изменили методы разработки приложений.

При использовании IDE создается единая модель предприятия, в которой многократно используются шаблоны объектов. Таким образом, разработчик изолируется от сложной вычислительной среды и может полностью сосредоточиться на моделировании производственных объектов. Это позволяет ему уделить основное внимание различным физическим ресурсам и производственным процессам, образующим систему диспетчерского управления всего предприятия.

После создания модели предприятия уже несложно реализовать функции контроля и управления. Скромные вложения в создание шаблонов объектов дают большой выигрыш в производительности разработок. Создание управляющего приложения с помощью ArchestrA System Platform включает в себя следующие шаги:

1. Обследование объекта для понимания структуры производственных операций или процесса.

2. Создание списка сходных компонентов оборудования или физических ресурсов. При этом также проводится разграничение областей деятельности.

3. Для каждого стандартного ресурса на объекте конфигурируются шаблоны объектов, включая графические элементы HMI. Это ключевой шаг, позволяющий зафиксировать проверенные решения и стандарты для использования во всех будущих прикладных проектах.

4. При необходимости создания сложных элементов оборудования шаблоны объектов устройств или компонентов можно включать в состав друг друга.

5. Шаблоны объектов устройств имеют атрибуты, соответствующие реальным входам и выходам в ПЛК

или в системе управления. Эти атрибуты в дальнейшем связываются с входами и выходами с помощью объектов DI (*от англ. device integration* – интеграция устройств).

6. Затем можно собрать приложение в среде IDE, просто перетаскивая объекты мышью.

7. После этого прикладные объекты включаются в группы безопасности.

8. Теперь модель предприятия, созданную в среде IDE, можно развернуть на компьютерах, предназначенных для работы приложения.

9. После завершения разработки приложения обслуживание системы уже не составляет труда. Изменения, вносимые в шаблоны объектов, могут распространяться на дочерние компоненты, функционирующие в развернутых приложениях.

**Экономия в течение всего жизненного цикла**

Объектно ориентированная архитектура обеспечивает значительную экономию затрат в течение всего своего жизненного цикла. При этом можно выделить четыре основных источника экономии, все они перечислены в таблице 2.

Приведем простой пример снижения затрат благодаря объектно ориентированному подходу.

Предположим, что для предприятия необходимо разработать управляющее приложение, в котором, помимо прочего, будет задействовано 27 двойных тарельчатых клапанов, каждый из которых имеет шесть контролируемых рабочих параметров (входных и выходных).

В ПЛК имеются точки ввода/вывода, предназначенные для измерения характеристик этого клапана.

В традиционной системе на основе тегов придется создать 162 тега (по 6 значений входных и выходных параметров на каждый из 27 клапанов). В объектно ориентированной системе SCADA создается общий шаблон объекта «клапан», а объекты, представляющие каждый отдельный клапан, являются экземплярами, или копиями, этого шаблона. Далее предположим, что для разработки приложения в традиционной, основанной на тегах, системе SCADA требуется затратить 15 минут на каждый тег. Сюда не входят временные затраты на разработку графических компонентов или логики управления ПЛК. Предположим, что разработка шаблона объекта «клапан» занимает два часа, и еще приблизительно 12% от этого времени (или 0,4 часа) отводится на каждый экземпляр объекта для настройки каждого клапана в приложении.

Тип устройства	Число экземпляров	Число входов и выходов на экземпляр
Двойной тарельчатый клапан	27	6

Напомним, что в шаблон объекта встраиваются настройки скриптов, средств безопасности, аварийных сигналов, событий, хронологии и связи устройств. В системе на основе тегов все это необходимо запрограммировать с помощью дополнительных тегов памяти. Теперь сравним общее время разработки приложения при использовании каждого из подходов.

Таблица 2. Источники экономии при использовании объектно ориентированной архитектуры

Статья экономии	Описание
Экономия затрат на начальное проектирование, связанное с разработкой приложения	Экономия достигается за счет сокращения времени разработки приложений пользователями благодаря многократному использованию единой разработанного шаблона объекта.
Экономия затрат на начальное проектирование, связанное с модернизацией приложения	Затраты на разработку сокращаются благодаря возможности распространять изменения, вносимые в шаблон объекта, на все исполняемые экземпляры, полученные из этих шаблонов. При необходимости изменения нескольких приложений в процессе разработки экономия может оказаться весьма существенной.
Экономия затрат на обслуживание в течение всего жизненного цикла системы	Использование распределенной системы значительно снижает расходы на обслуживание за счет возможности удаленного мониторинга, изменения и развертывания ПО на всех компьютерах HMI в сети. Это особенно важно для территориально распределенных сетей, поскольку дает пользователям возможность экономить время и деньги, избавляя их от необходимости выезжать на каждый объект для ремонта или модернизации.
Экономия по всем объектам	Экономия достигается за счет повторного использования созданных для одного проекта шаблонов и приложений в других проектах. Такой подход используется компаниями для стандартизации своих проектов. Это особенно полезно для системных интеграторов, реселлеров услуг с добавленной стоимостью (VAR), производителей комплектующего оборудования (OEM), машиностроительных и эксплуатационных компаний

Затраты на начальное проектирование:

Традиционный НМИ на основе тегов	Компонентная объектно ориентированная система SCADA	Экономия
162 тега × 0,4 часа на тег = 64,8 часа	(2 часа × 1 шаблон объекта) + (27 клапанов × 0,4 часа на экземпляр) = 12,8 часа	52 часа, или 80%

Впечатляющая экономия! Даже если принять половину этого значения, экономия затрат на разработку составит 40 %.

Теперь рассмотрим случай, когда необходимо внести изменения, затрагивающие 10 % приложения. При проектировании с использованием тегов логично предположить, что трудозатраты на внесение изменений составят 10 % от трудозатрат на оригинальную разработку. Однако при объектно ориентированном подходе – например, с применением Wonderware ArchestrA System Platform – эти 10 % трудозатрат потребуются применить только к шаблону объекта, благодаря наличию отношений «родитель – потомок» между объектами и их компонентами. В этой ситуации дополнительная экономия может быть рассчитана следующим образом.

Трудозатраты на доработку приложения:

Традиционный НМИ на основе тегов	Компонентная объектно ориентированная система SCADA	Экономия
64,8 часа × 10 % на изменения = 6,48 часа	2 часа на шаблон объекта × 10 % на изменения = 0,2 часа	6,28 часа, или 96 %

### Будущее – за объектно ориентированной разработкой

Объектно ориентированная архитектура демонстрирует неоспоримые преимущества при проектировании и обслуживании SCADA и систем управления. При оценке архитектуры важно учесть несколько технических аспектов.

› Обеспечивает ли средство разработки реалистичную модель заводского оборудования и производственных площадей, процессов и производственных линий?

› Можно ли легко интегрировать в приложение средства сетевой безопасности, включая централизованную настройку безопасности?

› Предлагаются ли гибкое подключение устройств и экономичные средства взаимодействия со всеми периферийными устройствами на предприятии?

› Предоставляются ли централизованные средства диагностики?

› Позволяет ли среда разработки масштабировать приложения от одиночного узла до многоузловой конфигурации без перестройки архитектуры всего приложения?

› Возможно ли удаленное развертывание приложений НМИ на компьютерах в сети?

› Поддерживает ли средство разработки единое пространство имен, обеспечивающее просмотр тегов по всей сети ПЛК, как во время работы, так и в процессе проектирования?

› Возможно ли распределение вычислительной нагрузки между несколькими компьютерами?

› Поддерживает ли система экономичное резервирование с использованием готовых коммерческих технологий виртуализации?

› Является ли подсистема аварийной сигнализации распределенной?

› Настраивается ли хронологическое архивирование в ходе разработки НМИ или для этого требуется отдельный инструмент?

Современная система SCADA должна обеспечивать решение всех перечисленных выше вопросов.

Экономические условия, в которых работают промышленные предприятия, требуют максимальной

производительности проектирования и маневренности производства. Разработка приложений управления и SCADA с применением современной объектно ориентированной архитектуры позволяет на 80 % сократить затраты по сравнению с архитектурой на основе тегов.

Решающее значение для экономии имеет возможность многократно использовать созданные объекты. Объектно ориентированная разработка позволяет использовать в приложениях проверенные решения и обеспечивает стандартизацию в масштабе предприятия. Объектно ориентированные системы можно быстро расширять или модернизировать, приспосабливаясь к изменениям рыночной конъюнктуры, что повышает маневренность. Кроме того, объектно ориентированное графическое представление процессов обеспечивает не только повторное использование результатов разработки, но и единообразный операторский интерфейс, что снижает требования к квалификации персонала и облегчает его адаптацию к изменениям в системе.

Единое «пространство имен» позволяет поддерживать и модернизировать системы управления с помощью объектно ориентированного уровня управления. Это избавляет от необходимости полностью заменять существующие системы, поскольку их возможности можно расширить, заплатив лишь небольшую часть от стоимости новой системы управления.

Таким образом, объектно ориентированная архитектура обеспечивает существенную экономию и маневренность производства, что делает ее очень привлекательной альтернативой традиционным системам на основе тегов.

Стивен Д. Гарбрехт (Steven D. Garbrecht), Invensys Operations Management, ЗАО «Клинкманн СПб», Санкт-Петербург, тел.: (812) 327-3752, e-mail: klinkmann@klinkmann.spb.ru, www.klinkmann.ru