

*На правах рукописи*

Чыонг Динь Тяу

**ИССЛЕДОВАНИЕ И РАЗРАБОТКА УНИВЕРСАЛЬНОГО СЕРВЕРА С  
ОТКРЫТОЙ АРХИТЕКТУРОЙ ДЛЯ ИНФОРМАЦИОННОЙ ИНТЕГРАЦИИ  
СИСТЕМ УПРАВЛЕНИЯ**

Специальности:

05.13.01 – Системный анализ, управление и обработка информации  
(информатика)

05.13.06 – Автоматизация и управление технологическими процессами  
и производствами (машиностроение)

**АВТОРЕФЕРАТ**

диссертации на соискание учёной степени

кандидата технических наук

Санкт-Петербург 2004

Работа выполнена на кафедре «Автоматика и вычислительная техника» в государственном образовательном учреждении высшего профессионального образования «Санкт-Петербургский государственный политехнический университет»

Научный руководитель: кандидат технических наук, доцент  
Давыдов Владимир Григорьевич

Официальные оппоненты: доктор технических наук, профессор  
Харазов Виктор Григорьевич

кандидат технических наук, доцент  
Щербина Александр Николаевич

Ведущая организация: научно-производственное предприятие  
«Буревестник»

Защита диссертации состоится 17 февраля 2005 г. в 16 часов 00 минут на заседании диссертационного совета Д 212.229.18 при государственном образовательном учреждении высшего профессионального образования «Санкт-Петербургский государственный политехнический университет» по адресу: 195251, Санкт-Петербург, Политехническая ул., 29, корпус 9, аудитория 325.

С диссертацией можно ознакомиться в фундаментальной библиотеке государственного образовательного учреждения высшего профессионального образования «Санкт-Петербургский государственный политехнический университет».

Автореферат разослан 15 января 2005 г.

Ученый секретарь диссертационного  
совета д.т.н, профессор

Шашихин В.Н.

## ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

**Актуальность темы диссертации.** Объектом исследования в диссертационной работе являются OPC (OLE for Process Control) -серверы и OPC-клиенты как средство информационной интеграции компонентов в промышленной автоматизации и как компонент человеко-машинной системы управления (СУ) процессами и производствами. Используемый в них стандарт OPC является основным промышленным стандартом взаимодействия устройств связи с объектами (УСО) и современных систем сбора данных и диспетчерского управления – Supervisory Control and Data Acquisition (SCADA). На сегодняшний день OPC-стандарт признан и поддерживается всеми ведущими фирмами-производителями SCADA-систем и оборудования промышленной автоматизации, обеспечивая их совместное функционирование. Популярным классом OPC-приложений являются OPC-серверы конкретных аппаратных устройств или, иначе, OPC-серверы доступа к данным реального времени (РВ). Далее для краткости будем называть их OPC-серверами. OPC-серверы обеспечивают предоставление информации о состоянии параметров технологического процесса (ТП) от устройств сбора данных OPC-клиентам (в т.ч. SCADA-системам). Производители оборудования автоматизации для стандартизации своих продуктов разрабатывают OPC-серверы для этих устройств так, чтобы максимально продвинуть на рынок свою аппаратуру. Аппаратура же других фирм, подключение которой к OPC-серверу не предусмотрено, в ряде случаев является более эффективной с технической и экономической точек зрения. Новые устройства сбора данных и управления (СДУ), как правило, поставляются с OPC-серверами, но существует значительная часть используемых на промышленных предприятиях устройств более ранней разработки, которые ими не снабжены. Важно и то, что достаточно большая часть промышленных предприятий, в том числе и в России, еще не применяет SCADA-системы для автоматизации и их ожидает переход к управлению на основе SCADA-систем. Таким предприятиям важно сохранить в модифицированных СУ возможность использования уже применяемых устройств более ранней разработки (их замена может быть невыгодной с финансовой точки зрения либо подобные уст-

ройства вообще нельзя заменить другими из-за их уникальности или по соображениям информационной безопасности). Поэтому весьма актуальной является *исследование и разработка универсального OPC-сервера с открытой архитектурой* (далее сервер УСОА), позволяющего интегрировать в СУ и, в частности, в SCADA-системы любое оборудование при ограниченных затратах финансовых и временных ресурсов. Этому и посвящена данная диссертационная работа.

**Целью диссертационной работы** является исследование и разработка высокопроизводительного универсального OPC-сервера с открытой архитектурой, позволяющего информационно интегрировать УСО любого вида, автоматизированные системы управления (АСУ) ТП и SCADA-системы при ограниченных финансовых и временных затратах. Цель работы достигается решением следующих *задач*:

1. Обзор и анализ компонентов промышленной автоматизации, COM/DCOM (Component Object Model/Distributed COM) -технологий, OPC-стандарта, OPC-серверов и OPC-клиентов.

2. Анализ и синтез структуры сервера УСОА и его частей. Разработка теоретических положений для обоснования принятых решений по выделению общей и специализированной частей, иерархической декомпозиции сервера, моделям базовых объектов, взаимодействию между ними и разработка сервера в целом.

3. Апробация предложенных решений на ряде типовых примеров интеграции распределенных систем управления (PCY).

**Методы исследования и используемые инструментальные средства.** В диссертации применяются методы системного анализа и принятия решений для OPC-приложений – принципы системности, иерархичности, формализма; критерии синтеза, выбор на их основе наилучших решений, методы теорий распределенных и параллельных вычислений, автоматического управления и передачи данных.

Использованы подходы и технологии получения, обработки информации и программирования – объектно-ориентированный подход к анализу, проектированию и программированию, технологии COM/DCOM, OPC-стандарты, SCADA-система GeniDAQ и диагностический OPC-клиент, спроектированный автором.

**Научная новизна работы** определяется поставленной целью:

1. Предложена и обоснована оригинальная структура сервера УСОА, содержащая *универсальную* подсистему (собственно ОРС-сервер) и *специализированные* компоненты для подключения УСО (далее компоненты УСО). Универсальная подсистема – ОРС-сервер – разрабатывается однократно и реализует наиболее объемные, сложные и ответственные функции сервера, инвариантные к используемым на нижнем уровне УСО. Компоненты УСО дорабатываются в соответствии с конкретными разновидностями применяемых на нижнем уровне УСО.

2. Предложена и обоснована иерархическая организация сервера УСОА, дополненная новым уровнем иерархии. Для всей совокупности полученных уровней иерархии в результате декомпозиции разработан *общий набор моделей базовых объектов* – объекта-элемента, объекта-диспетчера, объекта-интерфейса – и *модель взаимодействия между ними*. Определена нотация описания моделей объектов, компонентов, уровней и взаимодействия между ними, учитывающая особенности предметной области и снижающая трудоемкость разработки. Разработана *методика построения компонентов* сервера УСОА и алгоритмы их взаимодействия.

3. Для повышения быстродействия сервера УСОА в соответствии с полученной иерархической структурой предложена и использована *модель трехуровневого кэша*, применяемого для получения, обработки и хранения данных компонентов сервера УСОА.

4. Введен и обоснован *показатель универсальности* инструментальных средств разработки ОРС-серверов (далее ОРС-инструментов). Предложен и разработан новый ОРС-инструмент, обеспечивающий организацию СДУ с учетом технологических условий СУ и повышение эффективности работы за счет более полного использования ресурсов УСО и сервера УСОА.

5. Усовершенствовано *понятие тэгов* в СУ и разработаны *новые типы тэгов*, а известные тэги модифицированы в сторону *расширения* их возможностей.

**Практическая ценность** работы состоит в следующем:

1. Разработанный сервер УСОА обеспечивает подключение любого оборудо-

вания промышленной автоматизации к SCADA-системам (широкая возможность выбора оптимальных компонентов для СУ) при существенном снижении материальных, временных затрат. Использование сервера облегчает процесс модернизации СУ промышленных предприятий и информационной интеграции их подсистем.

2. Построенный сервер УСОА предоставляет возможность замены программируемых логических контроллеров (ПЛК) компонентами УСО, что экономически эффективнее.

**Реализация и внедрение результатов.** Результаты работы реализованы в виде ПО и использованы при интеграции РСУ рентгенолюминесцентными сепараторами (РЛС) и SCADA-системы (подтверждено актом об использовании, разработанный сервер приобретен НПП «Буревестник»); при опросе и управлении параметрами насосов-дозаторов масла фирмы Ismatec в РСУ промышленной установкой СПУТ-1000 для получения пироуглеродной ткани (подтверждено актом об использовании); при интеграции различных типов контроллеров со SCADA-системами в лаборатории проектирования компьютерных СУ кафедры автоматике и вычислительной техники Санкт-Петербургского государственного политехнического университета – СПбГПУ (подтверждено актом об использовании); при создании межкафедральной учебной лаборатории, возможности которой являются объединением возможностей учебных лабораторий АСУ ТП факультета технической кибернетики СПбГПУ (подтверждено актом об использовании).

**Апробация работы и публикации.** Основные результаты и положения обсуждены на научно-технических советах (НТС) и ряде конференций: НТС НПП «Буревестник» (СПб, 18 ноября 2002 г.), научной конференции студентов и аспирантов «XXXI Неделя науки СПбГПУ» (СПб, 25 – 30 ноября 2002 г.), научно-практической конференции и школе-семинаре «Формирование технической политики инновационных наукоемких технологий» (СПб, 14 – 16 июня 2003 г. и 17 – 20 июня 2004 г.) и международной научно-практической конференции «Системный анализ в проектировании и управлении» (СПб, 27 июня – 4 июля 2003 г.).

По теме диссертации опубликовано 11 печатных работ, из них – 5 статей в

центральных журналах, одно учебное пособие.

**На защиту выносятся:**

1. *Структура* сервера УСОА, содержащая универсальную, неизменяемую часть – собственно ОРС-сервер – и специализированную, изменяемую часть – компоненты УСО.

2. *Иерархическая* организация сервера УСОА. Общий для полученных уровней иерархии набор *моделей базовых объектов*, модель взаимодействия между ними и *алгоритм разработки* объектов. *Модель трехуровневого кэша*, обеспечивающая повышение быстродействия и надежности сервера УСОА.

3. *Показатель универсальности* ОРС-инструментов. Более универсальный ОРС-инструмент. Способы организации СДУ, выбираемые в зависимости от технологических условий СУ и позволяющие полнее использовать ресурсы УСО и сервера УСОА. Понятие *тэгов* в СУ, новые типы тэгов, модификация известных тэгов в сторону расширения их возможностей и способ их разработки для использования в компонентах УСО.

4. Методика разработки компонентов УСО, включая *практическую реализацию*.

**Структура и объем работы.** Диссертация состоит из введения, четырех глав, заключения, двух приложений и списка литературы из 80 наименований. Материал изложен на 217 страницах машинописного текста, содержит 59 рисунков и 25 таблиц.

### СОДЕРЖАНИЕ РАБОТЫ

Во **введении** обсуждена актуальность решаемой задачи, раскрыты научная новизна и практическая значимость результатов работы, приведены сведения о внедрении и апробации результатов, публикациях, структуре и объеме диссертации и проаннотированы основные положения, выносимые на защиту.

**Первая глава** диссертационной работы является обзорной. В ней рассмотрены типовые компоненты иерархической РСУ – SCADA-системы, УСО, ОРС-серверы, средства взаимодействия между компонентами, обобщенная структура

OPC-сервера, интерфейсы, регламентируемые OPC-стандартом и используемые во многих OPC-серверах фирм-производителей УСО. Обобщенная структура OPC-сервера содержит три уровня. На нижнем уровне находятся объекты OPCItem, каждый из которых характеризуется значением ( $v$ ), меткой времени происхождения ( $t$ ) и признаком качества и состояния ( $q$ ). На среднем уровне OPC-сервера находятся объекты OPCGroup, которые используются для объединения объектов OPCItem в группы. На верхнем уровне находится объект OPCServer, который управляет всеми объектами OPCGroup. Рассмотрены и проанализированы механизмы обмена данными между OPC-клиентом и OPC-сервером (синхронный, асинхронный, обновления и подписки).

Полученные результаты использованы в качестве отправной точки при разработке сервера УСОА.

**Вторая глава** является основной главой диссертационной работы, в которой на основе разработанных теоретических положений синтезируется сервер УСОА.

Для получения структуры сервера предложено учитывать зависимость компонентов сервера от подключаемых УСО. На этой основе разработана *обобщенная* структура сервера УСОА. В соответствии с ней сервер разделен на две части – *универсальную*, общую (фиксированную) и *специализированную* (изменяемую) части. Фиксированная часть включает большую часть разрабатываемого сервера и представляет собой сервер компонентов, который предоставляет OPC-клиентам интерфейсы доступа к данным РВ, т.е. фиксированная часть является, по существу, OPC-сервером. В общей части скрыты и решены все сложные задачи, инвариантные к конкретным УСО. OPC-сервер имеет доступ к специализированной части для реализации обмена данными с УСО. Специализированную же часть предложено построить из множества программных компонентов УСО, предназначенных для сбора данных от устройств и управления ими. В каждом компоненте УСО в качестве небольшой по размеру составной части содержится код программы, написанной программистом, работающим в прикладной области, и необходимый для доступа к конкретным физическим устройствам. Анализ результатов программирования обеих частей сервера



показал, что размер программного кода OPC-сервера (фиксированной части) занимает 80% общего объема. При этом, в соответствии с проведенными экспертными оценками, коэффициент трудоемкости разработки OPC-сервера в 2-3 раза выше, чем для компонента УСО. В итоге получено отношение трудоемкостей разработки компонента УСО и OPC-сервера 1:10. При этом важно отметить, что OPC-сервер разрабатывается *однократно*. Таким образом, предложенное решение сокращает материальные и трудовые затраты, связанные с подключением УСО, при одновременном уменьшении сроков разработки сервера.

На полученной основе в соответствии с используемым при системном анализе принципом иерархичности и предложенными автором *принципами* адаптируемости, расширяемости, квалификационно-ориентированного и предметно-ориентированного разбиения и уменьшения степени зависимости между декомпозированными компонентами разработана развернутая иерархическая структура сервера УСОА (рис. 1),

содержащая совместно со SCADA-системой четыре уровня иерархии и дополненная новым уровнем – уровнем ЦМД. Для всех уровней иерархии предложено унифицированное решение. На основе декомпозиции каждого из уровней иерархии и интеграции

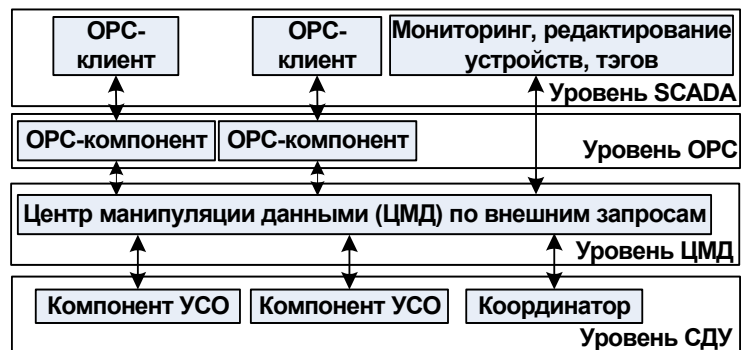


Рис. 1. Декомпозиция распределенной системы SCADA и сервера УСОА

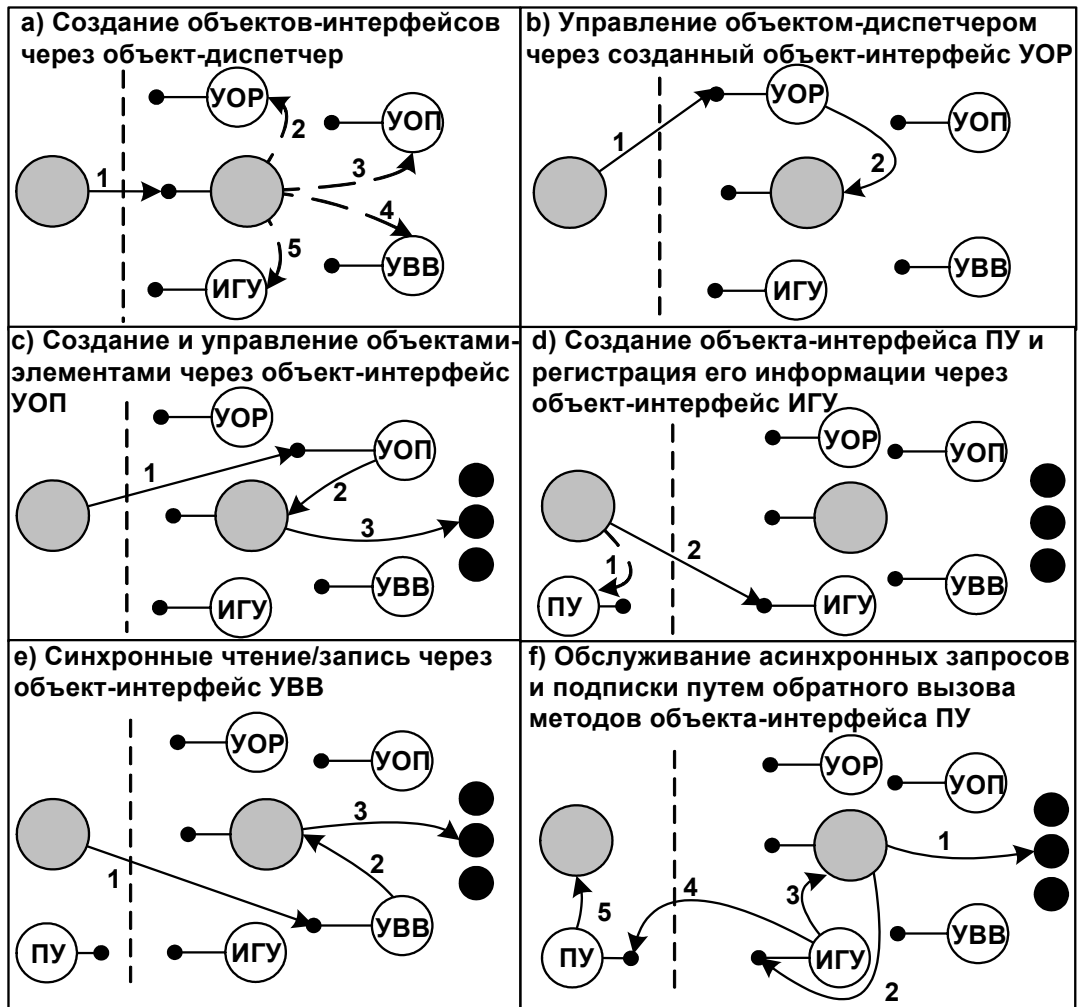
полученных результатов предложено использовать три модели базовых объектов (объекты-элементы, объекты-диспетчеры и объекты-интерфейсы). Программная реализация подобного решения хорошо поддерживается объектно-ориентированной технологией программирования (ООП). **Объект-элемент** является пассивным объектом, который может изменить свое состояние только под воздействием других объектов. Этот объект создается и управляется объектом-диспетчером (его объектом-родителем) и является терминальным объектом уровня иерархии, в котором он находится. Объект-элемент предназначен для сохранения, получения и передачи данных для другого уровня иерархии. В составе характеристик объекта-элемента имеется

хронологическая характеристика  $(v, q, t)$ . **Объект-диспетчер** является активным объектом со своим потоком управления. Он, в общем случае, автономен, т.е. может проявлять свое поведение без воздействия со стороны других объектов и является источником управляющих воздействий. Используя объект-диспетчер, можно создавать другие объекты-продукты (объекты-диспетчеры, объекты-элементы) и объекты-интерфейсы. Объект-диспетчер обрабатывает и синхронизирует запросы от объектов-интерфейсов. Особенностью объектов-диспетчеров является наличие в них таймеров, предназначенных для управления работой главных задач объекта. В объекте-диспетчере содержатся также синхронизирующие переменные и очередь. В диссертационной работе разработан *алгоритм работы рабочего потока* объекта-диспетчера для управления главными задачами и фоновой задачей объекта, ориентированный на повышение производительности сервера. **Объект-интерфейс** реализует методы интерфейса и является пассивным объектом. Он создается и уничтожается объектом другого уровня иерархии – объектом-диспетчером (его объектом-родителем). Чаще всего поддерживаются следующие объекты-интерфейсы: управление объектом-родителем (УОР), управление объектами-продуктами (УОП), управление вводом/выводом (УВВ), источник генерации уведомления (ИГУ) объекту-клиенту (объекту, находящемуся в другом уровне иерархии) и приемник уведомления (ПУ). Предложенный автором процесс создания объектов и взаимодействия между ними иллюстрирует рис. 2. Модели базовых объектов и процесс взаимодействия между ними соответствуют использованным в работе принципам предметно-ориентированного разбиения, уменьшения степени зависимости между декомпозированными компонентами и расширяемости.

В соответствии с другим основополагающим принципом системного анализа – принципом формализма – в диссертационной работе предложена оригинальная нотация, обеспечивающая удобное для данной предметной области описание уровней иерархии, компонентов, объектов сервера и взаимодействия между ними. С помощью указанной нотации для всех уровней иерархии описаны модели компонентов, объектов и модель взаимодействия между ними. На основе перечисленных моделей в ра-

боте построены конкретные компоненты трех уровней сервера и исследовано их взаимодействие.

В соответствии с иерархической структурой предложено еще одно оригинальное решение – использовать трехуровневый кэш для передачи, получения, обработки и хранения данных уровней. Это решение является одним из факторов, обеспечивших повышение



● Объект-диспетчер ○ Объект-интерфейс ● Объект-элемент ● Интерфейс  
 A - - -> B A создает B A -> B A вызывает B | Граница 2-х уровней  
 Рис. 2. Механизм работы базовых объектов и взаимодействие между ними

производительности и надежности разработанного сервера. Методы ввода/вывода объектов сервера описаны в табл. 1. Объект-интерфейс для управления вводом/выводом компонента УСО поддерживает *новые*, эффективные (быстродействие, универсальность) методы *одиночного чтения* – чтения одного значения из устройства или из первичного кэша, *одиночной записи* – записи одного значения в устройство или в первичный кэш, *множественного чтения* – чтения группы значений из устройства, *множественной записи* – записи группы значений в устройство. В диссертации выполнен анализ и приведено подробное описание реализации механизмов обмена данными – синхронного чтения/записи, асинхронного чтения/записи и режима подписки. Таким образом, предложенные в диссертации организация трех-

уровневого кэша, планирование выполнения задач и способы запуска потоков в компонентах сервера УСОА обеспечивают повышение производительности разработанного сервера и повышение надежности системы за счет рационального планирования параллельных задач и очередей запросов.

Табл. 1. Математическая интерпретация методов ввода/вывода объектов в сервере УСОА

Уровень	Объект	Ввод	Вывод	Комментария
ОРС	Объект-интерфейс УВВ	$x = r(i, s)$ , $X = mr(I, s)$	$y = w(i, z)$ , $Y = mw(I, Z)$	$r$ – чтение, $w$ – запись, $mr$ – множественное чтение, $mw$ – множественная запись,
	Объект-элемент	$x = r(s)$	$y = w(z)$	
ЦМД	Объект-интерфейс УВВ	$x = r(i, s)$	$y = w(i, z)$	$x = \{v, q, t\}$ , $z = \{v\}$ ,
	Объект-элемент	$x = r(s)$	$y = w(z)$	
СДУ	Объект-интерфейс УВВ	$x = r(i)$ , $mr(I)$	$y = w(i, z)$ , $mw(I)$	$y = \{q, t\}$ , $s$ – источник данных, $i$ – идентификатор объекта, $X, Y, Z, I$ – векторы из $x, y, z, i$
	Объект-диспетчер	$mr(I)$	$mw(I)$	
	Объект-элемент	$x = r(s)$	$y = w(s, z)$	

В заключительной части второй главы на основе анализа взаимосвязей между базовыми объектами каждого из уровней иерархии сервера предложены следующие этапы разработки его компонентов, направленные на использование ООТП: формирование иерархической структуры объектов на соответствующем уровне иерархии и взаимодействия между ними (1); выделение в соответствии с синтезированной структурой отдельных классов объектов – объектов-элементов, объектов-диспетчеров и объектов-интерфейсов (2); создание атрибутов и методов выделенных классов, сначала для класса объектов-элементов (3); затем для класса объектов-диспетчеров (4); создание интерфейсов для класса объектов-диспетчеров (5); добавление методов в классах объектов-диспетчеров, предназначенных для реализации созданных интерфейсов (6); реализация интерфейсов на основе созданных на шаге 6 методов (7). В диссертации выполнена полная разработка наиболее трудоемкой общей, фиксированной части сервера, включающая его однократное программирование и отладку.

Качество разработанного сервера УСОА для конкретных УСО зависит не только от фиксированной части сервера, но и от реализации (модификации) компонентов УСО программистом, работающим в прикладной области. В связи с этим **третья глава** диссертационной работы посвящена разработке компонентов УСО.

Для разработки компонентов УСО в диссертации предложен ОРС-инструмент, отличительной особенностью которого являются универсальность (возможность подключать любые УСО) при снижении финансовых и временных затрат. Для разработки ОРС-серверов или их частей, в принципе, можно использовать существующие ОРС-инструменты, наиболее представительными из которых являются ОРС-инструменты корпорации WINPASO (США) и Fastwel (Россия). Для сравнения предложенного ОРС-инструмента с аналогами введены понятия - закрытый и открытый модули. *Закрытый модуль* – готовая программа, поставляемая производителем (исходный код недоступен). *Открытый модуль* – программа, написанная программистом, для которой исходный код доступен. Примером закрытого модуля является разработанная в предыдущей главе общая, фиксированная часть сервера, а примерами открытых модулей являются разработанные в диссертации компоненты УСО. Для сравнения предложенного ОРС-инструмента с его аналогами автор предложил использовать характеристику (показатель) универсальности вида  $U = k_{\alpha} * \alpha + k_{\beta} * \beta + k_{\gamma} * \gamma + k_{\delta} * \delta + k_{\varepsilon} * \varepsilon$  где  $\alpha$  – степень независимости тэгов открытого и закрытого модулей,  $\beta$  – степень упорядоченности списка тэгов закрытого модуля;  $\gamma$  – степень изоляции тэгов открытого модуля от закрытого модуля,  $\delta$  – степень взаимодействия сервера с УСО,  $\varepsilon$  – степень взаимодействия УСО и тэгов с оператором. Здесь  $k_{\alpha}, k_{\beta}, k_{\gamma}, k_{\delta}, k_{\varepsilon}$  – весовые коэффициенты. Большее значение показателя универсальности  $U$  является предпочтительным и соответствует более универсальному решению. На основе выполненного анализа в диссертации получены значения показателя универсальности для предложенного ОРС-инструмента  $U_1 = 1.00$ , а для аналогов лишь  $U_2 = 0.31$  (ОРС-инструмент Fastwel) и  $U_3 = 0.75$  (ОРС-инструмента WINPASO). Таким образом, в соответствии с критерием принятия решения  $\max\{U_i\}, i = \overline{1,3}$  обоснованно выбран предложенный в диссертационной работе ОРС-инструмент. Использование этого инструмента, наряду с возможностью подключения любых УСО и обеспечением быстрой разработки компонента УСО, предоставляет более широкую свободу реализации компонента УСО, его

адаптации к конкретному устройству и возможность конфигурирования или интерактивного взаимодействия пользователя с УСО и тэгами.

В соответствии с архитектурной концепцией компонент УСО является компонентом сервера, содержащим объект-диспетчер, объект-элемент и объекты-интерфейсы. С целью построения эффективных классов объектов-элементов на основе анализа требований SCADA-систем и OPC-стандарта *расширено понятие тэга* (абстракция параметра ТП). В работе предложено различать следующие типы тэгов – аналогового ввода (модифицирован), аналогового вывода (модифицирован), дискретного ввода (модифицирован), дискретного вывода (модифицирован), тревоги и события (новый тип), тренда (новый тип), алгоритмического (новый тип), конфигурирования тэга (новый тип), конфигурирования устройства (новый тип), состояния устройства (новый тип) и технического параметра устройства (новый тип). Подобный прием обеспечивает эффективную обработку, хранение, передачу данных параметров ТП и управления ими в системе SCADA – сервер УСОА. На этой основе построены различные классы объектов-элементов с использованием ООТП, выполнен анализ и предложена методика разработки классов объекта-диспетчера и объектов-интерфейсов, представляющих собой средство общения между OPC-сервером и компонентом УСО. Классы объектов-элементов и объекта-диспетчера спроектированы как расширяемые классы (свойство *расширяемости* сервера). Последнее очень важно, так как эти классы подлежат модификации при выполнении привязки к конкретному УСО или конкретным параметрам ТП.

Предложенные в диссертации решения обеспечивают модификацию параметров устройств (объектов-диспетчеров компонентов УСО), характеристик тэгов, добавление, удаление и модификацию тэгов, выбор методов ввода/вывода, наиболее подходящих для конкретных УСО *во время работы системы* (свойство *универсальности* сервера УСОА).

В диссертации предложены способы организации СДУ в компоненте УСО, выбираемые в зависимости от условий работы СУ, и алгоритмы работы задач компонента УСО – задачи обновления первичных тэгов и задачи заполнения трендов

тэгов. Проанализированы три способа задания периода  $T_3$  запуска задачи обновления первичных тэгов – фиксация  $T_3$ , ручное задание  $T_3$  оператором и динамическая коррекция  $T_3$  во время работы в зависимости от интенсивности работы сервера. На рис. 3 приведены результаты физических экспериментов для различных алгоритмов динамической коррекции  $T_3$ .

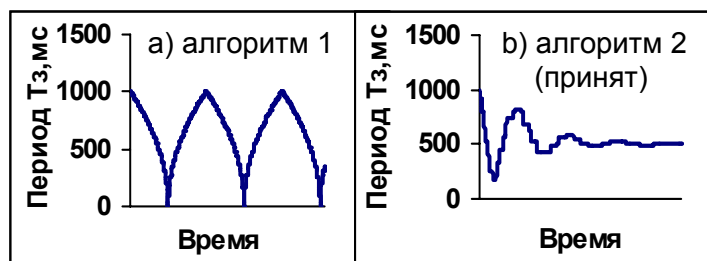


Рис. 3. Изменение  $T_3$  при разных алгоритмах его динамической коррекции

**Четвертая глава** диссертационной работы посвящена изложению методики подключения устройств УСО к

разработанному серверу и экспериментальной проверке предложенных в работе решений на представительной совокупности примеров. Методика подключения устройств УСО к серверу состоит в последовательном выполнении следующих этапов – анализ и определение физической конфигурации УСО (1), определение и изучение методов доступа к данным УСО (2), определение характеристик и требований к СУ (3), исследование и разработка классов тэгов каждого УСО (4), исследование и разработка классов устройств компонентов УСО (5), назначение способов чтения/записи методов интерфейса ввода/вывода (6) и интеграция компонентов УСО в состав сервера (7). Реализация этапов 4-6 базируется на использовании ООП. При этом возможны следующие варианты реализации этих этапов, ориентированные на сокращение трудоемкости и времени разработки: для простых случаев – модификация подходящих классов из имеющейся библиотеки классов (пополняемая библиотека) в сторону сужения их возможностей; для более сложных случаев – использование наследования абстрактных базовых классов и методов (модифицируемые методы перегружаются в производном классе, неиспользуемые – удаляются из базового класса, а отсутствующие – добавляются в производный класс).

Для экспериментальной проверки предложенной методики компоненты УСО разработаны для представительной группы устройств – эмулятора сигналов, нестандартного устройства связи с объектом с платами АЦП и ЦАП и промышленного контролера типа 7188E2 фирмы ICP DAS и фирменных модулей. Для комплексной

экспериментальной апробации предложенных в диссертационной работе решений на основе разработанного сервера выполнены интеграция PCY PJC (НПП «Буревестник») и двух типов SCADA-систем, а также интеграция двух учебных лабораторий АСУ ТП (СПбГПУ) и SCADA-системы GeniDAQ. При экспериментальной проверке интеграции учебных лабораторий, функционирующими в разных локальных вычислительных сетях, выполнено планирование эксперимента и на его основе получена оценка величины запаздывания передачи данных тэгов между разработанными серверами, расположенными в разных лабораториях – величина запаздывания не превышала 600-700 мс в наихудшем случае при погрешности измерения не более 10 мс. Указанное значение запаздывания приемлемо для мониторинга в человеко-машинной системе.

В заключительной части главы выполнен сравнительный анализ производительности разработанного автором сервера с его аналогом – OPC-сервером фирмы Fastwel (рис. 4). Эксперимент был спланирован таким образом, что в обоих случаях использовались одинаковые инструментальные средства (ЭВМ, ПО, УСО). Результа-

ты эксперимента показали, что в СУ со средним и большим количеством параметров ТП разработанный сервер имеет производительность в 5-10 раз бо-

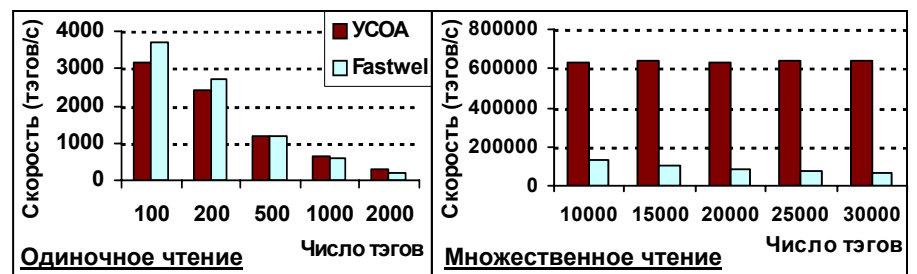


Рис. 4. Сравнение производительности сервера USOA и универсального OPC-сервера фирмы Fastwel

лее высокую, чем OPC-сервер фирмы Fastwel, а при небольшом количестве параметров ТП они имеют практически одинаковую производительность. По сравнению с OPC-сервером фирмы Fastwel предложенный автором сервер позволяет подключать любые устройства УСО при меньших финансовых и временных затратах.

В **заключении** перечислены основные результаты и изложены выводы по диссертационной работе.

В **приложениях** представлены программы эмулятора сигналов и реализация классов компонентов УСО для эмулятора и нестандартного УСО.



## ОСНОВНЫЕ РЕЗУЛЬТАТЫ ДИССЕРТАЦИОННОЙ РАБОТЫ

1. Структура сервера УСОА для информационной интеграции компонентов СУ промышленной автоматизации, содержащая универсальную подсистему, инвариантную к используемым на нижнем уровне УСО, и специализированные компоненты УСО. Предложенный сервер обеспечивает подключение как вновь разрабатываемых, так и существующих УСО к SCADA-системам, позволяет снизить материальные и трудовые затраты на подключение устройств автоматизации, повысить производительность сервера в 5-10 раз для задач средней и большой размерности при одновременном повышении надежности работы.

2. Иерархическая организация сервера УСОА, дополненная новым уровнем иерархии. Модели базовых объектов и взаимодействия между ними, модель трехуровневого кэша. Нотация описания моделей объектов, компонентов, уровней и взаимодействия между ними, учитывающая особенности предметной области. Методика использования моделей базовых объектов для построения компонентов сервера.

3. Новый ОРС-инструмент, обеспечивающий организацию СДУ с учетом технологических условий СУ и повышение эффективности работы за счет более полного использования ресурсов УСО и сервера УСОА. Новые типы тэгов и модификация известных тэгов в сторону расширения их возможностей.

4. Спланирована и проведена экспериментальная проверка сервера УСОА на представительной совокупности практически значимых примеров, подтвержденная четырьмя актами использования из трех организаций.

## ПУБЛИКАЦИИ ПО ТЕМЕ ДИССЕРТАЦИИ

1. Чыонг Д.Т. Взаимодействие открытых систем промышленной автоматизации – состояние и проблемы // Информационно-управляющие системы. 2003. №2-3. С. 52-57.

2. Давыдов В.Г., Чыонг Д.Т. ОРС-серверы с открытой архитектурой – средства взаимодействия компонентов в промышленной автоматизации // Автоматизация в промышленности. 2003. №7. С. 10-15.

3. Чыонг Д.Т., Давыдов В.Г. Организация обмена данными между OPC-приложениями // Автоматизация в промышленности. 2003. №10. С. 23-27.
4. Чыонг Д.Т., Давыдов В.Г. Метод подключения нестандартных контроллеров к современным SCADA-системам // Материалы межвузовской конференции «XXXI недели науки СПбГПУ». СПб.: СПбГПУ, 2003. С. 87-88.
5. Чыонг Д.Т., Давыдов В.Г. OPC-серверы – средства организации взаимодействия между компонентами в промышленной автоматизации // Труды VII Междунар. науч.-техн. конф. «Системный анализ в проектировании». СПб.: СПбГПУ, 2003. С. 548-557.
6. Чыонг Д.Т., Давыдов В.Г. OPC-сервер с открытой архитектурой как основа информационной интеграции сложных систем автоматизации // Материалы науч.-практ. конф. «Формирование технической политики инновационных наукоемких технологий». СПб.: СПбГПУ, 2003. С. 424-430.
7. Давыдов В.Г., Чыонг Д.Т. Проектирование компьютерных систем управления на основе SCADA-систем: Учеб. пособие. СПб.: СПбГПУ, 2004. 102 с.
8. Владимиров Е.Н., Волк Е.Б., Давыдов В.Г., Морозов В.Г., Таткин Л.З., Чыонг Д.Т. Интеграция системы автоматического управления рентгенолюминесцентными сепараторами и SCADA-систем // Автоматизация в промышленности. 2004. №9. С. 19-23.
9. Чыонг Д.Т., Давыдов В.Г. OPC-сервер и организация обмена данными между OPC-приложениями // Научно-технические ведомости СПбГПУ, 2004. №1. С. 194-202.
10. Чыонг Д.Т. Интеграция учебных лабораторий АСУ ТП // Материалы науч.-практ. конф. «Формирование технической политики инновационных наукоемких технологий». СПб.: СПбГПУ, 2004. С. 376-382.
11. Давыдов В.Г., Чыонг Д.Т. Непосредственный обмен данными между несколькими клиентами и общей аппаратурой // Сборник научных трудов аспирантов и молодых ученых факультета технической кибернетики СПбГПУ. СПб.: СПбГПУ, 2004. С. 88-95.