

УДК 681.325.5

ВЗАИМОДЕЙСТВИЕ ОТКРЫТЫХ СИСТЕМ ПРОМЫШЛЕННОЙ АВТОМАТИЗАЦИИ: СОСТОЯНИЕ И ПРОБЛЕМЫ

Чыонг Динь Тяу,

аспирант

Санкт-Петербургский государственный
политехнический университет

Рассматривается технология OPC (OLE for Process Control) — основной стандарт взаимодействия средств современных систем сбора данных и диспетчерского управления – Supervisory Control and Data Acquisition (SCADA), затрагиваются механизм обмена данными между OPC-приложениями и вопрос производительности OPC-серверов, OPC сравнивается с другими стандартами, перечисляются преимущества и недостатки стандарта. В качестве задачи, подлежащей решению, отмечается создание универсального OPC-сервера.

This article examines the OLE for Process Control (OPC) technology — the basic standard of systems interaction in modern Supervisory Control and Data Acquisition (SCADA) system. The mechanism of data communication between OPC-applications and performance of OPC servers are mentioned. The article compares OPC by other standards. The advantages and disadvantages of the standard are listed. As a task subject to the decision, the creation universal OPC server is marked.

Введение

В настоящее время информационная автоматизированная система управления промышленным предприятием имеет открытую архитектуру (рис. 1), включающую следующие неотъемлемые уровни [1].

1. Нижний уровень, который включает полевые шины и отдельные контроллеры (Field Management), представляющие собой «интеллектуальные» технологические устройства: датчики, контроллеры, механизмы и т. д. Поток информации от нижнего уровня должен быть предоставлен пользователю и всем приложениям верхнего уровня, использующим ее посредством цифровых коммуникационных протоколов связи. При этом в системе не должно возникать проблем несовместимости.

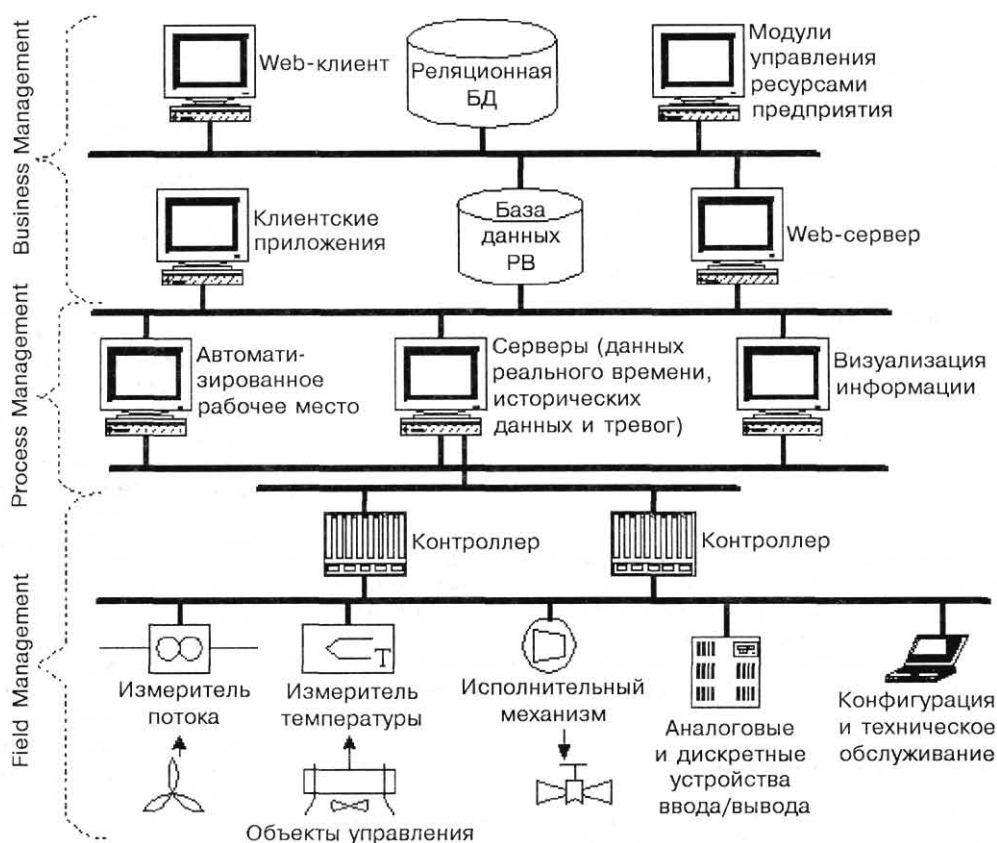
2. Уровень АСУТП (Process Management) — уровень работы систем типа SCADA (систем для сбора данных и диспетчерского управления технологическими процессами на производстве). Этот уровень обеспечивает вторичную обработку данных, которые получены с нижнего уровня, сохранение данных и их доступность приложениям и пользователям верхнего уровня.

3. Уровень АСУП — уровень приложений управления ресурсами предприятия (Business Management). Информация с уровня АСУТП должна быть доступной для уровня АСУП, т. е. доступ к данной информации с помощью прикладных программ не должен вызывать проблемы несовместимости.

Для обеспечения совместимости между уровнями и создания эффективной интегрированной системы управления предприятием системный интегратор или разработчик АСУТП должен извлекать данные технологического процесса в реальном времени с самого нижнего уровня и выстраивать «прозрачный» путь получаемым данным к самым верхним уровням. Чтобы получить систему, отвечающую всем требованиям заказчика, системному интегратору или разработчику необходимо использовать инструментальные средства управления различных уровней — SCADA-пакеты, базы данных, электронные таблицы. Ключ к этому — открытая и эффективная коммуникационная архитектура взаимодействия между приложениями, которую предлагает стандарт OPC.

Что такое OPC?

OPC — это технология связывания и внедрения объектов для систем промышленной автоматизации, предназначенная для обеспечения универсального механизма обмена данными между датчиками, исполнительными механизмами, контроллерами, устройствами связи с объектом, системами представления технологической информации, оперативного диспетчерского управления, а также системами управления базами данных [1, 2, 3].



■ Рис. 1. Архитектура информационной автоматизированной системы управления промышленным предприятием

На сегодняшний день технология OPC в определенной степени реализована и продолжает развиваться. Консорциум «OPC Foundation» пытается охватить все аспекты, связанные с взаимодействием между компонентами программного обеспечения, между программным обеспечением и между системами типа SCADA и технологическим оборудованием. «OPC Foundation» привлекает к разработке спецификаций ведущих производителей оборудования и систем автоматизации. Существует много спецификаций OPC — Data Access (доступ к данным реального времени), Alarms & Events (обработка тревог и событий), Historical Data Access (доступ к историческим данным) и т. д. Поэтому Д. Теркель [4] определяет OPC как стандарт взаимодействия между программными компонентами сбора данных и управления, основанный на объектной модели COM/DCOM фирмы Microsoft. Через интерфейсы OPC одни приложения могут читать или записывать данные в другие приложения, обмениваться событиями, оповещать друг друга о нештатных ситуациях, осуществлять доступ к данным, зарегистрированным в архивах. Эти приложения могут как располагаться на одном компьютере, так и быть распределенными в сети. При этом, независимо от фирмы-поставщика, стандарт OPC, признанный и поддерживаемый всеми ведущими фирмами-производителями SCADA-систем

и оборудования, обеспечит их совместное функционирование.

Популярный класс OPC-приложений представляют собой OPC-серверы конкретных аппаратных устройств, обеспечивающие предоставление информации о состоянии параметров многочисленных устройств технологического процесса, полученной OPC-приложением, OPC-клиентам на локальном компьютере или в компьютерной сети. OPC-сервер создает свою абстракцию устройств (это загружаемый модуль, предоставляющий интерфейс нижнего уровня с оборудованием, на котором исполняется OPC-сервер). OPC-сервер скрывает аппаратно-зависимые детали, такие как интерфейс ввода/вывода, контроллеры прерываний и механизм коммуникации в многопроцессорных системах — любые функции, зависящие от конкретной архитектуры и машин, — позволяя любому OPC-клиенту записывать и считывать данные с устройства. Устройство, для которого есть OPC-сервер, может использоваться вместе с любой современной SCADA-системой.

Современные SCADA-системы не ограничивают выбор аппаратных средств нижнего уровня, так как SCADA-пакеты предоставляют большой набор драйверов или серверов ввода/вывода (в том числе OPC-серверы) и имеют хорошо развитые средства создания собственных программных модулей или драйверов для новых и нестандартных уст-

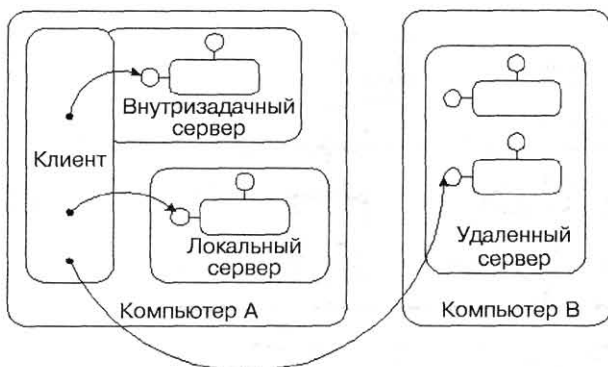


Рис. 2. Различные серверы COM-объектов

роиств нижнего уровня. Производители аппаратных средств нижнего уровня используют спецификации OPC, имеют возможность разрабатывать свой OPC-сервер для обеспечения доступа к данным реального времени и передачи данных приложениям-клиентам различных производителей программного обеспечения для промышленной автоматизации (в том числе SCADA-систем).

Механизм обмена данными OPC

Технология OPC основана на модели распределенных компонентных объектов DCOM (Distributed COM). Модель представляет собой сетевое расширение COM или распределенную COM (Component Object Model). Технология COM — модель компонентных объектов — создана Microsoft для совместного использования различных офисных приложений в Windows, например, электронных таблиц Excel, редактора Word и т. п.

COM поддерживает модель «клиент—сервер». Объекты, называемые серверами, предоставляют некие функции в распоряжение объектов, называемых клиентами. Серверы всегда являются COM-объектами, т. е. объектами, которые подчиняются спецификации COM. С другой стороны, клиенты могут быть COM-объектами или не быть таковыми. Это значит, что некоторые объекты могут быть простыми объектами C++, приложениями Visual Basic и т. п. [5]. Каждый COM-объект существует внутри конкретного сервера. Этот сервер содержит программный код реализации операций, а также данные активного COM-объекта. Один сервер может обеспечивать несколько объектов и даже несколько COM-классов. Как показано на рис. 2, используются три типа серверов [6]:

1) внутризадачный сервер (in-process) — объекты находятся в динамически подключаемой библиотеке и, следовательно, выполняются в том же процессе, что и клиент;

2) локальный сервер (local) — объекты находятся в отдельном процессе, выполняются в том же компьютере, что и клиент;

3) удаленный сервер (remote) — объекты находятся в DLL или в отдельном процессе, который расположен на удаленном от клиента компьютере.

Клиенты и локальные серверы находятся в различных адресных пространствах. OPC берет на себя передачу данных между ними. В случае внутризадачных серверов родительский процесс загружает DDL-файл, содержащий COM-сервер. Это означает, что внутризадачный сервер находится в том же адресном пространстве, что и вызвавший его процесс. В случае DCOM вызов любой функции объекта перехватывается специальным агентом-посредником, называемым Proxy Manager или Stub Manager. Proxy Manager получает все запросы на транспортировку и определяет, стандартные ли данные запрашивает или предлагает клиент. Если это так, Proxy Manager передает запрос подходящему Proxy-объекту. Подходящие Proxy и Stub — это объекты, предназначенные для транспортировки данного интерфейса и/или типа данных. Аналогичный процесс происходит на стороне Stub Manager (рис. 3). Встроенные Proxy и Stub называются транспортировщиками интерфейса.

Популярным вариантом OPC-клиента является SCADA-система. С помощью OPC-серверов SCADA-система может обращаться к любому аппаратному устройству (рис. 4) путем вызова определенных функций OPC-сервера, подписываться на получение определенных данных какого-либо канала или устройства с какой-либо частотой. В свою очередь, OPC-сервер опрашивает аппаратное устройство, вызывает известные функции клиента, подтверждает получение данных и посылает клиенту требуемые данные.

Обмен данными между OPC-клиентом и OPC-сервером может быть реализован в трех режимах: 1) синхронного чтения/записи; 2) асинхронного чтения/записи; 3) в режиме подписки (только чтение).

При синхронном чтении/записи клиент посылает серверу запрос со списком интересующих его переменных и ждет, когда сервер его выполнит.

При асинхронном чтении/записи клиент посылает серверу запрос, а сам продолжает работать. Когда сервер выполнил запрос, клиент получает уведомление. Этот режим обладает большей гибкостью.

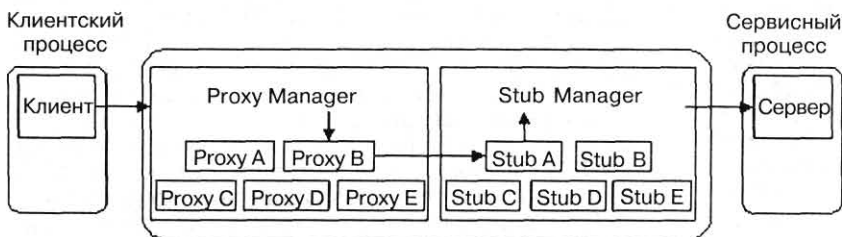


Рис. 3. Вызов функции объекта в случае DCOM

тью и рекомендуется в большинстве случаев, так как он минимизирует затраты процессорного времени и сетевых ресурсов, особенно при передаче большого количества элементов данных.

В случае подписки клиент передает серверу список интересующих его переменных, а сервер присылает клиенту информацию об изменившихся переменных из этого списка. При этом сервер будет передавать клиенту информацию только тогда, когда данные изменились, причем эти данные передаются единым блоком. Эти меры позволяют существенно ускорить обмен данными, особенно если речь идет о взаимодействии через сеть.

OPC и другие стандарты

Для подключения устройства ввода/вывода к SCADA-системе в настоящее время используются следующие механизмы:

Dynamic Data Exchange (DDE) — стандартный динамический обмен данными;

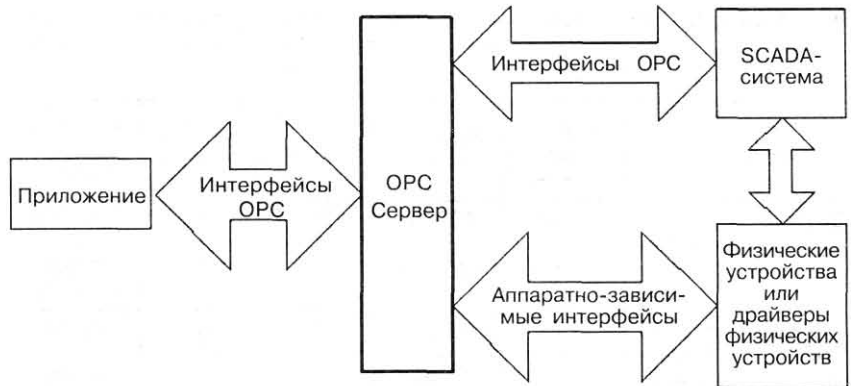
собственные протоколы фирм-разработчиков SCADA-систем; OPC-протокол.

В SCADA-системах основным механизмом, используемым для связи с внешними уровнями (верхними и нижними), изначально являлся механизм DDE, использовавшийся в качестве механизма разделения данных между прикладными системами и аппаратными устройствами. Но DDE присущ ряд ограничений по скорости и надежности. Стандарт DDE плохо подходит для обмена данными в реальном масштабе времени. Для преодоления недостатков DDE разработчики SCADA-систем предложили свои собственные протоколы, такие как FastDDE (Wonderware), AdvancedDDE (Rockwell), I-NET (Adastra). Подобные протоколы обеспечивают самый быстрый обмен данными с контроллерами, но имеют ряд недостатков [3]:

1) для каждой SCADA-системы пишется свой драйвер для каждого поставляемого устройства;

2) в общем случае две SCADA-системы не могут иметь доступ к одному устройству в одно и то же время, поскольку каждая из них поддерживает обмен через свой драйвер;

3) данные протоколы не поддерживают удаленного доступа к устройству.



■ Рис. 4. Обмен между SCADA-системой, приложениями и физическими устройствами через OPC-интерфейс

Взамен DDE компания Microsoft предложила более эффективное и надежное средство передачи данных между процессами COM/DCOM. На базе COM/DCOM появляется новый стандарт OPC, ориентированный на рынок промышленной автоматизации. Стандарт OPC поддерживается во многих современных SCADA-системах. В табл. 1 перечислены некоторые из популярных на западном и российском рынках SCADA-систем поддержки OPC.

Важной характеристикой является производительность OPC-сервера. Она зависит от ряда факторов — вида OPC-сервера (внутризадачный, локальный или удаленный), возможности группировки данных для отправки клиентам (передача 100 элементов раз в 1 с займет значительно меньше ресурсов, чем одного элемента через каждые 10 мс [4]), операционной системы (по результатам исследований центра Doculabs [7], любое приложение выполняется в среде Windows 2000 быстрее, чем на Windows NT, на 16–122 %) и т. д. Сказанное иллюстрирует табл. 2, в которой представлены результаты тестирования производительности OPC-серверов, разработанных консорциумом «OPC Foundation» [8]. OPC-клиентом в данном случае является простая специализированная программа, разработанная консорциумом для тестирования производительности OPC-серверов.

Преимущества и недостатки OPC

Как уже говорилось, основная цель OPC-стандарта заключается в определении механизма доступа к данным с любого устройства из различных приложений. OPC позволяет производителям оборудования стандартизовать свои продукты, обеспечивает подключение оборудования различных производителей, в том числе подключение к SCADA-системам.

■ Таблица 1. Популярные SCADA-системы поддержки OPC

SCADA-системы	Фирма-изготовитель	Страна
InTouch	Wonderware	США
GenieDAQ	Advantech	США
Genesis32	Iconics	США
Trace Mode	Adastra	Россия
Citect	Ci Technologies	Австралия
FactoryLink	United States DATA Co.	США
RSView	Rockwell Software Inc.	США
Lookout	National Instruments	США
Fix Dynamics	Intellution	США
Контур	Объединение ЮГ	Украина
Eclipse SCADA	Eclipse Software	США

■ **Таблица 2.** Производительность OPC-серверов «OPC Foundation» (на компьютере с процессором Pentium 233, ОС Windows NT 4.0 Workstation)

Тип сервера		Число элементов транзакций	Кратность передач	Время, с	Затраты процессорного ресурса, (%)	Скорость опроса элементов, с
Внутри-задачный	Клиент	100	100000	9	100	1111000
	Сервер	—	—	—	—	—
	Клиент Сервер	1 —	5000000 —	22 —	100 —	227000 —
Локальный	Клиент	100	10000	16	40	62500
	Сервер	—	—	—	60	—
	Клиент Сервер	1 —	50000 —	15 —	50 50	3333 —
Удаленный (Ethernet 10Base-T)	Клиент	100	1000	15	10	6666
	Сервер	—	—	—	12	—
	Клиент Сервер	1 —	3000 —	9 —	10 10	333 —

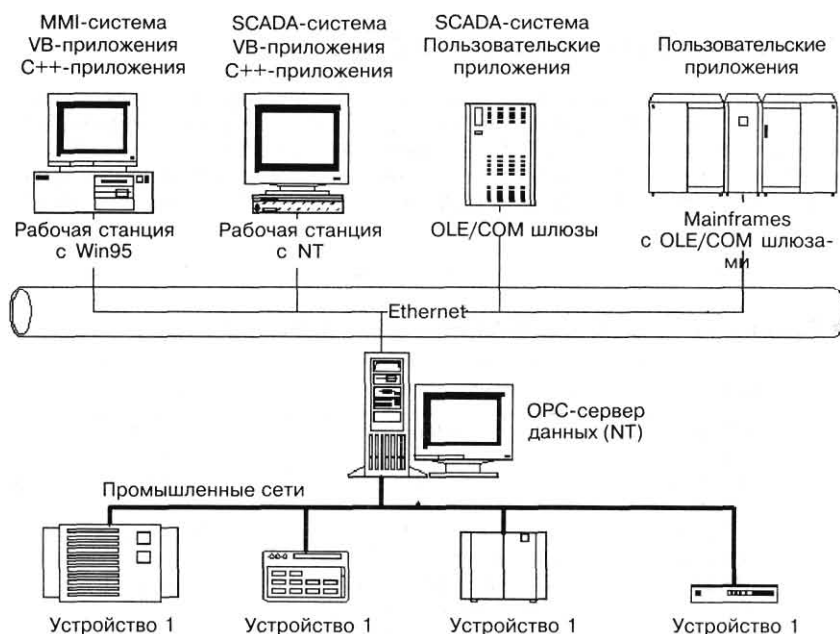
Идеальной выглядела бы следующая картина. Все в мире признают OPC своим стандартом. При этом все поставщики оборудования снабжают свои продукты OPC-серверами. Все поставщики программ для систем управления делают свои продукты OPC-клиентами — а все производители операционных систем поддерживают в своих ОС технологии COM/DCOM, а также предоставляют сервисный инструментарий. Все это они делают на высоком профессиональном уровне и очень грамотно рассказывают сборщикам систем, как это все собирать и конфигурировать. Тогда пользователи будут иметь широкие возможности, чтобы выбирать оптимальные для своей системы компоненты, а не создавать программу «с нуля» и не выбрасывать имеющийся контроллер. Таким образом, по образному выражению Теркеля [4], OPC — это свобода выбора.

Преимущество применения стандарта OPC с точки зрения интеграции достаточно прозрачно. С применением OPC-стандарта появляется возможность интегрировать в единую систему неоднородные узлы (рис. 5). Каждый из представленных на рисунке узлов может обслуживаться OPC-сервером или OPC-клиентом, которые обеспечивают простой приток данных в программу визуализации, в базы данных и т. п. с нижнего уровня на более высокий уровень.

OPC-стандарт можно рассматривать как «универсальный переходник». Если мы заменяем какой-нибудь компонент клиентского приложения, то нет необходимости корректировать серверное приложение, поскольку COM обеспечивает эффективное управление изменением программы. Если мы хотим добавить в систему новое оборудование, то достаточно включить его в OPC-сервер. При этом клиентские программы не требуют изменений, в том числе для SCADA-систем.

В настоящее время консорциум «OPC Foundation» набирает силу в разработке открытых промышленных стандартов на основе OPC-стандарта на базе PC и операционных систем Microsoft. Сейчас в состав «OPC Foundation» входят более 300 членов, среди которых практически все мировые ведущие производители технологического оборудования, систем автоматизированного управления и программного обеспечения. Членами организации являются, например, фирмы Iconics Inc. (США), Wonderware (США), Adastrа (Россия), Siemens (Германия), Rockwell Software (США), Intellution (США), Ci Technologies (Австралия), Indusoft Russia (Россия), Fastwel Inc. (Россия), ABB Automation (США), Fieldbus Foundation (США), Toshiba Corp. (Япония), Hitachi (Япония), National Instruments (США). Организация пытается охватить все аспекты, связанные с взаимодействием с технологическим оборудованием. Ведущие производители стараются максимально учесть свой опыт и предоставить все необходимое пользователям OPC. Этот факт показывает большой авторитет OPC-технологии. Это перспективная технология для использования в развитии автоматизированных систем управления.

Однако OPC-технология несет в себе и ряд недостатков.



■ **Рис. 5.** Стандарт OPC — путь к интеграции разнородных систем

1. OPC не обеспечивает работы в жестком реальном времени, поскольку в DCOM отсутствуют понятия качества обслуживания, крайних сроков и т. п. В то же время контроль за «устареванием» данных имеется: каждое передаваемое значение сопровождается меткой времени происхождения.

2. OPC может использоваться только там, где установлен Microsoft DCOM. В настоящее время технология COM/DCOM поддерживается в ОС Microsoft, в Unix-подобных ОС, в ОС реального времени VxWorks; обеспечивается фирмой WindRiver, имеется поддержка OPC, встроенная в систему разработки Tornado [9]. В других операционных системах поддержки COM/DCOM нет. Это не очень отрадный факт, поскольку разработчиков систем автоматизации в первую очередь интересуют ОС реального времени.

3. OPC-интерфейс — это интерфейс для системы верхнего уровня. Это означает, что нижележащие слои — контроллеры, полевые шины и т. д. — недоступны для самого OPC-сервера и для пользователей и представлены в виде «черного ящика».

4. Программирование COM непростое, даже с применением ATL. Сами OPC-объекты и их OPC-интерфейсы достаточно сложны и громоздки. Необходимо владеть вопросами системного уровня, такими как асинхронный обмен, многозадачность, синхронизация. Кроме того, в COM допускается выделение памяти в сервере, а удаление ее возлагается на клиента. Малейшая неточность вызывает трудно устранимые утечки памяти. А учитывая, что OPC-сервер обычно должен работать стационарно, рано или поздно крах системы неизбежен.

5. Технология DCOM не работает в глобальных сетях. Поэтому идея применения OPC-технологии в Internet нереальна. Вместе с тем, организовать обмен в этом случае также возможно следующим образом. Расширение Web-сервера является OPC-клиентом, собирающим данные от OPC-серверов. На стороне клиентов запускается динамическая html- или xml-страница, которую даже можно сделать OPC-сервером для других приложений. Указанная страница получает данные от Web-сервера [9].

Заключение

OPC-технология содержит стандарты, обеспечивающие взаимодействие программных средств в промышленной автоматизации. В технологии заложены богатые возможности, которые дают организаторам предприятия возможность интеграции разнородных систем и обеспечивают разработчикам свободу выбора с применением OPC-драйверов, позволяют не задумываться по поводу поддержки аппаратуры завтрашнего дня. Вместе с тем можно отметить, что процесс становления технологии еще далеко не завершен и есть много проблем, которые предстоит решить. Одной из таких проблем является отсутствие серьезных универсальных OPC-серверов, обеспечивающих подключение любого оборудования (основные корпорации не заинтересованы в поддержке конкурентов). Решение этой задачи представляется весьма актуальным.

Литература

1. **OLE for Process Control Data Access Standard Version 1.0A.** — September 11, 1997 OPC Foundation, Austin, Texas.
2. **Fastwel Universal OPC-сервер.** Версия 1.0. Руководство пользователя. — Fastwel Inc., 1999–2001.
3. **Куцевич Н. А., Жданов А. А.** Программное обеспечение систем контроля и управления и Windows-технологии // Мир компьютерной автоматизации. — 1999. — № 3.
4. **Теркель Д.** OLE for Process Control—свобода выбора // Современные технологии автоматизации. — 1999. — № 3. — С. 28–32.
5. **Пьюполо Д.** OLE: создание элементов управления: / Пер. с англ. — Киев: Издательская группа BHV, 1997. — 432 с.
6. **Технология** разработки программного обеспечения: Учебник / С. Орлов. — СПб.: Питер, 2002. — 464 с.
7. **М. Pendleton, G. Desai,** @Bench Test Report: Performance and Scalability of Windows 2000. Doculabs, 2000. <http://www.doculabs.com>
8. **DCOM,** OPC and Performance Issues, Intellution Inc, 1998, http://www.opcfoundation.org/07_download/performance.doc
9. **Куцевич И. В., Григорьев А. Б.** Стандарт OPC — путь к интеграции разнородных систем // Мир компьютерной автоматизации. — 2001. — № 1.