

# Отличия технологий в STM и OPC

*Любое сравнение различного всегда имеет погрешности,  
но если вы достаточно компетентны,  
то можете добавить своих (погрешностей) ©*

Итак, мысли по поводу плюсов и минусов технологии OPC (Ole for Process Control) и STM (Network Server TeleMetric).

На самом деле не корректно сравнивать технологию (OPC) и технологии продуктов (STM).

Корректнее сравнивать реализации систем на технологии (OPC) с продуктами STM.

OPC – это технология, позволяющая на выходе “OPC сервера” получить стандартный интерфейс для получения данных от контроллера.

“OPC сервер” – это программа - транслятор протокола обмена конкретного устройства (серии устройств) в протокол по спецификации OPC Foundation, иначе говоря конвертор аппаратно – зависимого представления данных в контролере (контролерах) в абстрактное представление данных (“tag”).

В состав семейства продуктов STM включена технология OPC как средство получения доступа к различным контроллерам.

В продуктах STM связь с конкретными устройствами обеспечивают “драйвера”. Они конвертируют аппаратно – зависимое представление данных в контролере (контролерах) в тематическое представление данных (расход прямой, расход обратный, напряжение... - иначе говоря обеспечивают на выходе драйвера объектное представление данных, “tag” является частным случаем представления данных в таком драйвере). Кроме того, драйвера, как правило, осуществляют промежуточные вычисления (например расчет средних значений, времени отключения/простоя или включения/работы и т.п.), авто синхронизацию архивов драйвера с устройствами и другие необходимые для формирования объектной модели данных процедуры и действия). Драйвера обеспечивают реконнекты по резервным маршрутам и многие другие необходимые функции.

Большая часть SCADA систем, если не все, используют технологию OPC как основу для работы с контроллерами (PLC). В связи с этим можно смело выделить ряд особенностей:

1. OPC - это надстройка над COM/DCOM, а значит, что работает только под управлением Microsoft Windows. Реализации под другие ОС скорее “экспериментальны”.
2. COM – технология того же Microsoft и под той же Windows, да еще и локальная.
3. DCOM – расширение COM для работы по сети.
4. COM/DCOM – работает только в сети Microsoft Windows. Примеров реально работающих SCADA под другие ОС я не нашел.
5. За счет того, что передача данных посредством COM/DCOM замаскирована внутри библиотек и сервисов (так называемый МАРШАЛИНГ), реализующих

SOM/DCOM, она происходит очень медленно, что становится дорогим, а при больших масштабах – очень дорогим решением.

## СТРУКТУРЫ ДАННЫХ

Теги (“tag”) OPC имеют только 4 элемента:

1. Тип Значения
2. Значение
3. Качество
4. Время значения

Типы значений строго определены в спецификации OPC – это стандартные типы данных для этой технологии.

Теги в OPC и объекты в STM - синонимы. Элемент в теге (OPC) и поле в объекте (STM) – синонимы. Количество элементов в теге – 4. Количество полей в объекте – до 250 (реально требуется до 30). Поле может быть значение (чего – либо: переменной, времени...), формула, или ссылка на другой объект. Объекты (STM) объединяются в “типы” (разные типы могут иметь одинаковую (для увеличения реактивности) или присущую только этому типу структуру полей). Количество объектов, включаемых в тип искусственно ограничено 125'000 (для повышения реактивности обработки). Количество значений каждого объекта, входящего в конкретный тип (глубина ретроспективы) ограничивается операционной системой (общий объем всех типов для 32-х битной ОС – 2Gb, для 64-х битной – 4 Tb). Количество типов на одном сервере ограничено 4'000'000'000'000. Количество непосредственно взаимодействующих между собой серверов (сервер - сервера) ограничено числом одновременно открываемых в операционной системе сокетов (около 65'000, практически более 100 открывать не приходилось). Иерархическая архитектура построения системы сбора и управления в корпорациях это ограничение делает не несущественным (сложно “придумать” архитектуру системы, где 65'000 низовых серверов непосредственно соединены с одним сервером – концентратором, не имея промежуточных серверов - концентраторов).

Синонимы “tag” - объект:

1. Тип значения – это поле в объектах STM отсутствует, его (логически) заменяет имя типа, входящее в состав полного имени объекта.
2. Значение - значение (аналогично)
3. \$Valid – качество значения (аналогично)
4. \$Time – время значения (аналогично)
- Остальные поля в объектах STM могут иметь доступные в современных языках типы, (целый, плавающая точка, строковая переменная...).
- Группы и подгруппы в STM описываются в имени объекта и разделяются через нижнее подчеркивание: xxx\_yyy\_mmm.

## АДРЕСАЦИЯ ДАННЫХ.

Полная адресация объектов в STM имеет фиксированную точечную нотацию, несущую смысловую нагрузку: «Сервер.Тип.Объект.Поле». Она отражает суть древовидной адресации.

Т.е. каждый ТИП находится на определенном СЕРВЕРЕ (имеется в виду процесс в операционной системе). Внутри этого ТИПа есть ОБЪЕКТ, а в нем – ПОЛЕ. Такое

(логически четкое) построение адресного пространства стандартизует обращение к объектам и их полям на разных серверах и позволяет избежать путаницы в мульти серверных конфигурациях (системах).

Адресация OPC тоже имеет точечную нотацию, но адресное пространство строится исходя из логики производителя оборудования и соответственно OPC сервера. Следовательно, замена оборудования требует модификации линейки алгоритмов обработки на серверах и отчетных формах. В STM этого недостатка нет. На крупных предприятиях и в корпорациях уровень КИПиА постоянно изменяется... Все что имеется в распоряжении OPC это: ГРУППА(Ы), которых может быть произвольное количество с произвольным числом вложенности, и ТЕГ (см. выше). Таким образом обращение к ТЕГУ (объекту сервера) выглядит следующим образом: «Группа1.Группа2...ГруппаN+1.Тег». Стандарты у каждого производителя оборудования свои и это вносит некоторую путаницу при эксплуатации. Однако дает преимущество в удобстве при интерактивном обращении к объектам сервера из программы конфигуратора. Одним словом адресация в OPC очень удобна при описании локальных (включенных в одну задачу) данных, а адресация STM ориентирована на многосерверную и распределенную обработку данных. В STM это ограничение при работе с OPC серверами обходится приведением к адресации STM на выходе драйверов.

### **OPC – технология для “настоольных” систем.**

1. Так как OPC построена на COM/DCOM, несмотря на то, что транспортным протоколом может быть TCP/IP, технология практически не применима и как минимум затратна в условия корпоративных и гетерогенных сетей. (Это утверждение будет верно, до тех пор, пока не появятся более или менее работоспособные реализации, использующие OPC, под \*nix-подобными системами).
2. С безопасностью COM/DCOM тоже есть не решенные проблемы. На практике чтобы заставить работать OPC сервер через сеть уровень безопасности должен быть НУЛЕВЫМ. Помимо этого Microsoft постоянно меняют реализацию COM/DCOM не только от версии к версии Windows, но даже и в пакетах обновлений.

Двух выше указанных проблем вполне достаточно для того чтоб отказаться от применения технологии OPC при автоматизации крупных или территориально разнесенных (использующих корпоративные сети) предприятий.

OPC технология хороша для автоматизации одного объекта, причем все уровни автоматизации должны быть расположены на одном или нескольких компьютерах, стоящих рядом.

STM технология изначально предназначена для работы в распределенных TCP/IP сетях с несколькими (десятками и сотнями) серверов, где могут обитать хакеры и вредоносные программы.

### **STM – технология автоматизации произвольного масштаба**

STM для работы с оборудованием, как правило, используются драйвера (описаны в начале статьи). Драйвера в STM обычно портируемы (работают под различными операционными системами, включая Windows и Linux. Некоторые драйвера (из-за

удобства среды разработки) сначала пишутся под Win32, а потом портируются под \*NIX системы.

Вся серверная часть комплекса изначально ориентирована на \*NIX системы, что повышает отказоустойчивость и масштабируемость. Под заказ любая из версий ПО может компилироваться под Win32, так, как все “внутренние” компоненты сервера написаны на стандартном языке Си/Си++.

Еще одной особенностью STM является не обязательность внешних History серверов. Сам STM, да и многие драйвера ведут собственные базы, которые по сути есть ОЗУ – резидентные, дампируемые кольцевые “кубы памяти” – обладающие теоретически предельным быстродействием и защитой от переполнения. Отсюда же вытекает следствие простоты резервирования баз на разных аппаратных платформах (дублирование и т.д) и работа с внешними SQL базами через драйверы ODBC.

Базовый протокол обмена между компонентами системы TCP/IP, что позволяет эксплуатировать комплекс в сети, где в качестве серверов используются распределенные аппаратные и/или программные платформы. В сети данные передаются минуя различные высокоуровневые библиотеки, что существенно сказывается на реактивности всей системы. При прогоне трафика между серверами, а так же между серверами и клиентами данные могут дополнительно шифроваться на сетевом уровне.

Учитывая потребности госструктур, крупных предприятий и корпораций в информационной безопасности и защите инвестиций ВСЕ исходные тексты и “технологическая” - ДСП документация на продукты “STM” предоставляется по отдельному соглашению, содержащему ряд гарантий для обеих сторон, передаются Заказчику.

Денис Парыгин

[den@rest.ru](mailto:den@rest.ru)

Группа предприятий «РЭСТ»

ООО «Инфо-Сети Плюс»