

## ФУНКЦИИ И СТРУКТУРЫ ИНТЕГРИРОВАННЫХ СИСТЕМ

Взаимосвязь процессов проектирования, подготовки и управления предприятием и производством.

### Методические указания

Теоретические попытки классифицировать задачи, решаемые предприятиями, привели к разделению их на две группы:

- технологические производственные;
- экономические, административные и логистические.

Первая относится строго к производственной деятельности, вторая - к административно-хозяйственной. Сотрудники вычислительных центров - постановщики задач и программисты - решали задачи обеих групп, ориентируясь, естественно, на те технические и программные средства, которые были доступны в текущий момент на данном предприятии. Если новая задача требовала модернизации существующих систем, происходила частичная замена устаревшего оборудования и/или программного обеспечения. Связь между двумя группами задач была «человеко-бумажной», т. е. данные между ними курсировали на бумажных носителях через персонал предприятия.

Итак, *первая группа задач* управления промышленным предприятием - то, что в СССР именовалось АСУТП (Автоматизированные системы управления технологическими процессами).

В сегодняшней интерпретации эту группу задач в иерархии управления производством относят к системам типа **SCADA** (Supervisory Control and Data Acquisition) или **DCS** (Distributed Control Systems). Оба указанных типа систем принадлежат классу MMI (Man-Machine Interface), что означает «человеко-машинный интерфейс» в смысле обеспечения двусторонней связи «оператор - технологическое оборудование». Системы MMI все чаще называют HMI (Human-Machine Interface). То есть MMI - это средство отображения и представления технологической информации.

К классу **DCS** можно отнести однородные системы, распределенные не только территориально, но и композиционно - в том смысле, что они состоят из равноправных разнофункциональных узлов. Это могут быть подсистемы контроллеров, узел ведения архива, операторские станции, узел связи с другими системами, инженерные станции. Системы же типа **SCADA** тяготеют к серверной архитектуре. Выделенный узел осуществляет сбор информации от контроллеров, ее обработку и передачу им управляющих значений. Этот же узел может быть рабочим местом оператора или сервером отдельной операторской станции.

Задачи программирования логических контроллеров PLC (Programmable Logic Controller) также относятся к сфере АСУТП и реализуются чаще всего с использованием методики Ladder Logic, которая представляет собой специализированный язык, близкий по типу к мнемонике электрических схем.

Системы *второй группы* относятся к классу ERP (Enterprise Resource

Planning) - планирование ресурсов предприятия или MRP II (Manufacturing Resource Planning) - планирование ресурсов производства. Системы ERP ориентированы на предприятие в целом, а MRP - на его технологические подразделения. Если опять-таки вспомнить нашу старую терминологию, то это задачи АСУП (автоматизированные системы управления предприятием).

САПР – это системы автоматизированного проектирования, без которых не могло обойтись ни одно промышленное предприятие, чья продукция требует конструкторской документации. Современные технологии САПР для предприятий представлены системами CAD/CAM/CAE/PDM (Computer Aided Design, Manufacturing, Engineering, Product Data Management). Эти системы позволяют обойтись без «бумажной» документации, осуществляя прямую связь между процессами разработки изделия и его производства, что позволяет повысить качество продукции и сократить период разработки.

Представьте, что предприятие в течение нескольких десятилетий закувало или самостоятельно разрабатывало программные пакеты, решающие все вышеперечисленные задачи. Означает ли это, что оно превратилось в единую информационно-управляемую структуру? Нет. Теперь необходимо как-то связать приложения между собой.

Возникает еще одна задача - обмен данными между приложениями, разработанными разными фирмами в разное время на разной технике. Причем ее следует понимать не как необходимость связать две или три группы систем, о которых говорилось выше, а в более общем смысле - как задачу связи и мобильности всех компонентов информационно-управляющей среды предприятия.

Предприятия столкнулись с этой проблемой уже давно и решали ее чаще всего в частном порядке, путем написания драйверов для связи приложений или с помощью других «житейских хитростей». Столь популярный в различных областях деятельности метод «заплаток» и здесь показал себя как неоправданно дорогой, долгий и малоэффективный. Кроме того, полученные решения трудно было распространить на другие приложения и системы.

Существует еще один важный аспект этой проблемы: информационные технологии стареют намного быстрее, чем производственное оборудование. «Железо» устаревает в среднем через 6-12 месяцев после выхода на рынок. Период появления новых версий современных информационных систем - один год. Вместе с тем в промышленности до сих пор применяются ЭВМ 70-х годов, не говоря уже о датчиках, исполнительных механизмах, кабельном хозяйстве и пр. Необыкновенно стремительное развитие информационных технологий - большая проблема для промышленности. Предприятие не может себе позволить менять все измерительное оборудование каждые несколько лет. Выход один - постепенная частичная замена устаревшей техники.

Задача объединения промышленных приложений не только приобретает ключевое значение для конечных пользователей, но и становится модной и популярной среди разработчиков, т. е. тех, кто должен найти ответ.

Рассмотрим подходы к решению задачи объединения промышленных

приложений, предлагаемые современными производителями программных продуктов.

Прежде всего, задачу объединения АСУТП и АСУП условно относят к системам уровня MES. Стремление связать системы типа SCADA/DCS с системами верхнего уровня ERP/MRP II существовало всегда. Однако в процессе развития различных промышленных приложений выявились участки, где обеспечение обмена данными представлялось особенно перспективным и интересным, например: между CAD-системами и ERP/MRP II, между ERP/MRP II и подсистемами ведения архива значений технологических параметров, между системой SCADA и подсистемами контроллеров и т. д. В этом смысле правильнее было бы представить техническое решение, обеспечивающее взаимодействие приложений, как "программный слой", с которым другие системы имеют двустороннюю связь.

Каковы же условия этой задачи? Если проанализировать все требования, выдвигаемые пользователями и разработчиками систем управления предприятием, то можно выделить два основных:

- Единое информационное пространство. Ситуация взаимного обмена данными для приложений должна стать обыденной. Необходимо, чтобы данные одного приложения были доступны другому в реальном времени.
- Гибкость (в смысле способности к быстрому перестроению). Возможность безболезненного добавления новых приложений и технологий, которое не требует изменения существующей структуры. Одновременно с этим удаление (замена) рабочих компонентов не должно разрушать систему.

Пожалуй, сегодня можно говорить о трех ключевых направлениях решения задачи: стандартизация, использование связующего ПО (middleware), внедрение глобальных промышленных серверов.

В офисных приложениях преимущества стандартизации неоспоримы. Это и уменьшение цены на приложение, и повышение его эффективности и удобный пользовательский интерфейс. Промышленные приложения совсем другое поле деятельности. И если образно сравнивать возможности освоения этих двух "пространств", то в первом случае мы имеем дело с равниной, а во втором - с сильно пересеченной местностью. Кроме того, что промышленные приложения делятся на группы по типам систем, внутри каждой группы тоже нет функциональной однородности. Многие отрасли промышленности предъявляют к системам управления свои, уникальные требования, связанные с конкретными технологиями производств.

Еще один нюанс - период обновления программных продуктов в промышленности намного больше. Здесь пока нет своего аналога Microsoft, который диктовал бы такой стремительный темп развития. Скорее всего, таковой и не появится из-за невозможности охватить столь огромный и неоднородный рынок. Однако, несмотря на очевидные подводные камни, положительный опыт стандартизации офисных пакетов неизбежно открывает путь введения стандартов на разработку промышленных приложений. Под

этим не следует понимать переход к использованию одной операционной системы, базы данных или сетевого протокола, например TCP/IP, поскольку все это никоим образом не гарантирует возможности обмена данными между промышленными приложениями.

Первым шагом в направлении стандартизации была попытка создать однородные протоколы для связи с производственным оборудованием. В начале 80-х годов корпорация General Motors разработала протокол автоматизации производства MAP (Manufacturing Automation Protocol). В его основе лежит идея стандартного коммуникационного стека и генерации сообщений в едином формате. Несмотря на вес General Motors на мировом рынке, протокол MAP так и не получил всеобщего признания: он очень сложен и требует больших вычислительных ресурсов.

Современные решения в области стандартизации связаны прежде всего с фирмой Microsoft. Это в первую очередь технология OPC (OLE for Process Control), т. е. OLE (Object Linking and Embedding) для технологического управления. Она представляет собой стандартный метод для доступа к периферийным устройствам, системам SCADA/MMI или другим промышленным приложениям, основанным на технологиях OLE, COM (Component Object Model) и DCOM (Distributed COM). В общих словах OPC представлена набором стандартных объектов, методов и свойств, отвечающих требованиям промышленных приложений реального времени. Эти требования включают в себя синтаксис для доступа к объектам, эффективную передачу данных от оборудования к приложениям, способность клиента работать с несколькими серверами одновременно и поддержку конфигурации сервера. Программные пакеты на основе OPC легко интегрировать в бизнес-приложения, поддерживающие OLE. К сожалению, набор продуктов, разработанных по этому стандарту, весьма и весьма ограничен. Сегодня судьба OPC - в руках производителей промышленных приложений.

Последняя разработка от Microsoft в этой области анонсирована в сентябре 1997 г. и называется - Windows DNA (Windows Distributed interNet Applications Architecture). Эта архитектура также основана на объектно-ориентированной COM-технологии создания функциональных пользовательских компонентов. Новая идея - разработка спецификаций по отдельным отраслям и сегментам промышленности (Vertical Industry Specifications) - не лишена логики и позволит сконцентрироваться на нескольких областях производства, где наиболее популярны программные продукты, поддерживающие COM. Это можно расценивать, как признание поражения всех попыток выработать общий промышленный стандарт.

После неудачи General Motors на пути стандартизации для решения задачи интеграции был создан консорциум, в который вошли крупные автомобильные производители (Renault, Mercedes-Benz и др.), представители авиационной промышленности и некоторые фирмы бытовой электроники (Bosch, Siemens Automation и др.). Он получил название AIT (Advanced Information Technology). С его помощью была разработана интеграционная

платформа CCE (Common Computing Environment). Данная среда позволяет приложениям независимо от протоколов, операционных систем, баз данных и методов доступа общаться друг с другом. Развитие этой идеи привело к возникновению архитектуры CORBA (Common Object Request Broker Architecture).

Консорциум OMG (Object Management Group), основанный в 1989 г., взял на себя труд разработать теорию объектно-ориентированной технологии для развития распределенных компьютерных систем. Основное направление деятельности консорциума можно сформулировать так: развитие общей архитектурной платформы для объектно-ориентированных приложений на основе открытых спецификаций.

Их усилия увенчались появлением архитектуры CORBA, позволившей решить проблему "информационного Вавилона" в мире промышленных компьютерных технологий. Она дает возможность приложениям обмениваться данными вне зависимости от того, где они находятся и кто их разработал. CORBA - это не набор директив, а только спецификации, которые свели воедино идеи членов OMG.

Если говорить о технологии CORBA, то это связующее ПО, "расположенное" между операционной системой и приложениями. Использование данного программного "слоя" облегчает процесс создания приложений, так как дает возможность разработчику абстрагироваться от особенностей операционной системы, сетевых протоколов и конкретных технических решений.

Но не все так хорошо, как кажется. Отметим некоторые недостатки внедрения связующего ПО:

- ускользают возможности использования в приложениях преимуществ конкретной операционной системы, сетевого протокола и т. п.;
- идея всеобщей открытости сомнительна в условиях конкурентной борьбы. Не все фирмы-производители встретят ее на ура;
- идея стандартизации будет погребена, и вряд ли впоследствии удастся вернуться к столь заманчивой концепции;
- очевидны технические трудности такого решения. В промышленности существуют сотни протоколов, с которыми надо научиться работать.

В процессе исследования вопроса выяснилось, что для многих фирм - поставщиков промышленного оборудования и программных пакетов идея использования связующего ПО все же ближе, чем идея стандартизации. Отделение современного программирования от реалий операционных систем и протоколов, как видимо, неизбежно и очень быстро происходит во всех областях. Трудности реализации успешно преодолеваются с помощью среды межобъектных запросов (Object Request Broker - ORB).

ORB - это связующее ПО, которое позволяет устанавливать клиент-серверные отношения между объектами. Используя ORB, клиент может легко вызывать сервис на объект-сервере, при этом аппаратно клиент и

сервер могут быть как на одной машине, так и на разных и общаться между собой по сети. ORB перехватывает запрос и отвечает за его доставку, передачу параметров, вызов сервиса, а также за доставку результатов. При этом клиенту совершенно не нужно "знать", где объект-сервер находится, какова его операционная система и на каком языке программирования он написан. Все это — вне интерфейса взаимодействия самих объектов. Таким образом, ORB обеспечивает обмен информацией между приложениями на различных устройствах в неоднородной распределенной среде, создавая связную объектно-ориентированную систему.

Версия 1.1 спецификации CORBA была выпущена OMG в 1991 г. В ней были определены язык описания интерфейса (Interface Definition Language - IDL) и интерфейсы прикладного программирования (Application Programming Interfaces - API), сделавшие доступным взаимодействие клиент-объект в среде ORB. Суть модели в следующем. Клиент запрашивает сервисы у объекта (который выступает в качестве сервера) через хорошо определенный интерфейс (последний специфицирован в IDL). Для доступа к объекту клиент создает запрос, т. е. сообщение, содержащее информацию о действии, ссылку на сервис, параметры.

Спецификация CORBA 2.0, "родившаяся" в декабре 1994 г., определила информационный обмен между приложениями различных фирм. Она добавила к уже освоенным высотам возможность обмениваться данными через Интернет по протоколу ИОР (Internet Inter-ORB Protocol) и платформенную независимость. Ее внедрение позволяет пользоваться преимуществами Интернет без перестройки промышленной системы.

Казалось бы, объектно-ориентированная технология стала панацеей от всех бед в области промышленных приложений. Применяй ее - и обретешь реальную гибкую информационную среду, объединяющую все приложения, используемые на предприятии. Причем приложения не надо переписывать.

Архитектура OMG/CORBA более зрелая, чем OPC и тем более Windows DNA. Уже существуют многочисленные ее реализации, применение которых в промышленности делает приложения независимыми от используемых устройств, сетей, операционных систем и компьютеров.

Третий путь - вполне самостоятельное направление, цель которого - удовлетворить в одном "сверхпродукте" или комплексе продуктов одной фирмы-производителя все потребности современного промышленного предприятия - своего рода скатерть-самобранка от автоматизации. Многие фирмы - производители систем SCADA движутся сейчас в этом направлении, больше всех здесь преуспела фирма Wonderware, выпустившая продукт FactorySuite. Кроме возможностей систем SCADA, в нем реализованы функции Batch Control, программирование логических контроллеров, ведение проектов, контроль качества продукции и некоторые функции автоматизации административного управления. Фирма Intellution, помимо систем SCADA, предлагает пакет типа Batch Control и пакеты с Интернет-функциями. Список фирм можно продолжить. Но все их предложения еще так далеки от полного комплексного решения задач автоматизации.