

Все возможные критерии объединяются в полихроматический универсум, из которого в дальнейшем формируются поэтапные подмножества оценочных критериев. В качестве примера сформируем полихроматический универсум A , состоящий из специальных критериев $A = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9\}$, где a_1 – надежность выполняемых функций, a_2 – сложность конструкции, a_3 – точность, a_4 – безопасность, a_5 – величина вспомогательного времени, a_6 – траектория и путь инструмента, a_7 – габариты, a_8 – возможность совмещения функций, a_9 – экономичность (рис. 2).

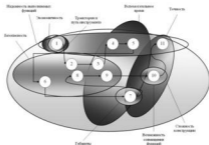


Рис. 2. Полихроматический универсум специальных критериев

Указанный подход позволяет, максимально используя накопленный опыт в виде множества конструктивных элементов, создавать в кратчайшие сроки новые конструкции, оптимизированные по ряду параметров.

Библиографический список

1. К.П. Жуков, Ю.Е. Гуревич. Проектирование деталей и узлов машин. – М.: изд-во «Стандарт», 2004. – 671с.
2. Быстриков И.К., Месерсон Е.А., Карязина Т.Н. Социальная экология: Курс лекций / Под общ. ред. Е.А. Месерсон. – Волгоград: Изд-во Волгоградского государственного университета, 1999. – 256 с.
3. ГОСТ 9.101-2002. Единая система защиты от коррозии и старения. Основные положения. Межгосударственный стандарт. Межгосударственный совет по стандартизации, метрологии и сертификации. Минск, 2002.
4. ГОСТ 27.004-85. Надежность в технике. Системы технологические. Термины и определения.
5. Мушак Э., Мюллер П. Методы принятия технических решений: Пер. с нем. – М.: Мир, 1990. – 208с.
6. Техническое творчество: теория, методология, практика. Энциклопедический словарь-справочник / Под ред. А.И. Половинкина, В.В. Попова. – М.: НПО «Информ-система», 1995. – 408с.
7. Логика проектирования машин. Учеб.пособие / Косов М.Г., Гуревич Ю.Е., Симанюков Е.А. – М.: ИЦ ГОУ МГТУ «Станкин», 2008. – 271с.
8. Павлов В.В. Структурное моделирование в CALS-технологиях / Под ред. Ю.М. Соломенцева – М.: Наука, 2006. – 307с.

ПРИМЕНЕНИЕ ПРОТОКОЛОВ CANBUS И PROFIBUS-DP ДЛЯ УПРАВЛЕНИЯ ПРИВОДАМИ В СИСТЕМАХ ЧПУ

Некметдинов Рамиль Амирович, Пушков Роман Львович, Ковалев Илья Александрович, Сорокоунов Артём Евгеньевич

Россия, Москва, ГОУ ВПО МГТУ «Станкин»
pushkov@incsystems.ru

Специфика современного управляемого технологического процесса требует распределенного функционирования вычислительных компонентов, т.е. включения их в общую информационно-вычислительную среду через промышленные сети. Решение заключается в применении принципов открытости и модульности архитектуры, что позволяет компоновать системы ЧПУ под конкретные технологические задачи.

Заложенное в систему свойство инвариантности позволяет реализовывать комплектацию системы ЧПУ для управления контроллерами приводов по промышленным сетям на базе различных интерфейсов. В данной статье рассмотрен вариант управления приводами на базе интерфейсов CANBUS и PROFIBUS-DP.

Обзор протокола CAN

В 1983 году компанией RobertBoschGmbH был разработан новый стандарт промышленной сети CAN (ControllerAreaNetwork- сеть контроллеров), а первые микросхемы контроллеров были выпущены фирмами Intel и Philips в 1987 году. В настоящее время CAN- контроллеры и трансиверы широко распространены в промышленной автоматизации, технологиях «умного дома», очень широко применяются в автомобильной промышленности и многих других областях.

Если рассмотреть структуру семипровиной модели ISOOSI, то CAN охватывает только два уровня – физический и канальный. Стандарт не предусматривает никакого протокола прикладного уровня для работы с CAN. Для его воплощения в жизнь некоторые фирмы разрабатывают его сами: CANopen (организация CIA), DeviceNet (Allen-Bradley), CAN Kingdom (Kvaser), SDS (Honeywell Micro Switch Division), J1939.

Итак, на канальном уровне при подуровне CANLLC (LogicalLinkControl) происходит подтверждение фрейма, уведомление о перегрузке, управление восстановлением данных. При подуровне CANMAC (MediumAccessControl) происходит формирование пакетов данных, кодирование, управление доступом, обнаружение ошибок, сигнализация об ошибках, подтверждение приема, преобразование из последовательной формы в параллельную и обратно.

На физическом уровне происходит обеспечение надежной передачи на уровне байтов, устанавливаются требования к линии передачи.

CAN является синхронной шиной, с типом доступа Collision Resolution (CR), который в отличие от Collision Detect (CD) сетей (Ethernet — это CD) детерминировано (приоритетно) обеспечивает доступ на передачу сообщения, что особо ценно для промышленных сетей управления (fieldbus). Передача ведется кадрами (фреймами) и сообщение посылается сразу всем узлам сети, т.к. в CAN-сети не один из узлов не имеет адреса. Полезная информация в кадре состоит из идентификатора длиной 11 бит (стандартный формат) или 29 бит (расширенный формат, надмножество предыдущего) и поля данных длиной от 0 до 8 байт. Идентификатор говорит о содержимом пакета и служит для определения приоритета при попытке одновременной передачи несколькими сетевыми узлами, таким образом в CAN не происходит непроизвольной потери пропускной способности канала при коллизиях.

Существуют следующие типы фреймов:

- DATAFRAME (фрейм данных) служит для передачи данных
- REMOTEFRAME (дистанционный фрейм, фрейм вызова) служит для передачи одним из устройств с последующим получением от другого в формате DATAFRAME с тем же идентификатором.
- ERRORFRAME (фрейм ошибок) передается узлом, обнаружившим в сети ошибку
- OVERLOADFRAME (фрейм перегрузки) обеспечивает промежуток между кадрами данных или запроса.

Фрейм данных состоит из следующих полей (рисунок 1): начало фрейма (StartOfFrame), поле арбитража (ArbitrationField), поле контроля (ControlField), поле данных (DataField), поле циклического избыточного кода (CRCField), поле уведомлений о приеме (ACKnowledgementField) и поле конца фрейма (EndOfFrame). Поле данных может иметь нулевую длину.

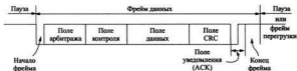


Рисунок 1 - Формат фрейма в протоколе CAN

Формат кадра данных имеет вид, представленный в таблице 1:

Таблица 4 - Формат сообщения технологического протокола.

Байт данных в сообщении CAN	D0	D1	D2	D3	D4	D5	D6
Формат технологического протокола	Команда технологического протокола		Адрес параметра			Значение параметра	

Большим преимуществом является то, что протокол CAN обладает развитой системой обнаружения и сигнализации ошибок. Для этих целей используется поразрядный контроль, прямое заполнение битового потока, проверка пакета сообщения CRC-полиномом, контроль формы пакета сообщения, подтверждение правильного приема пакета данных. Хемминговый интервал $d=6$. Общая вероятность необнаруженной ошибки $4,7 \times 10^{-11}$. Если сообщение доставлено с ошибкой, то происходит автоматический повтор передачи, как только сеть станет свободной.

Если говорить о скорости передачи, то все узлы сети должны работать с одной скоростью. Стандарт CAN сам как таковой не определяет скоростей работы, но большинство адаптеров позволяют плавно менять скорость в диапазоне от 10 Кбит/с до 1024 Кбит/с.

Схема передачи сообщений выглядит как показано на Рисунок 2.

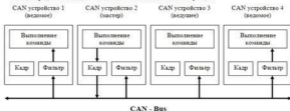


Рисунок 2 - Схема передачи сообщений

В CAN-сети ни один из узлов не имеет адреса. Вместо этого сообщения посылаются «всем», но содержит идентификатор, который описывает смысл посланных данных. Сообщение принимается узлом, если его идентификатор проходит через фильтр сообщений, имеющийся в каждом узле.

Как уже было сказано выше, разработка CAN закончилась на первых двух уровнях модели OSI, что и привело к появления разных не совместимых между собой протоколов прикладного уровня.

Самым распространенным является CANopen.

CANopen определяет протокол для распределенных промышленных автоматизированных систем на базе CAN. CANopen разработан организацией CAN in Automation (CIA) и стандартизирован CENELEC EN 50325/4.

Семейство CANopen профилей базируется на "Communication Profile" (профиль связи), который определяет базовые механизмы связи и множество различных профилей устройств или приложений, содержащих описание основных типов устройств, таких как шифровые и аналоговые I/O модули, приводы, программируемые контроллеры и кодирующие устройства. В профилях устройств содержится функциональность, параметры и доступ к данным процесса стандартных устройств соответствующих типов.

Главным элементом CANopen стандарта является описание функциональности устройства через словарь объектов. Каждая точка ввода словаря объектов обозначается через 16-ти битный индекс и 8-ми битный субиндекс. Некоторые точки ввода словаря объектов представляют собой базис для стандартизированного доступа к сети «Объектам приложения» устройства, например, входные и выходные сигналы, параметры устройства, функции устройства или переменные сети, а также «Объектам связи», которые описывают функциональность связи устройства.

Аналогично известным системам с полковой шиной, CANopen имеет два базовых механизма передачи данных:

- Высокоскоростной обмен небольшими объемами данных процесса через так называемые ProcessDataObjects - PDO (объекты данных процесса)
- Доступ к точкам ввода в словаре объектов через ServiceDataObjects - SDO (объекты данных сервиса).

PDO могут использоваться для передачи до 8 байтов данных. Передача и прием PDO может быть синхронизированной по всей сети с помощью синхронизирующих сообщений (Синхронные PDO). Назначение объектов приложения PDO (Объект передачи) настраивается с помощью структуры данных (PDO отображение) в словаре объектов. Передача SDO выполняется с подтверждением посредством двух CAN объектов, аналогично логическому соединению точка-точка между двумя устройствами сети. Адресации к соответствующим точкам ввода словаря объектов выполняется с помощью заданного индекса и субиндекса точки ввода в поле данных указанных CAN кадров. Передаваемые данные имеют произвольную длину. Передача SDO сообщений содержит дополнительные служебные данные протокола.

Все же при всех преимуществах CAN имеет недостатки, которые связаны с сравнительно высокой стоимостью CAN-устройств, с тем, что максимальная длина сети обратно пропорциональна скорости передачи, невозможно передача сообщений длиной более 8 байт. Но все же большим недостатком является отсутствие единого протокола привязанного уровня (хотя в таком случае, стандарт сети предоставляет широкие возможности для практически безымянной передачи данных между узлами, оставляя разработчику возможность вложить в этот стандарт всё, что туда сможет поместиться).

Обзор протокола Profibus

PROFIBUS (PROcess Field BUS) – промышленная сеть для технологических процессов, разработанная SiemensAG.

С помощью PROFIBUS, устройства разных производителей могут работать друг с другом без каких-либо специальных интерфейсов. Семейство PROFIBUS состоит из трех совместных друг с другом версий:

- PROFIBUS DP – применяется в высокоскоростных недорогих системах обмена данными между системами автоматизации (master) и децентрализованной периферией (slave) на полевом уровне.
- PROFIBUS PA – основан на PROFIBUSDP, применяется для связи устройств, находящихся во взрывоопасной зоне.
- PROFIBUS FMS – применяется при передаче больших объемов данных на уровне PLC и PC.

В системах ЧПУ обычно применяются средства коммуникации, основанные на версии протокола PROFIBUS-DP. В такой сети могут применяться ПЛК, устройства программирования ПЛК или ПК, устройства человеко-машинного интерфейса, приводы.

PROFIBUS имеет модульную структуру и предлагает целый спектр коммуникационных технологий, многочисленных приложений и системных профилей, а также инструментов управления устройством.

Сеть PROFIBUS построена в соответствии с семипровинной сетевой моделью ISO OSI, в которой используются только три из них (рисунок 3):

- физический уровень — отвечает за характеристики физической передачи;
- канальный уровень — определяет протокол доступа к шине;

- прикладной уровень — отвечает за прикладные функции.

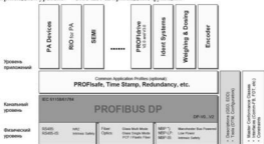


Рисунок 3 - Структура протоколов PROFIBUS

На физическом уровне PROFIBUS-DP может представлять из себя как электрическую сеть с шинной топологией на основе экранированной витой пары, соответствующую стандарту RS-485, так и сеть на основе волоконно-оптического кабеля. Скорость передачи данных в таких сетях может составлять от 9,6 кбит/с до 12 Мбит/с. Сеть строится из набора сегментов, в каждом из которых может работать до 32 устройств. Для увеличения размера сети до максимального размера (128 устройств), используются повторители интерфейса.

Канальный уровень модели OSI в PROFIBUS называется FDL-уровнем (Fieldbus Data Link – промышленный канал связи).

Объект MAC-control (Medium Access Control – управление доступом к каналу) на канальном уровне определяет процедуру передачи данных устройствами, включая управление правами на передачу данных через сеть. Протокол канального уровня обеспечивает выполнение следующих важных требований:

- в процессе коммуникации каждое устройство за отведенное ему время должно выполнять все свои задачи.
- взаимодействие ведущих устройств (контроллеров) с ведомыми должно происходить максимально быстро.

Для доступа ведущих устройств к сети используется метод передачи маркера. В этом методе сеть имеет логическую топологию кольца на уровне адресов устройств, каждое ведущее устройство получает доступ к сети только при получении маркера. По истечении определенного времени это устройство должно передать маркер следующему устройству, которое получает доступ к шине так же на определенный, точно заданный интервал времени, пока маркер находится у него. Интервал может быть установлен при конфигурировании системы.

Каждому мастеру в сети назначается свои ведомые устройства. В методе «ведущий/ведомый» процедуру коммуникации с ведомыми устройствами выполняет мастер, который обладает маркером. На время обладания маркером мастер становится ведущим также по отношению к другим мастерам и может осуществлять коммуникацию типа «мастер-мастер».

Прикладной уровень в PROFIBUS-DP представляется профилем.

Профиль – это набор правил работы и поведения устройств и систем, формирующих интерфейс программного приложения, предлагающие различные службы для организации синхронного и асинхронного обмена информацией.

Существует несколько типов различных профилей, гарантирующих полную совместимость оборудования различных фирм-производителей. Для систем ЧПУ наиболее интересным является «Профиль для приводов с изменяемым числом оборотов (3.072)» или PROFIDRIVE-профиль, разработанный ведущими производителями приводной техники.

Профиль устанавливает, как приводы параметрируются и передают заданные и истинные значения.

Профиль содержит необходимые установки для вида работы регуляторов числа оборотов и позиционирования. Профиль устанавливает основные функции приводов и дает достаточное свободное пространство для специфических пользовательских расширений. Профиль содержит описание пользовательских функций DP или альтернативных функций FMS.

Профиль PROFdrive определяет поведение устройств и метод доступа к данным электрических приводов по шине PROFIBUS, от простых частотных преобразователей до высокоточных приводов с сервоприводами.

Профиль содержит подробное описание коммуникационных функций «прямой обмен данными», «жесткая синхронизация» и «тактовая синхронизация», главным образом для приводов. Кроме того, он указывает все характеристики устройства, которые влияют на интерфейс подключения к контроллерам по PROFIBUS или PROFINET. Сюда входит состояние машины, интерфейс датчика положения, нормирование значений, определение стандартных телеграмм, доступ к параметрам привода и т.д. При этом профиль PROFdrive поддерживает как централизованную, так и децентрализованную концепцию управления движением.

Профиль PROFdrive старается сохранить интерфейс привода настолько простым, насколько это возможно, и освободить его от технологических функций. Эта философия гарантирует, что опорные модели, а также функциональность и производительность ведущего устройства PROFIBUS/PROFINET не будут иметь никакого или будут иметь очень маленькое влияние на интерфейс привода.

Интеграция приводов в решения автоматизации сильно зависит от задач привода. Для того чтобы охватить весь диапазон применений привода от самого простого преобразователя частоты до высокоточных, синхронизированных многоосевых систем в одном профиле, PROFdrive определяет 6 категорий приложений, которые соответствуют большинству применений приводов:

1. Стандартные приводы (насосы, вентиляторы).
2. Стандартные приводы с технологическими функциями.
3. Приводы позиционирования.
4. Приводы управления движением с централизованной системой управления верхнего уровня.
5. Приводы управления движением с централизованной системой управления верхнего уровня и патентованной концепцией позиционирования DSC.
6. Приводы управления движением с децентрализованной системой управления, расположенной в приводе.

Для задач системы ЧПУ наиболее применима категория 4 (рисунок 4).

В PROFIBUS сообщения можно передавать двумя способами: циклически (отправка данных без уведомления) и ациклически (отправка и приём данных с уведомлением).

Для передачи данных используется NRZ-кодирование и 11-битный формат (рисунок 5), включающий стартовый бит (0), 8 бит данных младшими разрядами вперёд, бит паритета (чётный) и стоп-бит (1). Телеграмма может содержать до 256 байтов, из них 244, плюс 11 служебных байтов (заголовков телеграммы).

Содержание телеграммы, представленной на рисунок 5:

SD – стартовый разделитель. Используется для указания начала телеграммы и её формата.

Имеется четыре типа разделителей для телеграмм запроса и ответа и один тип для короткого уведомления;

LE – длина передаваемых данных (DA+SA+FC+DSAP+SSAP+DU);

LEr – повторение поля LE с целью его резервирования;

DA – адрес устройства-получателя телеграммы;

SA – адрес отправителя;

FC – код типа телеграммы (запрос, уведомление, ответ, диагностические данные, тип устройства – master или slave, приоритет, уведомление);

DSAP – устройство-получатель использует это поле, чтобы определить, какой тип сервиса нужно выполнить;

SSAP – COM-порт отправителя;

DU – данные длиной от 1 до 244 байт;

FCS – контрольная сумма телеграммы (сумма значений полей DA + SA + FC + DU по модулю 255);

ED – признак конца.

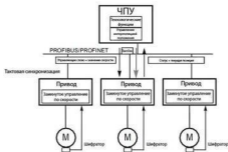


Рисунок 4 - Профиль ProfiDrive категории 4

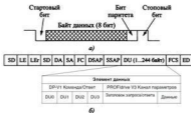


Рисунок 5 - а) структура слова б) структура телеграммы PROFIBUS

Данные (DU) профиля PROFIdrive состоят из двух частей. DP ID, содержащий уставку и текущее значение параметра, и PROFIdrive ID, включающий в себя заголовок запроса/ответа и данные.

Декомпозиция ядра системы ЧПУ на уровне управления приводами

В настоящее время существует большое многообразие промышленных сетей, которые можно в зависимости от типов устройств и способа управления разделить на три основные категории: сенсорные (sensebus: CAN, Seriplex, ASI, LonWorks и др.); сети с простыми полевыми устройствами (devicebus: Modbus, CAN, DeviceNet, Profibus DP, Lonworks, FIPRO, SDS, Interbus S и др.); полевые сети со сложными устройствами (fieldbus: Foundation FieldBus, Profibus PA, Hart и др.). Число проводных сетей промышленной автоматизации непрерывно растет. Этот процесс сопровождается внедрением и беспроводных технологий в их сетевую архитектуру. Однако если современные системы управления предприятием базируются на коммуникационных протоколах и стандартах, широко используя сети Ethernet и интернет-технологии, то в области систем управления промышленной автоматикой существует множество различных стандартов и протоколов. [1]

Для реализации поддержки в ядре максимально возможного количества протоколов взаимодействия с приводами предлагается выделить несколько уровней абстракции в компонентах управления приводами системы управления (Рисунок 6).

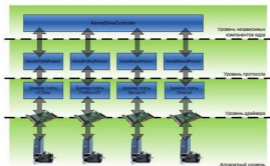


Рисунок 6 - Уровни абстракции компонентов управления приводами

На аппаратном уровне абстракции располагаются непосредственно приводы, приводящие в движение исполнительные органы станка или робота, а также интерфейсные платы, обеспечивающие связь компьютера, в том числе и с помощью ядра системы управления с приводами. Платы могут быть от различных производителей (многие производители, например Hilscher [2], предлагают решения для большинства известных полевых шин, используемых в системах автоматизации), главное – они должны поддерживать протокол, на котором работает привод.

Оболочка операционной системы реального времени, над которой построено ядро системы управления, с оборудованием компьютера производится за счет драйверов устройств. Драйверы устройств представляют собой программные модули, выстраивающиеся в операционную систему при помощи которых операционная система предоставляет интерфейс доступа к устройствам прикладным задачам [3]. Уровень драйвера представляет собой набор драйверов для различных, поддерживаемых устройств, которые загружаются в зависимости от наличия того или иного устройства.

Взаимодействие с приводами на информационном уровне представляет собой передачу приводу команд в виде пакета данных определенного формата и получение от привода ответа, также в виде пакета данных. Формат пакетов данных определяется протоколом передачи данных прикладного уровня [4]. На уровне протокола располагаются модули, которые предоставляют реализацию протоколов прикладного уровня.

Уровень независимых компонентов ядра управления приводами системы ЧПУ представляет собой универсальный модуль, который позволяет системе быть независимой от используемых типов приводов, так как предоставляет системе управления единый стандартный интерфейс не зависящий от используемых приводов.

Заключение

В статье рассмотрен вариант управления приводами в рамках системы ЧПУ на базе интерфейсов CANBUS и PROFIBUS-DP. Применение данных стандартов в рамках одной системы управления позволяет достигнуть инвариантности компоновки на базе модульных решений, что обеспечивает реализацию ядра системы управления под конкретные технологические задачи в рамках выбранного ценового диапазона.

Работа выполнена по Госконтракту №П1368 на проведение НИР в рамках ФЦП «Научные и научно-педагогические кадры инновационной России» на 2009 – 2013 годы.

Библиографический список.

1. Гайкович Г.Ф. Стандартизация в области промышленных сетей. Развитие беспроводных стандартов для АСУ ТП, Электронные компоненты, 2009, №1, с. 64.

2. <http://hibscher.com>
3. Alessandro Rubini, Jonathan Corbet. Linux Device Drivers. Second Edition. O'Reilly, 2001, 564p.
4. Олифер В.Г. , Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы. Издание 4-ое. СПб-6: Питер, 2010, 918с.

СОЗДАНИЕ ИНСТРУМЕНТОВ РАСШИРЕННОГО ПРЕДСТАВЛЕНИЯ ДЕЙСТВИЙ ОПЕРАТОРА В ПОЛЬЗОВАТЕЛЬСКОМ ИНТЕРФЕЙСЕ WINDOWS

Козак Николай Владимирович, Плаксин Александр Михайлович
РФ, г. Москва, МГТУ «Станкин»
alexander-boy@yandex.ru

Важную роль в процессе разработки, как программного продукта, так и программного комплекса играет процесс тестирования приложений. Тестирование выявляет логические ошибки, недостатки программы, проверяет и подтверждает стабильность работы.

Этап «тестирования» можно разделить на две части:

- тестирование в процессе разработки (применяется для выявления ошибок);
- визуальное тестирование (применяется для утверждения факта работоспособности программы, перед сдачей ее заказчику).

Ручное тестирование пользовательского интерфейса проводится тестировщиком-оператором, который руководствуется в своей работе описанием тестовых проверок в виде набора сценариев. Каждый сценарий включает в себя перечисление последовательности действий, которые должен выполнить оператор, и описание важных для анализа результатов тестирования ответных реакций системы, отражаемых в пользовательском интерфейсе. Типичная форма записи сценария для проведения ручного тестирования - таблица, в которой в одной колонке описаны действия (шаги сценария), в другой - ожидаемая реакция системы, а третья предназначена для записи того, совпала ли ожидаемая реакция системы с реальной и перечисления несоответствий.

Ручное тестирование пользовательского интерфейса удобно тем, что контроль корректности интерфейса проводится человеком, т.е. основным "потребителем" данной части программной системы. К тому же при чисто косметических изменениях в интерфейсах системы, не отраженных в требованиях (например, при перемещении кнопок управления на 10 пикселей влево), анализ успешности прохождения теста будет выполняться не по формальным признакам, а согласно человеческому восприятию.

При этом ручное тестирование имеет и существенный недостаток - для его проведения требуются значительные человеческие и временные ресурсы. Особенно сильно этот недостаток проявляется при проведении регрессионного тестирования и вообще любого повторного тестирования - на каждой итерации повторного тестирования пользовательского интерфейса требуется участие тестировщика-оператора. В связи с этим в последнее десятилетие получили распространение средства автоматизации тестирования пользовательского интерфейса, снимающие нагрузку на тестировщика-оператора.

Хотя на первый взгляд ручное тестирование – довольно простая операция, при неправильном применении она может привести к проблемам. Группы проектирования быстро выясняют, насколько сложно работать с многочисленными электронными таблицами и другими средствами тестирования и хранения результатов. Они отмечают большой объем повторной работы, связанный со многими используемыми сценариями тестирования, что обусловлено недостатком повторного использования и модульностью. И постоянно приходится следить за тем, чтобы свести к минимуму ошибки персонала и противоречивость результатов. Эти факторы сказываются на конечной цели ручного тестирования - на качестве оценки приложений и выявлении ошибок.

Существует множество средств для упрощения, автоматизации операций и управления процессом ручного тестирования, например можно использовать IBM® Rational® Manual Tester которое облегчает выполнение следующих задач:

- Ясного и краткого документирования процесса выполнения ручного тестирования;
- Возможности общего использования описанного тестирования в нескольких тестах;