

УДК 004.453 (045)

DOI 10.22213/2413-1172-2017-3-132-133

П. С. Ипатов, Удмуртский государственный университет, Ижевск

ТЕХНОЛОГИИ МЕЖПРОГРАММНЫХ ИНТЕРФЕЙСОВ

Основная проблема в IT-инфраструктуре крупных компаний состоит в том, что данная инфраструктура представляет собой сложный комплекс разработанных в различных стандартах приложений, работающих на разных программно-аппаратных платформах. Во-первых, так как в течение всего жизненного цикла информационной системы старые приложения модернизируются, то и сами межпрограммные интерфейсы всегда требуют модернизации. Во-вторых, линейный рост числа приложений приводит к квадратичному росту числа межпрограммных интерфейсов, а это, в свою очередь, приводит к образованию сложной смеси технологий. Существуют ли эффективные способы решения этой проблемы?

Чтобы понять, что такое межпрограммный интерфейс, каков принцип его работы, сначала обозначим, что такое интерфейс. Интерфейс – это не что иное как средство, при помощи которого пользователь может взаимодействовать с программой или операционной системой устройства. С его помощью пользователь может изучить любую программу, и когда человек овладевает навыками работы в одной программе, то он с легкостью сможет работать в других подобных, так как у них одинаковый интерфейс. Другими словами, интерфейс – это множество различных средств, которые позволяют человеку легко управлять вычислительной машиной [1].

Из большого количества требований, предъявляемых к интерфейсу, обычно называют дружелюбность, стандартизацию, функциональность и интеграцию различных приложений.

Дружелюбность. Удобство, простота, защита данных, подсказки, наличие справочной системы, множественность выполнения действий и другие элементы.

Стандартизация. Обычно одинаковые функции различных приложений управляются и выполняются одинаковыми элементами.

Функциональность. Почти все современные приложения представляют собой функциональные среды, направленные на решение некоторых прикладных задач. Приложение предоставляет в распоряжение пользователя ряд функций, которые выполняются под управлением и при участии пользователя.

Интеграция различных приложений. Каждое приложение работает с собственным представлением данных. Взаимодействие различных приложений возможно несколькими способами; обычно это называется межпрограммным интерфейсом [2].

Межпрограммный интерфейс – реализованная на компьютере возможность организации обмена дан-

ными между различными автономно функционирующими пакетами прикладных программ.

Технологии межпрограммного интерфейса появились с возникновением операционных систем полвека назад. Долгое время эти технологии непрерывно развиваются, предоставляя с каждым новым шагом развития новые преимущества программным комплексам, которые используют их [3].

Одним из таких шагов развития стало появление межпрограммных интерфейсов контейнерного типа (механизм DDE). Параллельно с этим появилась технология «клиент – сервер», которая позволила обслуживать большое количество прикладных программ (клиенты) одной программой (сервер). Также шло развитие межпрограммного интерфейса контейнерного типа, что позволило внедрять файлы из одной программы в другую.

Возможные варианты видов взаимодействия приложений

Конвертирование данных. Конвертирование файлов – это процесс изменения формата или свойств компьютерного файла. Для этого нужны специальные программы – конверторы – как правило, входящие в состав приложения.

OLE-технология. OLE (Object Linking and Embedding) – технология, разработанная корпорацией «Майкрософт», служит для связывания и внедрения объектов в другие объекты или документы. OLE позволяет создавать объекты в одном приложении, а после открывать эти объекты в других приложениях. Отличие между связанными и внедренными объектами заключается в месте их хранения, а также в способе обновления данных при их помещении в конечный файл. Например, при помощи OLE можно вставить таблицы Excel в документы Word, и наоборот. Приложения, поддерживающие технологию OLE, позволяют пользователю вызывать одно приложение из другого, не выходя из контекста интерфейса исходной программы. Используются принципиальные понятия «OLE-объект» и «OLE-контейнер».

Вставка данных. При обычной вставке данных другого приложения через буфер обмена или специальным образом в приложение добавляются «мертвые» данные, которые не связаны ни с каким приложением.

DDE-технология. Dynamic Data Exchange (DDE) – механизм взаимодействия приложений. Эта технология предоставляет доступ к данным через динамически действующие каналы, устанавливающие связь между приложением, принимающим данные (клиентом), и источником данных (сервером). Например, таблицы Excel могут быть источником данных для

приложения Word, принимающего данные по каналу связи DDE [4, 5].

Обычно межпрограммный интерфейс работает на распределение программного обеспечения на несколько взаимодействующих между собой уровней. Выделяют четыре уровня: базовый, служебный, системный и прикладной. Расположение этих уровней имеет пирамидальный вид. Такой вид обусловлен тем, что каждый последующий уровень базируется на программном обеспечении предыдущего уровня. Это позволяет значительно упростить работу на любом этапе, начиная с установки программы и заканчивая ее техническим обслуживанием.

Базовый уровень программного обеспечения. Это самый низкий из уровней, представляющий базовое программное обеспечение; такое обеспечение отвечает за взаимодействие с базовыми аппаратными средствами. Программные средства такого типа, как правило, входят в состав базового оборудования и хранятся в постоянно запоминающих устройствах (далее ПЗУ). Все данные записываются на ПЗУ на этапе производства, и в процессе эксплуатации их нельзя изменить.

Системный уровень программного обеспечения. Этот уровень программного обеспечения называют переходным. Программы, которые работают на этом уровне, позволяют программам базового уровня взаимодействовать с прочими программами компьютерной системы, выполняя «посреднические» функции.

От системного уровня зависят эксплуатационные показатели всей системы. Программы, которые отвечают за работу с устройствами, называют драйвера-

ми устройств. Такие программы, то есть драйвера, входят в состав программного обеспечения такого системного уровня.

Также существует класс программ системного уровня, который отвечает за работу с пользователем. Благодаря этим программам пользователь может получать возможность управлять работой системы и получать результаты в любой удобной для себя форме. Эти программные средства называют средствами обеспечения пользовательского интерфейса. От них напрямую зависит удобство работы с компьютером и производительность труда на рабочем месте.

Служебный уровень программного обеспечения. Программное обеспечение такого уровня работает с программами базового и системного уровней. Назначение этих программ заключается в автоматизации работ по настройке вычислительной системы. К тому же они используются для улучшения или расширения функций системных программ.

Библиографические ссылки

1. Купер А., Рейманн Р. М. Интерфейс. Основы проектирования взаимодействия. – М. : Питер, 2009. – 720 с.
2. Магда Ю. С. Программирование последовательных интерфейсов. – СПб. : БХВ-Петербург, 2009. – 292 с.
3. Википедия [Электронный ресурс]. – URL: <https://ru.wikipedia.org/wiki/>. – Интерфейс (дата обращения: 19.07.2016).
4. Щербо В. К., Степанов В. Н. Интерфейсы систем обработки данных. – М. : Радио и связь, 1989. – 416 с.
5. Гук М. Ю. Аппаратные интерфейсы ПК. – М. : Питер, 2002. – 528 с.