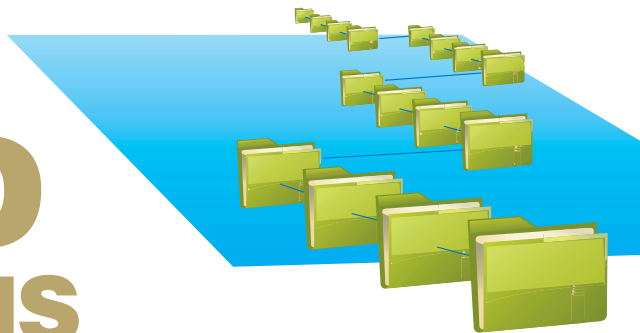


nanoCAD

в степени TDMS



Я помню, как лет двадцать тому назад впервые услышал слово *интерфейс*. Что за дурацкое слово, подумал я тогда. "Между лиц". Каких таких лиц? И некоторое время даже чурался его использовать. Сейчас я вспоминаю это время с улыбкой (а как еще можно думать о студенческих годах?), а термин полностью прижился в русском языке и используется не только в компьютерной технике и программном обеспечении, но и во многих областях науки и техники.

Термин "интерфейс" используют, чтобы описать сопряжение некоторых взаимодействующих сущностей. Чтобы не путать разные виды интерфейсов по назначению, обычно применяют более точное наименование типа сопряжения. Например, сетевой интерфейс, интерфейс пользователя, интерфейс программирования приложений². Эта статья посвящена разновидности программных интерфейсов, служащих связующим звеном между двумя информационными системами.

Где живут интерфейсы?

Как вы уже поняли, говоря "интерфейс", мы подразумеваем обязательное наличие как минимум трех сущностей. В качестве одного из программных продуктов мы возьмем платформу TDMS (разработка компании CSoft Development). С другой же стороны будем рассматривать прикладное программное обеспечение: системы автоматизированного проектирования и офисные приложения. Для каждой пары "TDMS — прикладное ПО" нам потребуется отдельный интерфейс.

Важной особенностью программного интерфейса является среда, в которой он существует. Этой средой может служить контекст TDMS, контекст прикладной программы или обособленное простран-

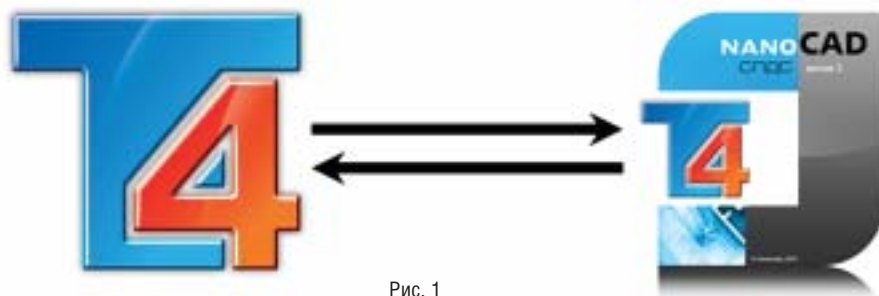


Рис. 1

ство (независимый процесс). Возможны и гибридные варианты, но они не появляются "по шучьему велению" — их использование строго обосновано. Так, мы запрягаем лошадь (устанавливаем интерфейс для управления лошадью) и берем в руки хлыст, чтобы улучшить управляемость в некоторых ситуациях. Упряжь будет находиться на лошади, а хлыст у человека. И никак не наоборот.

Как показывает опыт, наиболее эффективными и удобными являются интерфейсы, непосредственно встроенные в прикладное программное обеспечение (рис. 1). Это и неудивительно. Большую часть времени пользователь проводит, редактируя свои документы. Все новые функции, привносимые системой коллективного пользования, лучше всего иметь под рукой, чтобы выполнять их, не переключаясь из приложения в приложение. Существенным ограничением такого подхода является обязательное требование, чтобы приложение обладало программируемой средой, позволяло встроить дополнительный программный модуль и запустить его. Большинство современных программных продуктов обладает такими возможностями³. Поэтому, например, для AutoCAD или nanoCAD подобный интерфейс создать можно, а для Notepad — нельзя.

Для чего нужны интерфейсы

Основное измеряемое преимущество, которое нам дает использование интерфейса, — это повышение производительности труда.

Ускорение работы обеспечивается за счет следующих возможностей интерфейса:

- автоматическое заполнение (обновление) некоторых стандартных свойств документов. Например, в чертеже это будут данные, размещаемые в основной надписи: шифр документа, наименование проекта, наименование объекта проектирования (вида работ), номер листа и другие параметры;
- автоматизация выполнения ряда трудоемких операций — например, заполнение табличных значений, вставка вычисляемых значений, управление связями с другими документами;
- устранение негативного влияния на производительность дублирующего набора действий, обязательных при работе с системой коллективного пользования. Дополнение интерфейса возможностями системы коллективного пользования и унификация команд навигации, поиска, открытия и сохранения документа позволяют избежать потерь времени на выполнение ряда операций и необходимости переключения между приложениями.

¹ Face (англ.) — лицо, грань, передняя (наружная) сторона. На самом деле слово "face" имеет огромное количество значений, так или иначе отображающих лицевую часть чего-либо. Я этого не знал и поэтому поначалу отнесся к новому термину с подозрением.

² Интерфейс программирования приложений — Application Programming Interface (API). Раньше иногда использовался термин "прикладной программный интерфейс". Но этот перевод неверно отражает назначение такого типа интерфейса.

³ На сегодняшний день одной из наиболее распространенных и известных интегрированных сред является Visual Basic for Applications (VBA). Но среда VBA уже больше пятнадцати лет, она уже морально устарела и постепенно будет вытесняться более современной .NET framework и разработанными на ее основе инструментами Visual Studio Tools for Applications (VSTA).

Говоря об автоматизации заполнения документов, необходимо упомянуть о паре важных вещей.

Обновление свойств документов возможно и без присутствия человека: процедура может быть запущена в автоматическом режиме для произвольного количества документов и их файлов. При некоторых поворотах судьбы такая операция окажется просто неоценимой. Иногда может смениться шифр проекта или, чего доброго, по требованию заказчика будут выдвинуты новые или просто не оговоренные ранее требования к оформлению документов. Переработка документов вручную или автоматическое обновление шаблонов? А если все это происходит "как обычно", на висящем флажке?

Автоматизируя заполнение документов, мы невольно повышаем их качество. Сотрудники используют единые шаблоны документов, находящиеся в соответствии со стандартами организации, а автоматическое заполнение позволяет избежать механических ошибок, появляющихся, например, при копировании старых документов и чертежей.

Новое – это давно задуманное старое

При создании нового поколения интерфейсов мы исходили из нескольких основных требований, предъявленных пользователями в процессе эксплуатации предыдущих версий.

- Интерфейсы должны устанавливаться легко, с возможностью применения доменных политик Windows. Главной претензией администраторов больших и средних проектных

организаций были проблемы с автоматизацией установки и обновления. Самодельности здесь быть не должно.

- Интерфейсы должны быть более устойчивы в работе. Для выполнения этого требования интерфейсы к Microsoft Office, ранее написанные на VBA, были переведены на компоненты .NET.
- Интерфейсы должны легко интегрироваться в среду приложений. Это означает, что они должны быть легко подключаемы, отключаемы и настраиваемы. Чтобы решить поставленную задачу, в TDMS была добавлена панель подключения интерфейсов, в интерфейсы добавлены настройки поведения.
- Интерфейсы должны упростить взаимодействие TDMS и приложения. Если раньше дерево объектов TDMS (рис. 2) присутствовало только в интерфейсе к AutoCAD, то в новом поколении этот элемент представления информации стал единым для всех интерфейсов.

По поводу последнего требования нужно сказать особо. Если раньше для выполнения различных действий пользователю приходилось переключаться в TDMS, то новое поколение интерфейсов предоставляет значительную часть возможностей TDMS прямо в приложении. Вы полу-

чаете почтовые сообщения, отвечаете на них, берете документы в работу, завершаете и передаете дальше разработанные вами документы, не переключаясь из вашего основного приложения. Не важно, офисное это приложение или САПР. Вся требуемая функциональность вашего рабочего места TDMS присутствует на специальной панели, встроенной в приложение.

Следственный эксперимент

Чтобы лучше понять, что именно можно делать с помощью интерфейса, предлагаю рассмотреть несколько шагов в работе инженера, которые могут быть оптимизированы при использовании интерфейса.

Предположим, что исходное состояние нашего коллеги — позиция перед монитором, на всем экране которого одно из приложений nanoCAD. Если освоение средств не является приоритетной задачей руководства компании и оно не пытается "спроектировать окружающий мир", то это правильный

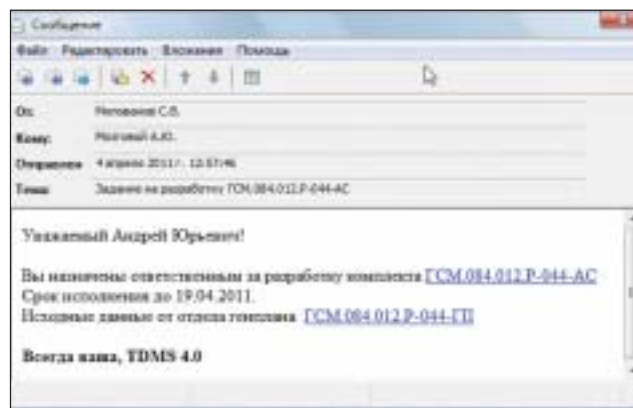


Рис. 3

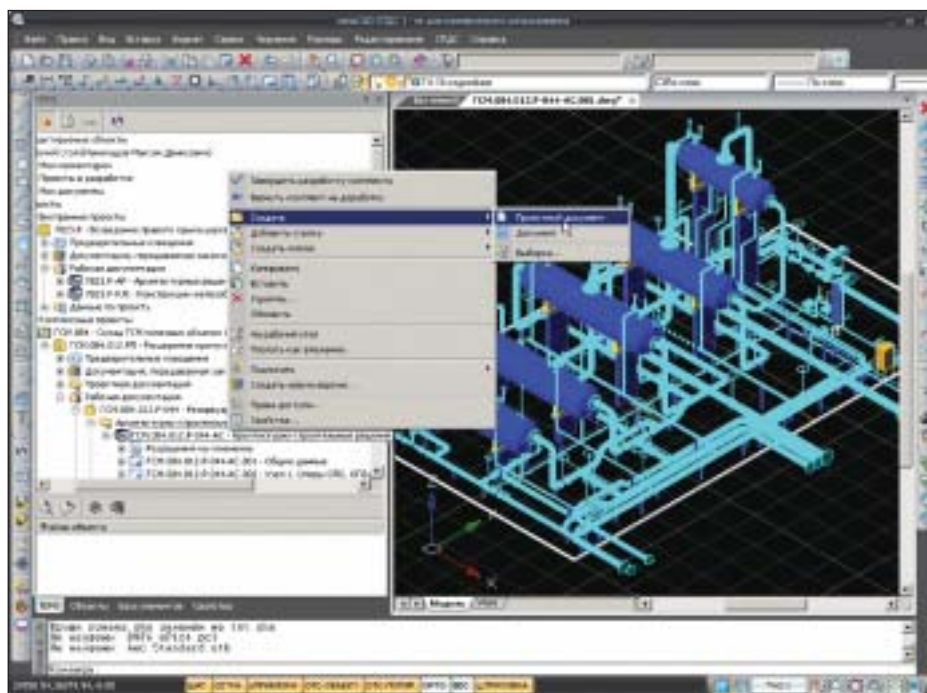


Рис. 2

выбор. Вертикальные решения, построенные на платформе nanoCAD, позволяют решать конкретные задачи, не затрачивая дополнительных средств на лейбл и "поистине безграничные возможности".

Если же вы спросите, какую версию TDMS взять, то вам предоставляется выбор между nanoTDMS 2.0 и TDMS 4.0. Пользователи nanoTDMS и TDMS знают, что эти системы используют одно и то же программируемое объектное ядро. Интерфейс, встраиваемый в приложения, работает через это ядро, а значит независимо от того, используете вы nanoTDMS Кордо 2.0, nanoTDMS Элэрос 2.0 или собственную конфигурацию TDMS 4.0, результат будет схожим с тем, который описан в данной статье.

Давайте посмотрим, чем нам поможет интерфейс, связывающий TDMS и nanoCAD.

Предположим, первое, что выполнит интерфейс — известит пользователя, что тот назначен разработчиком на комплект документов. В качестве исходных данных пользователю будет передана ссылка на задание от руководителя проекта или смежного подразделения (рис. 3).

Не переходя в TDMS, одним кликом по гиперссылке, которую он получил в письме, пользователь раскрывает нужную ветвь дерева объектов. Далее он выбирает команду *Создать/документ* (Чертеж). В открывшейся карточке документа заполняет основные свойства и закрывает ее, нажав *OK*. В дереве объектов появился новый узел с описанием вновь со-

ются значениями из карточки документа. Еще одним нововведением TDMS 4.0 является возможность вставлять атрибуты не только данного документа, но и вообще произвольные значения, полученные с помощью функции. Такими свойствами могут быть наименование проекта, наименование объекта проектирования, Ф.И.О. главного инженера проекта, логотип организации-заказчика и т.п.

Специальная синтаксическая конструкция, расширяющая стандартный набор инструкций VBScript, позволяет ввести имя функции, созданной в контексте обработчиков типа объекта, делая ее доступной для вызова из внешних прило-

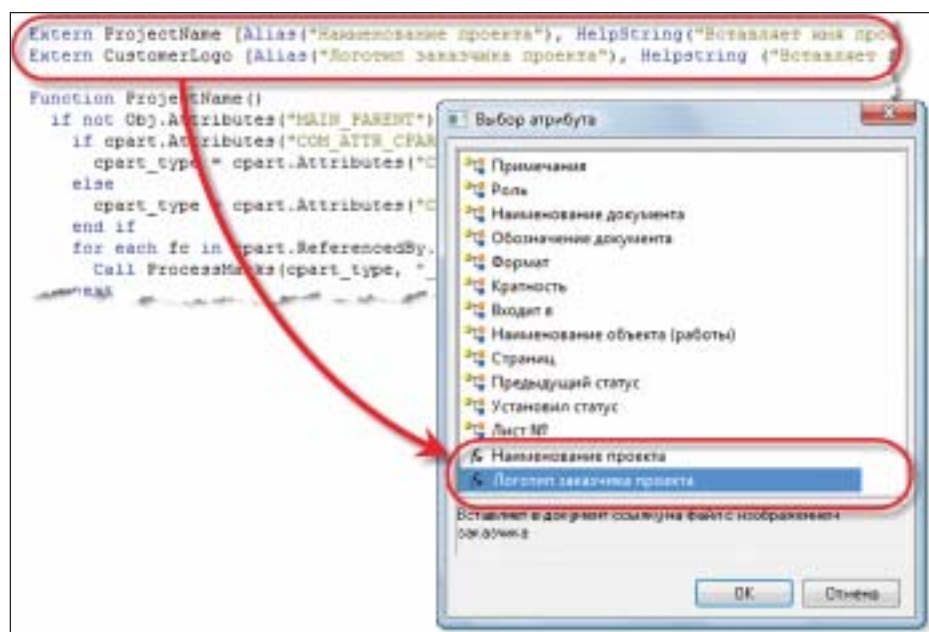


Рис. 4

зданного документа, а в списке файлов (он расположен в нижней части панели TDMS) — новый файл, созданный на основе автоматически подложенного шаблона. Двойной клик по файлу, и он открывается на редактирование в САПР. Важно отметить, что новый интерфейс с папоCAD поддерживает новое свойство файлового состава документа TDMS 4.0. Как и раньше, документ может содержать неограниченное количество произвольных типов файлов. Но с приходом TDMS 4.0 имена файлов могут содержать относительные пути загрузки. При открытии такого документа (иными словами, при загрузке файлов документа из основного файлового хранилища на локальный диск пользователя) файлы будут загружены по сохраненным путям. Это позволяет интерфейсу работать совместно с менеджерами проектов некоторых САПР, создающих иерархическую структуру хранения файлов. К таким САПР относится, например, папоCAD Электро. При открытии файла, созданного из шаблона, расставленные в нем поля заполня-

жений. Интерфейс находит все подобные функции, определенные в TDMS, и отображает их для выбора не только на естественном языке, но и со вспомогательной справочной строкой (рис. 4).

Следующим действием пользователь размещает в документе ссылку на чертеж, полученный из задания. Такой чертеж может содержать исходные данные сразу для нескольких специальностей. Например, план этажа, полученный от архитектурно-строительного подразделения, будут использовать инженеры других специальностей, чтобы провести отопление, вентиляцию, электрику и т.п. Располагают ссылочный документ на отдельном отключаемом слое, таких ссылочных документов может быть несколько. Важно отметить, что это не копия, а именно ссылка на чужой документ. В случае изменения ссылочного документа в системе пользователь незамедлительно получит уведомление об этом и одним кликом сможет обновить ссылку.

В дальнейшем, при подготовке проекта к выпуску, все документы должны быть переведены в единый формат файлов.

Электронным оригиналом должен считаться документ не в проприетарном формате DOC, DWG, DGN и т.п., а документ, переведенный в открытый и стабильный формат, — например PDF. При переводе в PDF все внешние ссылки заменяются на их содержимое или просто игнорируются.

После завершения редактирования пользователь должен сохранить свой документ. В обычной жизни, без применения TDMS, он сохранял свой файл в определенную папку, используя команды *Сохранить* или *Сохранить как...* Если не использовать интерфейс, то, чтобы сохранить отредактированные файлы в TDMS, пользователю придется сначала сохранить их на диск с помощью команд приложения, а затем перейти в TDMS и выполнить команду *Сохранить и закрыть*, которая переместит документ в файловое хранилище системы и снимет с него блокировку.

Использование интерфейса позволяет сохранить файл на диске и обновить его в TDMS одной командой. По умолчанию такая команда расположена на панели TDMS. Дополнительно пользователи могут настроить интерфейс таким образом, чтобы команды приложения тоже выполняли оба действия. Кроме того, при закрытии документов интерфейс проверит, были ли они обновлены в TDMS, и, если этого не сделано, предложит вам выполнить данную операцию.

В сухом остатке

Кроме наиболее очевидного довода для использования интерфейса — *повышения скорости работы пользователя*, — мы обнаружили ряд бонусов:

- *повышение качества получаемых документов.* Этому способствует и автоматизация выполнения ряда операций, необходимость которых не всегда может быть вовремя установлена пользователем, и возможность выполнить такие операции в его отсутствие;
- *повышение удобства работы.* Уменьшение количества рутинных операций способствует снижению утомляемости. Пользователь концентрируется на выполнении основной работы, а не на связанной с ней "бюрократии" и навязанной необходимости переключаться между приложениями.

Использование нового поколения интерфейсов дает синергетический эффект одновременного применения двух программных продуктов. Вот это я и называю папоCAD в степени TDMS.

Сергей Загурский
E-mail: serge@csoft.ru