

Почему так трудно работать с САД- и САЕ-системами, и как сделать их проще

L.Stephen Wolfe, P.E. (CAD/CAM Net)

©2006 Cyon Research Corporation

Я считаю, что инженеры тратят слишком много умственной энергии для того, чтобы заставить программное обеспечение (ПО), с которым они работают, делать то, что им надо. Эта борьба осложняет инновационную деятельность проектировщиков в их основной области – в разработке изделий. Неудивительно, что ведущие конструкторы, такие как *Burt Rutan* (см. *Observer* #3/2005), *Frank Gehry*, *Richard Seymour*, *Dick Powell* делают наброски своих идей при помощи простейших инструментов, а затем передают их для детальной проработки другим сотрудникам, хорошо освоившим системы САД и САЕ.

В этом году на конгрессе **COFES** (*Congress on the Future of Engineering Software*) мы собрали полную аудиторию пользователей и разработчиков инженерного софта с целью разобраться в причинах, по которым эти программные продукты остаются настолько сложными в применении, а также поискать способы, как сделать их более простыми и удобными. В этой статье я попытаюсь обобщить высказанные мысли и предложения.

Сегодня САД- и САЕ-системы требуют, чтобы конструктор владел тысячей специальных инструментов для моделирования всевозможных объектов. Каждый специальный случай должен быть изучен и запомнен. Каждый имеет свои

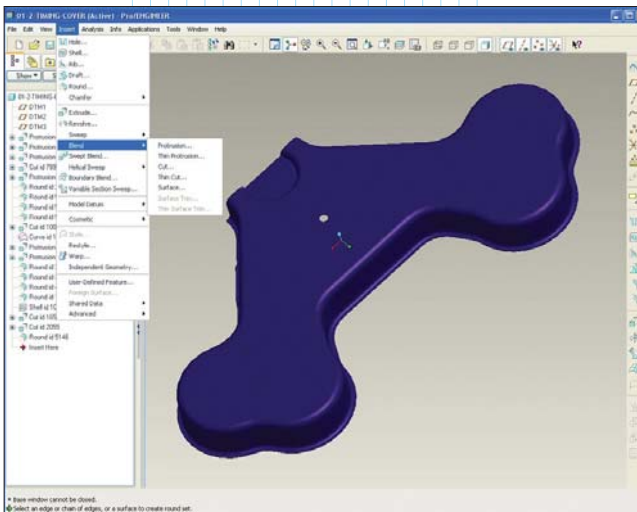


Рис. 1. Более пяти лет PTC внедряла методы, предписанные гуром проектирования г-ном Donald Norman, чтобы упростить работу в Pro/ENGINEER Wildfire. Однако пользователи до сих пор вынуждены оперировать многими разрозненными функциями, чтобы построить сложную 3D-модель

особенности и ограничения. Например, в большинстве систем твердотельного моделирования есть два типа команд *Loft*, *Sweep*, *Revolve* и *Extrusion*: один для твердотельных конструктивных элементов (КЭ), а другой – для поверхностных. При этом интерфейс у этих двух типов команд различный. И у тех, и у других бывают сбои в работе, однако условия, при которых это происходит, отличаются. В конце концов, действуя методом проб и ошибок, конструктор находит тот тип КЭ, который должен быть использован для создания желаемой формы, но процесс изучения оказывается небыстрым, недешевым и несчастливым. В общем, не так уж много людей (если таковые есть вообще) может овладеть в полной мере всеми функциями сегодняшних САД- и САЕ-систем (рис. 1).

Методы построения далеко не всегда очевидны (рис. 2). Форма детали, построенной при помощи большинства САД-систем на базе КЭ, зависит от последовательности применения этих КЭ.

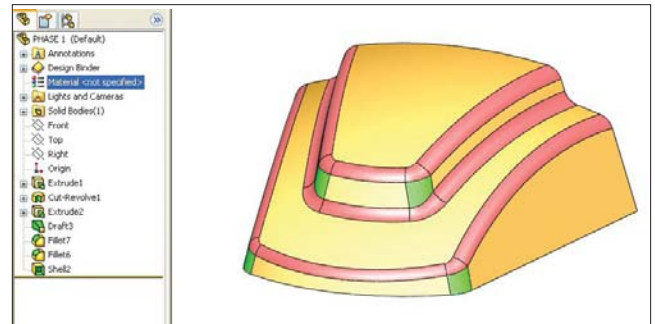
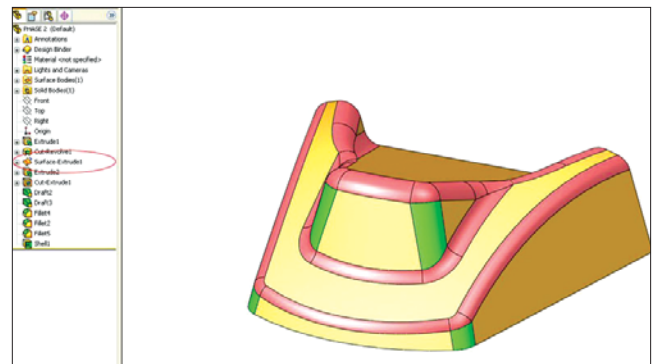


Рис. 2. Как показывает этот пример, заимствованный из материалов конкурса SolidWorks 2004 "Model Mania", методы построения не всегда очевидны. Чтобы изменить форму этой детали, участники должны были откатиться по дереву построений до операции Cut Revolve (отмечена красным кружком на нижней картинке) и вставить поверхность



Изменение порядка ведет к изменению формы. В некоторых случаях, для того чтобы вставить желаемый КЭ, придется откатиться назад по дереву построений. Удивительно, но десятки тысяч конструкторов освоили эту технику...

Наличие крошечных невидимых артефактов в CAD-модели часто приводит к проблемам при использовании CAM-систем и при инженерном анализе. В частности, этим опасны все перечисленные ниже случаи:

- неправильная обрезка в эскизах или поверхностях;
- геометрия привязана к граням, которые исчезают после изменения образующих их КЭ;
- имеются невидимые элементы – например, оси КЭ, построенных при помощи команды *Revolve*;
- очень маленькие поверхности (часто их называют осколками), созданные программой при построении сложных переходов (*Blend*) между поверхностями.

В этих, а также и в других подобных ситуациях, попытки построить скругления и переходы между поверхностями могут приводить к ошибкам (рис. 3). Наткнувшись же на дефект CAD-модели на следующем этапе создания изделия, система может прекратить построение сетки конечных элементов или расчет траектории инструмента. При этом конструктор, породивший проблему с геометрией, может даже не знать об этом. С другой стороны, специалист по инженерным расчетам или технолог-программист, столкнувшийся с такой проблемой, может не знать путей её устранения.

Современные системы дают возможность конструкторам описывать отношения между КЭ в детали или между деталями в сборке различными способами. При должном использовании эти отношения позволяют автоматически изменять размеры деталей проектируемого изделия, сохраняя первоначальный замысел при создании нового варианта. Однако способов описать отношения в CAD-системах предлагается так много, что конструктор может выбрать неправильный. Тогда при изменении размеров детали могут появиться случайные ошибки, что будет

иметь пагубные последствия. Особенно неприятны связи между гранями КЭ или деталей, поскольку при изменении формы эти грани могут исчезнуть, что послужит причиной ошибок регенерации.

Применение связей в сборках ставит новые проблемы перед инженерами, которые хотели бы использовать CAD-модели при моделировании физических процессов (динамика, напряжения, перемещения, вибрации). Конструкторы часто накладывают в сборках избыточные связи, что исключает подвижность деталей, присущую реальному механизму. В результате это ухудшает работу модуля моделирования кинематики – одного из важнейших приложений современного 3D-проектирования. Поскольку ограничения могут быть внесены как явно, так и неявно, различными способами, то приведение наложенных на модель ограничений в соответствие с реальной кинематикой механизма может оказаться сложной и требующей больших затрат времени задачей.

Кроме того, связи в сборочных моделях, используемые проектировщиками для построения, часто не соответствуют тому, что необходимо для моделирования движения. Такая ситуация часто вынуждает специалистов по инженерному анализу переопределять связи в сборках, прежде чем начинать изучение их свойств.

Разработчики CAD-продуктов слишком часто не понимают в достаточной мере процессы проектирования и производства. Когда поставщики программного обеспечения вводят новые модули, нацеленные на автоматизацию специфических процессов (таких как сварка, проектирование изделий из листовых материалов, прессформ, кабелей) или на индустриальный дизайн, то в них обычно недостает некоторых инструментов, необходимых для полного решения задачи. Например, в первых модулях для проектирования металлоконструкций не было полных библиотек стандартных профилей и возможности моделировать гнутые балки. Подготовка ведомости заготовок (перечня всего металлопроката с указанием длин) в ранних версиях требовала слишком много шагов.

Когда пользователи сталкиваются с подобными недостатками, они часто откладывают начало реального применения такого продукта до выхода следующей версии, бесполезно потратив время на попытку освоения новых возможностей. Сырые разработки являются причиной того, что клиенты тратят много времени впустую, в результате чего их доверие к поставщику ПО снижается. Более глубоко проработанные решения позволили бы избежать фальстартов и создали бы предпосылки для быстрого освоения клиентами новых возможностей.

Руководство по использованию системы зачастую бывает написано недостаточно ясно, и в нем не хватает примеров – особенно это касается новой функциональности. Набор ключевых



Рис. 3. По словам представителя службы технической поддержки Autodesk, из-за таких необрезанных элементов эскиза Inventor 11 впоследствии не позволяет делать скругления

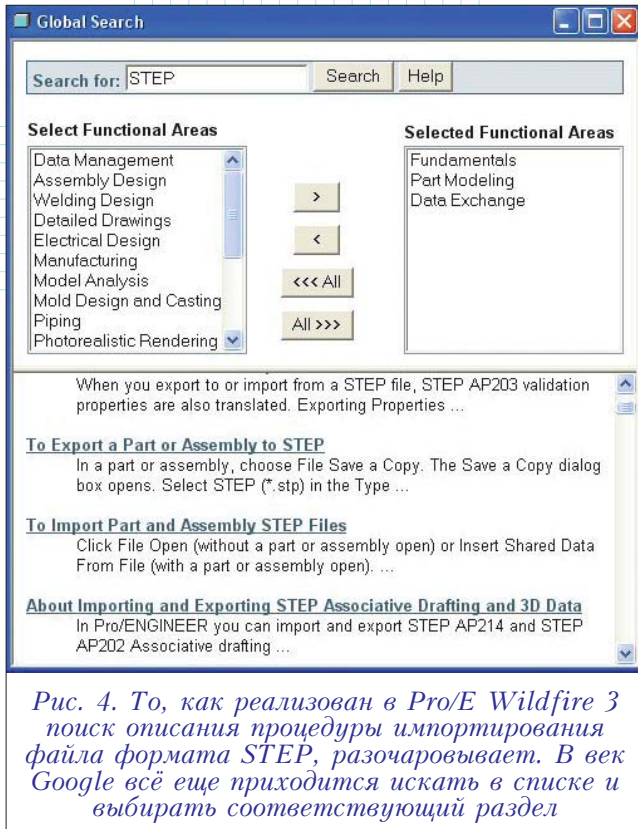


Рис. 4. То, как реализован в Pro/E Wildfire 3 поиск описания процедуры импортирования файла формата STEP, разочаровывает. В век Google всё еще приходится искать в списке и выбирать соответствующий раздел

слов в индексах системы помощи часто бывает неадекватен, так что конструктор не может быстро найти инструкции для функций, которые он хочет изучить и опробовать (рис. 4). Отсутствие терминов и описаний, которые были бы стандартными для разных систем, создает проблемы для конструкторов, знакомых с одной системой, при изучении другой – даже притом, что концепции большинства современных систем подобны.

Нередко от версии к версии происходят незначительные на вид изменения в расположении инструментов управления, но это сильно мешает пользователям работать. Неспособность системы помощи показать конструктору, где произошли эти изменения и как использовать новые и улучшенные инструменты управления, разочаровывает пользователей и отбивает желание переходить на новые версии.

Сообщения об ошибках и предупреждения дают проектировщикам мало ключей к пониманию того, что может быть неправильно в их моделях (рис. 5). В результате многие пользователи теряют не один час рабочего времени на том, что воспринимается ими как ограничения, накладываемые программой, хотя фактически дефект модели может быть легко исправимым. Кроме того, попытки обойти подобные проблемы часто делают модель еще более сложной для понимания другими людьми.

Нестабильность остается хронической проблемой для большинства CAD- и CAE-систем.

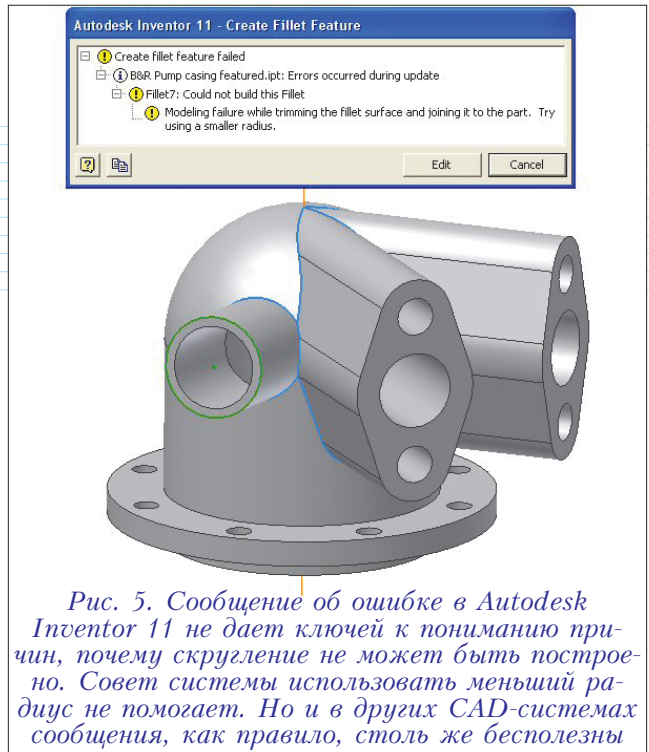


Рис. 5. Сообщение об ошибке в Autodesk Inventor 11 не дает ключей к пониманию причин, почему скругление не может быть построено. Совет системы использовать меньший радиус не помогает. Но и в других CAD-системах сообщения, как правило, столь же бесполезны

Каждый год клиенты теряют миллионы рабочих часов из-за краха приложений. Поставщики ПО не способны отличить проблемы, вызываемые компьютерной системой (например, графическим драйвером или операционной системой), от проблем, вызванных ошибками в их приложениях.

Медленная работа CAD-системы при загрузке модели и внесении изменений подрывает производительность. Большинство систем не позволяет прервать загрузку или регенерацию модели клавишей Esc.

Обмен данными и коллаборативность сдерживаются тем фактом, что современные системы не могут достаточно надежно читать модели или чертежи, созданные в других системах (рис. 6). Пользователи тратят свое время, исправляя или

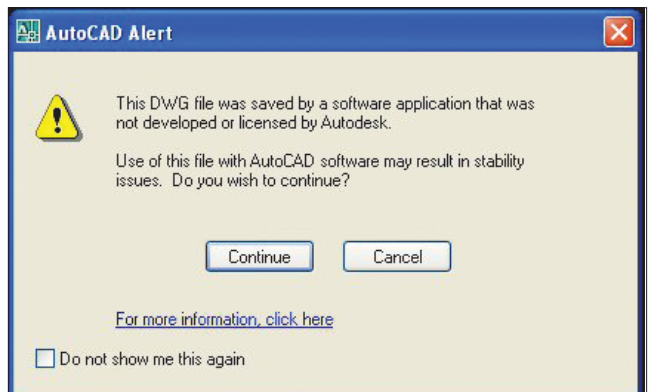


Рис. 6. Жуткие сообщения, подобные этому, заставляют пользователей AutoCAD 2007 открывать DWG-файлы, созданные в других системах

создавая заново 3D-модели, построенные в иной среде. В результате на практике трудно использовать модели, созданные поставщиками или подрядчиками, если те применяют какой-то другой программный инструмент.

Негибкая лицензионная политика и необходимость администрирования лицензий наказывают честных клиентов. Администрирование множества лицензий от разных поставщиков может стать непродуктивным занятием на полный рабочий день.

Какие же шаги пользователи и поставщики инженерного софта могут предпринять, чтобы сделать его более простым и менее дорогостоящим в эксплуатации?

Как сделать CAD- и CAE-системы более удобными

Коренная причина, по которой программное обеспечение столь сложно, состоит в том, что компьютеры сами по себе – тупые. Они не могут наблюдать, делать выводы или приобретать полезный опыт. Поэтому, чтобы моделировать бесчисленное количество форм, необходимых в современных изделиях и устройствах, программисты придумали сотни (или даже тысячи) процедур для различных вариантов создания геометрии. Конструктор же должен разбираться в огромном количестве ситуаций, чтобы моделировать сложные изделия.

Когда в с CAD-системе имеется 2 000 команд, у этой проблемы не может быть никакого технического решения. Большинство CAD-систем предлагает нам сумбурную смесь методов твердотельного и поверхностного моделирования, разработанных такими компаниями, как *Manufacturing and Consulting Services*, *Shape Data* и *Parametric Technology*. Попытки отойти от этой базы и реализовать более общий подход встретили весьма ограниченную поддержку на рынке систем для профессионального проектирования.

В этой ситуации поставщики софта и их клиенты могли бы принять меры для того, чтобы сделать системы CAD и CAE более легкими в изучении и использовании. Вот некоторые из идей, услышанных автором во время сессии форума *COFES 2006*.

Что могут сделать клиенты

Клиенты должны корректно объяснить поставщикам программного обеспечения, почему их продукты неоправданно громоздки. Я был на многих встречах с пользователями, где слышал фразы “Этот полуфабрикат ужасает!” или “Иногда в процессе моделирования я строю то, что могу построить, а не то, что хочет заказчик”. Такие замечания передают эмоции и разочарование проектировщиков, но мало что делают для того, чтобы программы стали лучше.

Пользователи должны обращать внимание разработчиков на недостатки ПО, основываясь

на конкретных примерах из реальной жизни. Например, если невозможно сделать надлежащий уклон стенки литой детали, отправьте эту модель поставщикам вашей CAD-системы, сопроводив её запиской с пояснением, что именно нельзя сделать и почему это сделать необходимо. Если для того, чтобы поставить размер или переориентировать эскиз, требуется слишком много действий, то поясните разработчику письменно, как этот процесс может быть упрощен.

Если система диалоговой помощи или руководство пользователя неясны, процитируйте страницу и объясните, как их можно улучшить. Например, иллюстрации, показывающие различные варианты работы *Loft* или *Round*, могут облегчить пользователю понимание опций в этих командах.

Пользовательские группы дают клиентам возможность сравнить свои представления о функциях, которые, по их мнению, являются неэффективными или неадекватными, и выработать какие-то рекомендации. Пользователи должны организовать комитеты для улучшения автоматизации конкретных процессов, таких как создание чертежей, разработка пресс-форм, индустриальный дизайн, проектирование машиностроительного оборудования, проектирование изделий из листовых материалов. На каждой встрече должен работать, по меньшей мере, один компьютер с соответствующим программным обеспечением, чтобы члены сообщества могли реально иллюстрировать обнаруженные проблемы и предлагать решения. Когда комитет приходит к согласию, он должен представить рекомендации в письменном виде и с иллюстрациями.

Пользователи должны охотно делиться с поставщиками софта тонкостями и особенностями процесса своей работы. Выгода, которую они получают от улучшения ПО, несомненно, превышает риск того, что конкуренты смогут узнать что-то, чего они еще не знают. Объясните, как и почему вы делаете то, что делаете; покажите, как инженерные данные распространяются по вашей компании вплоть до производственных отделов и поставщиков. Слишком часто клиенты просят о вводе каких-то новых функций, не объясняя, для чего они им нужны, и что они пытаются сделать. И когда разработчик реализует это, зачастую оказывается, что получилось не совсем то, чего ожидали...

Выбирая систему, не стоит формально составлять списки функций и отмечать те, которые, по утверждению продавцов, есть в той или иной системе. Такой подход поощряет включение в программы недоработанных функций, которые будут лишь впустую отнимать ваше время. Вместо этого настаивайте, чтобы новое программное средство могло эффективно автоматизировать весь ваш процесс – от концепции до конструкции или производства. Если что-то недоработано, дайте ясно понять, что вы не будете покупать этот софт до устранения всех недоделок.

Если вся эта активность выглядит как гора работы, значит так оно и есть. Однако альтернатива здесь такова: получить плохо разработанное и скверно документированное программное обеспечение. Фирмы, которые грамотно и продуманно побуждают разработчика к усовершенствованиям, должны видеть осуществление своих идей раньше других. Если это не так, то, может быть, пришло время подумать о более отзывчивом поставщике софта.

Что могут сделать разработчики ПО

В CAD-компаниях должно быть больше специалистов, проектирующих программное обеспечение, и меньше тех, кто просто пишет код. Этот пункт был четко сформулирован и исчерпывающе обоснован г-ном **Alan Cooper** в его книге *“The Inmates are Running the Asylum”*. В большинстве CAD-компаний менеджеры по продукту пишут спецификации для новых функций, но фактически всё делают программисты. Однако, они больше разбираются в компьютерных тонкостях (и это правильно!), чем в потребностях инженеров и конструкторов.

Г-н *Cooper* доказывает, что разработчики софта должны фокусироваться на потребностях клиентов. Им следует проводить с клиентами много времени, чтобы понять их потребности и применяемые ими процессы до того, как поставить задачу детальной разработки кода.

Для повышения надежности создатели CAD-систем должны также уделять больше времени внутренним механизмам работы программ. Современное программное обеспечение CAD и CAE не построено так, чтобы сбои и ошибки легко обрабатывались. Одна неправильная ссылка по адресу может привести к краху всего приложения. Однако, такой исход может быть вызван и внешней причиной – например, ошибкой в графическом драйвере или сбоем операционной системы. Недостаточно защищенная архитектура не позволяет надежно локализовать сбои, связанные с аппаратной частью.

Прошли времена, когда сообразительный программист мог написать полезную или просто забавную программу, быстро вывести её на рынок и стать миллионером (или миллиардером). Сегодня индустрия программного обеспечения вступила в стадию зрелости, и на выпуск нового продукта нужны миллионы долларов. Стандарты авиационной безопасности времён братьев Райт нынче уже неприемлемы. Точно также неприемлемы “быстрые и грязные” методы программирования, которые были в ходу на заре индустрии ПО.

Конструкторы реальных изделий перед запуском в серию изготавливают опытные образцы и проверяют их на потенциальных клиентах. Разработчики софта редко делают нечто подобное. Вместо этого “опытный образец” в полузаконченном состоянии сбрасывается клиенту, чтобы не были нарушены объявленные маркетинговые планы.

Фирмы, разрабатывающие ПО, должны написать прототипы новых функций и проверить их на потенциальных клиентах, чтобы убедиться, что процессы проектирования и производства выполняются должным образом. Когда эта работа сделана, программисты должны переписать программы, обращая внимание на производительность и надежность кода.

Чтобы упростить работу пользователя, следует группировать на экране те инструменты, которые необходимы для конкретного процесса. Лишние в данный момент инструменты, соответственно, должны быть скрыты. Большинство CAD-компаний реализует это с помощью программ-мастеров или инструментальных панелей. Однако многие такие интерфейсы следуют за общим функциональным разделением системы – поверхностное моделирование, твердотельное моделирование, создание чертежей, моделирование сборки...

Более полезной следует признать такую организацию взаимодействия с пользователем, когда интерфейс следует за значащими для инженера процессами – например, проектирование изделий из листовых материалов, литых деталей, пресс-форм, форм для литья, авиационных конструкций, кузовов автомобилей, промышленного оборудования, изделий из железобетона, стальных конструкций, электрических систем... На рабочем столе конструктора изделий из листовых материалов или проектировщика металлоконструкций не требуются поверхности или сплайны, но нужны базовые средства рисования

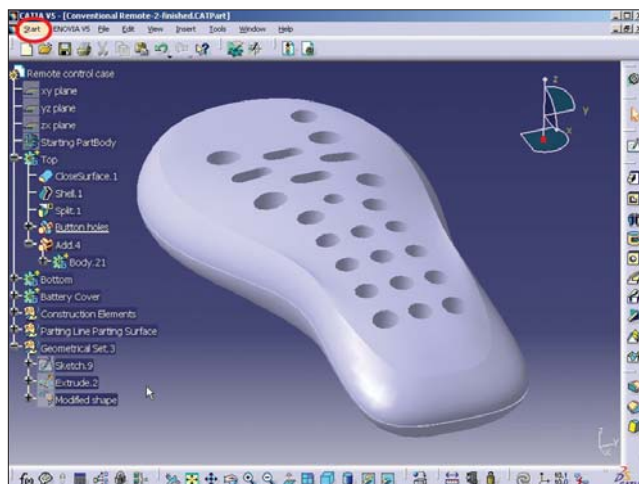


Рис. 7. Чтобы уменьшить количество иконок на экране, компания Dassault Systèmes в пакете CATIA V5 применяет концепцию автоматизированных рабочих мест (workbench) для разных процессов. На иллюстрации показан внешний вид АРМ для проектирования деталей, изготавливаемых литьём. Справа расположены специальные инструменты для разработки пластмассовых деталей. Конструктор может выбрать нужное рабочее место с помощью пункта меню “Start” в левом верхнем углу

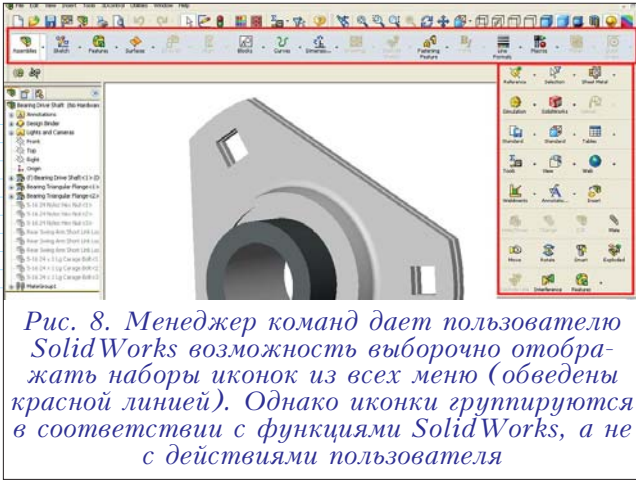


Рис. 8. Менеджер команд дает пользователю SolidWorks возможность выборочно отображать наборы иконок из всех меню (обведены красной линией). Однако иконки группируются в соответствии с функциями SolidWorks, а не с действиями пользователя

эскизов и возможности моделирования сборок. Введя понятие автоматизированных рабочих мест (*workbenches*), команда разработчиков CATIA V5 из Dassault Systèmes продвинулась в развитии этой концепции дальше, чем большинство других CAD-компаний (рис. 7).

Пользователям нужны и средства настройки интерфейса рабочих мест (рис. 8), возможность добавлять функции, не включенные в основную конфигурацию. Настроенные конфигурации должны сохраняться на жестком диске таким образом, чтобы скачанные патчи и обновления версий не уничтожили прежние настройки.

Поскольку сегодняшние программные продукты очень сложны, никто не может изучить все их функции, только сидя в учебном классе. Кроме того, конструктор, хорошо знающий одну систему, должен иметь возможность изучить эквивалентные функции других систем в кратчайшее время. Например, инженер с опытом в Pro/ENGINEER может поступить на работу в компанию, использующую Solid Edge. Он должен знать, как рисовать эскизы, строить конструкционные планы, создавать сложные элементы (к примеру, поверхности командами *Sweep* и *Loft*). Эти функции есть в обеих системах, но управление ими весьма отличается. Система диалоговой помощи должна давать пользователю Pro/ENGINEER возможность быстро найти процедуры, необходимые для выполнения аналогичных задач в Solid Edge.

Сейчас большинство онлайн-систем помощи мало способствует изучению пользователями основных функций программного обеспечения (рис. 9). Зачастую в них нельзя найти все термины, используемые в меню или диалоговых окнах. А индексный поиск даже по простым фразам в help-файлах CAD-систем невозможен вообще. Вместо этого открываются страницы со ссылками, в которых упоминается каждое из слов фразы.

Наконец, поставщики ПО должны изучить, как создаваемые пользователями программных систем данные распространяются внутри компании: от

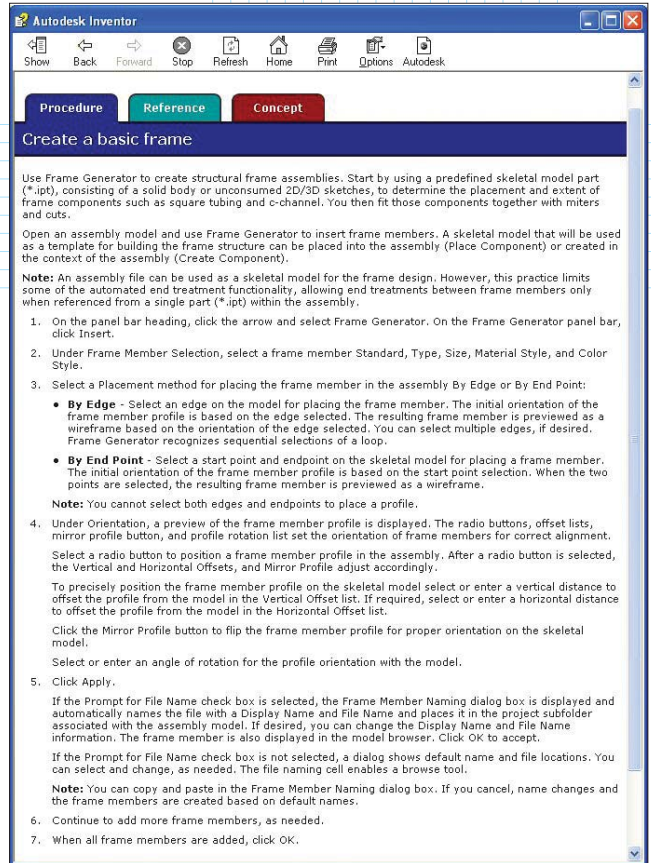


Рис. 9. В описании процедур использования модуля "Frame generator" (новая, важная функциональность Inventor 11 для проектирования металлоконструкций) не хватает иллюстраций. Поставщикам софта следует тратить больше времени на составление руководства пользователя и другой документации, чтобы их клиенты могли научиться применять новые инструменты

концепции, через проектирование, инженерный анализ, оптимизацию, изготовление комплекта инструментов, планирование ресурсов, вплоть до сборки. Это сложные процессы, и большинство клиентов применяет для их автоматизации программное обеспечение от разных поставщиков. Неэффективные процедуры и ошибки, возникающие при попытке использования данных из разных источников, увеличивают затраты на эксплуатацию CAD- и CAE-систем. Высокие затраты, в свою очередь, снижают спрос на программное обеспечение.

Не будет ли всё это дополнительное внимание к проектированию стоить слишком дорого? Г-н Cooper говорит, что нет. Более качественное программное обеспечение требует меньших затрат на поддержку, а сэкономленные средства могут быть потрачены на дополнительный маркетинг и, следовательно, на увеличение продаж.

В заключение, хочу выразить благодарность всем участникам COFES, которые своими идеями способствовали появлению этой статьи. ☺