

ХОЧЕШЬ ЗАНИМАТЬСЯ ДИСТАНЦИОННЫМ ОБУЧЕНИЕМ – СДЕЛАЙ СВОЙ ПАКЕТ УДОБНЫМ ДЛЯ УЧИТЕЛЯ

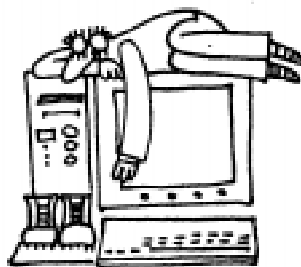
Что мешает хорошим программам для моделирования использоваться на практике не только их разработчиками? Во многих случаях причина в нежелании учителя подстраивать свои собственные методики под чужую компьютерную программу и вечный поиск достаточно гибких приложений для того, чтобы полностью удовлетворить свои потребности. Если программа предназначена для работы только на одном типе машин, то это тоже мешает ее распространению. Исходя из этого, авторы предприняли попытку преодолеть сразу обе трудности, взяв в качестве примера уже готовый, тщательно спроектированный программный продукт Phasor, проверенный на протяжении последних нескольких лет не на одном поколении студентов. Phasor был переписан заново для работы в новой Многофункциональной Среде Моделирования. В этой среде можно будет, как и обычно, использовать компьютерные модели и для лабораторных работ и для демонстраций на лекциях, но в то же время их интеграция с гипермедийными оболочками делает доступными заочное и дистанционное обучение. Благодаря возможности настраивать программу с учетом потребностей учителя и его стиля преподавания, обеспечена ее гибкость. Программный код модели и ее оболочки независимы друг от друга. Для того, чтобы упростить использование Web-технологий, модель написана на языке Java. Если мы сможем обеспечить проведение вычислительного эксперимента в той форме, в какой его хотят видеть учителя, то основное препятствие, мешающее более широкому использованию программных средств в обучении, будет преодолено.

Не вызывает сомнения, что компьютерные эксперименты являются важными наглядными средствами обучения. Члены ассоциации CoLoS разработали и используют много прекрасных программ для моделирования, но почему же так мало других пакетов получило широкое распространение? Даже если пакеты предназначены для обучения, то все равно они используются не столь широко, как того хотелось бы их авторам. В чем же причины невостребованности? Основным на сегодняшний день является нежелание учителей пользоваться пакетами, разработанными их коллегами-учителями. Бывает, и пакет хорош, да он все-таки чуть-чуть не такой, как хотелось бы учителю. А если он и настраивается, то сама процедура настройки для получения желаемого эффекта может оказаться чрезвычайно скучной.

И если мы хотим что-либо поменять в чужом пакете, даже столь малое, как цвет, но не можем этого сделать, то это весьма неблагоприятно сказывается на решении использовать пакет. Таким образом, если мы хотим извлечь максимальную пользу из наших пакетов и сделать их широко используемыми другими, то мы должны сделать их легко приспособляемыми к требованиям конкретного учителя.

Запросы, предъявляемые к пакетам для моделирования, весьма высоки. Прежде всего хочется спросить, а какая польза от них учащемуся? В различных статьях высказываются мнения, что вычислительный эксперимент помогает осваивать материал в процессе активных действий и «извлекать теорию из эксперимента» [1], что в исследовательской деятельности вычислительные эксперименты «способству-

ют усилению заинтересованности в углублении исследований» [2]. Хартел [3] и Клоус [4] пытались исследовать эффективность применения компьютерного моделирования в обучении и, понимая, что строго научный подход здесь труднодостижим, все-таки пришли к выводу, что вычислительный эк-



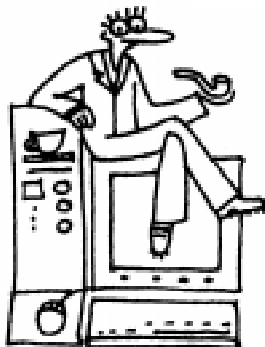
тельной и гибкой, насколько это возможно. Хотя данное желание и похвально, но оно, в свою очередь, осложняет настройку пакета и его использование, а это приводит к тому, что сужается круг его потенциальных пользователей. Для сти-

мулирования использования компьютерных моделей в обучении неоднократно предлагалось воспользоваться гипермедийными и мультимедийными средствами. Нельсон и Томас [7, 8] рассматривали влияние сетевых технологий на обучение и показали, что, по мере их распространения, роль компьютерных моделей будет возрастать, в частности, в области дистанционного обучения студентов.

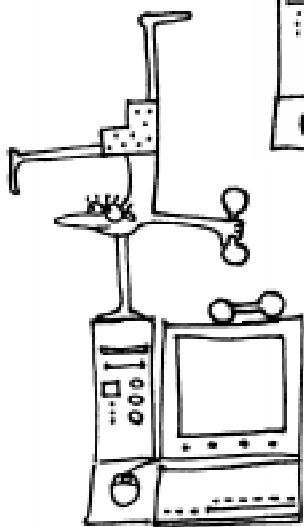
МНОГОФУНКЦИОНАЛЬНАЯ ОБОЛОЧКА

Многофункциональная Среда Моделирования была разработана с целью сделать компьютерное моделирование столь же простым в использовании, как и другие средства наглядного компьютерного обучения, и предоставить человеку, не знакомому с программированием, уникальные возможности управлять ходом эксперимента. В описании управляющей программы (пакет разработчика) нет жестких предписаний, как использовать программу в учебном процессе, нет и ограничений на форму, в какой модели предстанут перед пользователем. Управляющая программа написана на языке Java, что делает ее доступной для всех основных платформ. Ряд ее свойств делают ее максимально гибкой:

- Программа имеет распределенную архитектуру, что позволяет и разрабатывать модели и выполнять их в сети. Компьютерная модель и пользовательский интерфейс независимы друг от друга. Каждый компонент, если в том есть необходимость, может работать в сети на разных машинах. Это позволяет использовать различные режимы моделирования: однопользовательский режим, когда модель и интер-

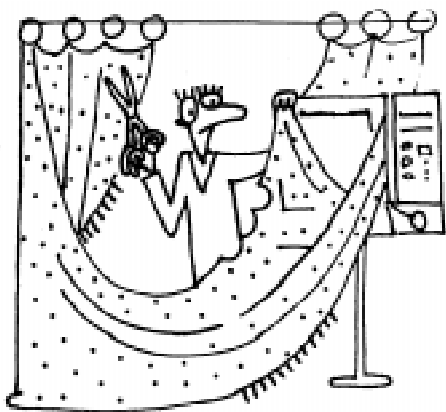


Благодаря возможности настраивать программу с учетом потребностей учителя и его стиля преподавания, обеспечена ее гибкость...



сперимент приносит пользу. Однако за любые преимущества надо платить. Учителя вынуждены писать руководства, как проводить вычислительный эксперимент, и предвосхищать любые недоразумения, проистекающие из того, что в таких руководствах невозможно охватить весь нужный материал. Реальные условия таковы, что разработчикам не выгодно создавать хорошие интерактивные пакеты, так как они требуют много времени и денег. Все это привело к тому, что компьютерные модели использовались в обучении менее эффективно, по сравнению с более простыми и статичными программами для поддержки учебных курсов [6].

Пытаясь добиться того, чтобы приложение оправдывало затраченные на его разработку усилия, авторы решили идти до конца по намеченному пути и сделать компьютерную модель настолько вырази-



...разработана с целью сделать компьютерное моделирование столь же простым в использовании...

фейс работают на одной машине, многопользовательский доступ, когда модель запускается на одной машине, а пользователи совместно получают к ней доступ со своих машин.

- Интерфейс пользователя может меняться в широких пределах. Если компьютерная модель написана так, что она совместима с управляющей программой, то все переменные модели доступны в пользовательском интерфейсе. Разработчик интерфейса сам может определить, видна или не видна та или иная переменная конечному пользователю, указать, можно ли ее редактировать, выбрать способ ее представления и ее образ на экране. Например, целочисленная переменная, может быть представлена как движок, как текстовое поле, измерительным прибором или аналоговым измерителем. Размещенный в окне объект можно редактировать (менять его размеры, расположение, цвет, шрифт). Свойства отображаемого объекта можно связывать с переменными модели. Так, например, разработчик интерфейса может выбрать границы изменения движка, в зависимости от некоторой переменной модели.

- Использование переменных модели можно комбинировать с другими доступными средствами. Так, к пользовательскому интерфейсу могут быть добавлены образы и текст. Конфигурация модели может запоминаться и, если нужно, восстанавливаться.

Это можно сделать и на одной машине и в сети. Можно добавлять управляющие кнопки, связывающие модели с различной конфигурацией в единое целое.

Многофункциональная среда обеспечивает два режима работы:

1. *Редактирование проекта* – разработчик (учитель или создатель курса) строит интерфейс. Модель может запоминаться как апплет, способный работать под управлением программы, демонстрирующей Web-страницы (web browser). Если разработчик хочет использовать другие мультимедийные ресурсы в сочетании с моделированием, их можно включить в интерфейс или, если интерфейс был сохранен как апплет, добавить в форме HTML на ту же Web-страницу.

2. *Вычислительный эксперимент* – конечный пользователь (ученик или учитель) запускает модель с ранее созданным интерфейсом либо под управлением Многофункциональной Среды, либо под управлением программы, демонстрирующей Web-страницы.

Из предшествующего короткого рассмотрения уже видно, что у программы достаточный потенциал для создания собственного пользовательского интерфейса и удовлетворения нужд конечного пользователя за счет включения в оболочку средств, повышающих эффективность моделирования. Мы покажем, как можно использовать новый подход, на примере нашей конкретной программы Phasor.

ПЕРВОНАЧАЛЬНЫЙ ВАРИАНТ ПРОГРАММЫ PHASOR

Phasor – динамическая, интерактивная программа для моделирования, которая была предназначена для работы под управлением операционных систем Unix и MS Windows. Она применялась как учебная программа на кафедре Программирования и Электроники университета Хериот-Ватт [9] для студентов различных курсов. Ее интерфейс нельзя изменить, и существует только один, жестко фиксированный, способ ее использования. Но

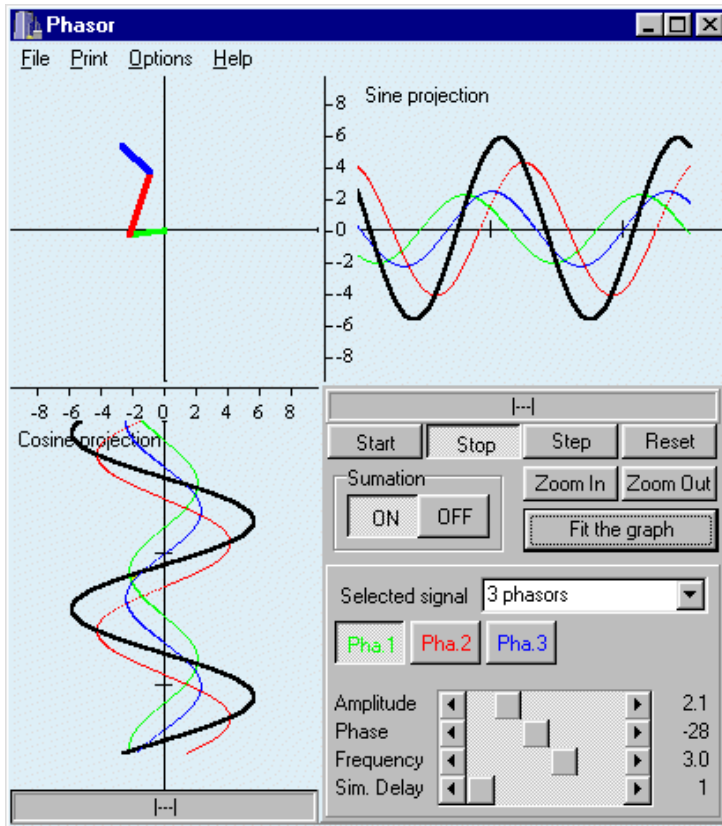


Рисунок 1. Интерфейс исходной программы Phasor, работающей под управлением MS Windows

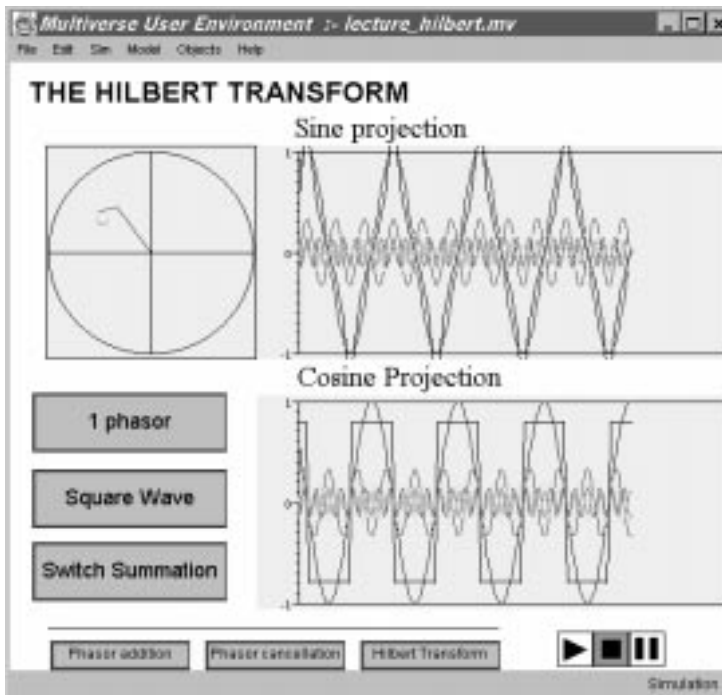


Рисунок 2. Лекционные слайды

даже при этих ограничениях Phasor – очень полезное и эффективное программное средство, позволяющее исследовать взаимосвязи между фазой и частотой колебаний во времени и допускающее постановку достаточно сложных экспериментов [9]. Однако, если учитель захочет изменить форму отображения колебаний (например, его больше привлекает временная диаграмма, чем имитация изображения на осциллографе), то это будет означать, что ему придется перепрограммировать всю модель, что не под силу большинству преподавателей.

МНОГОФУНКЦИОНАЛЬНЫЙ PHASOR

Многофункциональный Phasor отличается от первоначально разработанного варианта тем, что пользователь может работать с ним, не раскрывая сразу всех его возможностей. Упрощенные интерфейсы предназначены для учащихся с более низким уровнем знаний. Если это необходимо, можно восстановить все предусмотренные возможности. Учитель может легко изменить форму представления результатов моделирования для показа нужных свойств так, как ему больше нравится, а это, согласитесь, лучше, чем подстраиваться под формы, предложенные разработчиком приложения. Достоинства этого подхода лучше проиллюстрировать, рассмотрев, как это может быть применено на практике в типич-

ных ситуациях. Вот несколько примеров использования Многофункциональной Среды, когда хорошо видны преимущества этого подхода по сравнению с предыдущим.

Лекционный показ

Знакомые с работой программы Phasor могут представить себе ситуацию, когда учителю нужны все возможности этой программы. Конечно же, сложность интерфейса может ухудшать восприятие изучаемого материала. Чтение лекции и одновременная настройка параметров сложной модели – занятие не из легких, и здесь помощь Многофункциональной среды при моделировании заключается в том, что моделирование становится столь же простым, как пользование проектором. Пример на рисунке 2 показывает третий слайд из подготовленной заранее серии (это лишь иллюстрация, не связанная с конкретной лекцией) и демонстрирует, как можно объединить несколько слайдов с тем, чтобы в соответствующий момент лекции на экране была бы представлена только нужная информация. Теперь, чтобы запустить программу и задать требуемые значения параметров, достаточно нажать кнопку, и можно сосредоточиться собственно на лекции. Кнопки, расположенные в левом нижнем углу окна позволяют легко переходить к нужному слайду, а расположенные справа – управлять работой модели: включать ее, выключать, приостанавливать.

Если теперь возникнет необходимость для ка-

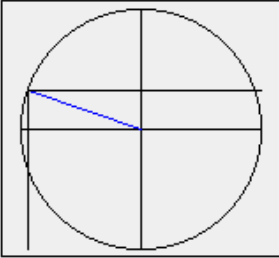
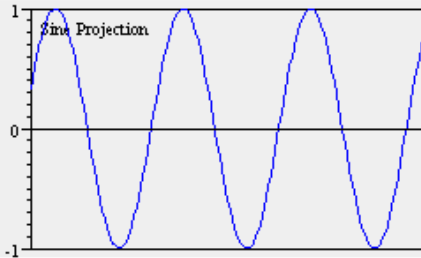
ких-то конкретных целей сменить форму представления информации, характерную для осциллографа, на временную диаграмму, то существует простой путь – вернуться в среду разработки и выбрать нужную опцию.

Дальнейшее усовершенствование лекционных слайдов

Лекционные интерфейсы можно сохранить в форме апплетов, которые могут содержать также пояснительный текст.

ВВЕДЕНИЕ В СИНУСОИДАЛЬНЫЕ КОЛЕБАНИЯ

Представьте себе отрезок прямой, вращающийся против часовой стрелки так, что его конец описывает круг, как показано ниже. Это один из возможных путей продемонстрировать, что представляет из себя синусоида. Проекция вращающегося отрезка на ось абсцисс развернутая во времени является синусоидой.


Постоянный угол f добавленный к переменной q при вычислении значения синусоиды, называется фазой.

Попробуй изменить начальную фазу и посмотри, что получится.

Initial Phase 233

Синусоида порождается вращением отрезка (phasor). Число циклов, которые повторяются в течении секунды, зависит от скорости вращения отрезка. Скорость вращения измеряется как отношение величины изменения угла за единицу времени. Это скорость называется угловой - ω радиан/секунда.

Отрезок начнет вращаться, если вы нажмете на изображение стрелки.



Вы также можете увидеть эффект влияния частоты.

Initial Frequency 1

Рисунок 3. Вариант окна Phasor, содержащий апплеты с интерфейсом, ориентированным на работу с Web-страницами

Таким образом, студенты получают и текст лекции и возможность экспериментировать, имея под рукой программу просмотра Web-страниц.

Сетевые сеансы и помощь студентам

Phasor является примером программы, которая хорошо работает в тех ситуациях, для которых он собственно и проектировался, а именно, для сопровождения лекций, разработки учебных пособий, проведения практических работ. Однако, для того чтобы извлечь максимальную пользу от усилий, затраченных на разработку моделей, мы должны предоставить средства для использования программы в ситуациях, когда непосредственная помощь учителя или разработчика может оказаться недоступной.

Так как в Многофункциональной Среде можно легко управлять ходом эксперимента и сопровождать увиденное дополнительным материалом, доступным в сети без перепрограммирования самой модели, то можно усилить возможности данного подхода, просто разрешая пользователям самим управлять ходом эксперимента. Как мы видели, даже если одна и та же модель используется многими, совсем не обязательно всем использовать один и тот же интерфейс. Интерфейс может помочь удаленным пользователям так, чтобы в какой-то мере заменить объяснения и показ материала лектором. Разработчик курса может построить изложение, максимально приближенное к изложению материала в учебнике, но использовать интерфейс так, чтобы «ожил» сопровождающие текст диаграммы.

Интерфейс многофункциональной оболочки

может быть сохранен таким образом, что каждый видимый объект на экране будет представлен своим собственным апплетом. Существует простой способ скомпоновать эти апплеты с HTML так, чтобы получилась интерактивная Web страничка. Рисунок 3, являющийся вводным материалом для объяснения синусоидальных колебаний, демонстрирует этот подход. Учащиеся могут наблюдать в динамике движение объекта, экспериментировать, меняя частоту и фазу. В этом случае задание не формулируется и тем самым поощряется эксперимент. В то же время есть возможность сохранить в виде апплета весь интерфейс, куда одновременно включены все образы и сопровождающий их текст.

Учащиеся, работающие в сети, должны иметь возможность общаться как с преподавателем, так и с другими учениками. Они должны иметь возможность получать ответы на свои вопросы и обсуждать результаты проделанной работы. Многофункциональная среда поддерживает эти возможности. Пользователь может «перехватывать» и комментировать уви-

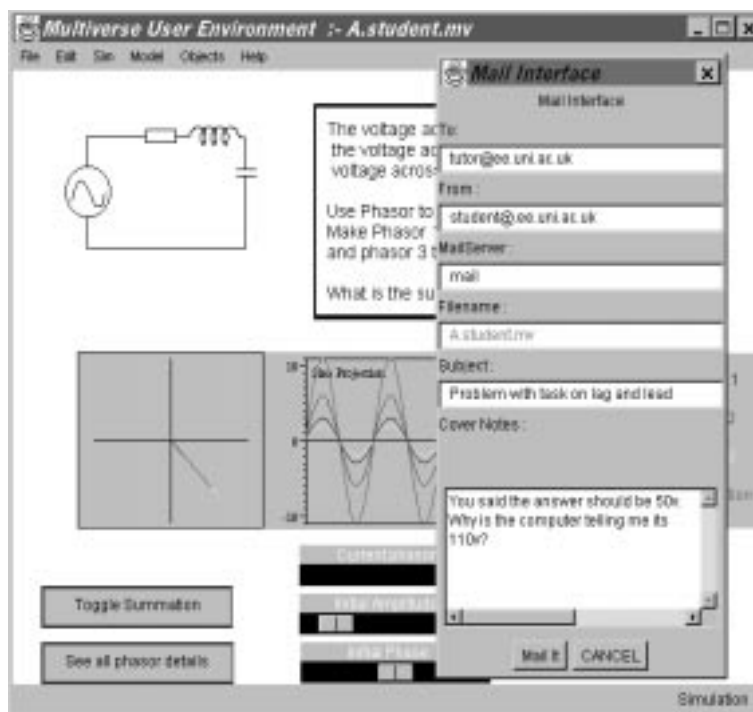


Рисунок 4. Пример диалога со студентом

денное на экране и затем по электронной почте либо посылать преподавателю, либо помещать на всем доступную «доску объявлений». На рисунке 4 показана одна из таких ситуаций. Учитель получит отправленное электронное письмо с файлом «A.student.mv». Это позволит ему полностью восстановить именно ту стадию моделирования, что вызвала затруднения у уче-

ника. Часто задаваемые вопросы и типичные ответы, сопровождаемые поясняющими экспериментами, можно накапливать, экономя тем самым время преподавателя.

Многофункциональная среда разработана компанией MultiVerse Solutions Ltd, работающей при университете Хериот-Ватт в Эдинбурге.

Литература.

1. Tawney, D.A. (Ed), CAL and learning. In Learning through Computers: An introduction to Computer Assisted Learning in Engineering, Mathematics and Sciences at Tertiary Level, Macmillan, London, 1979, pp 109-129.
2. Laurillard, D. Rethinking University Teaching: A framework for the Effective Use of Educational Technology. Routledge London, 1993.
3. Hartel H, Learning with Simulations, Proceedings of International CoLoS Conference, Maribor, Slovenia, 1998, pp 147-155.
4. A.M.Close, Testing Understanding – Does the Computer Make Any Difference?, International CoLoS Conference, Maribor, Slovenia, 1998, pp 107-115.
5. Miller, R., L. and Olds, B. M., Encouraging Critical Thinking in an Interactive Chemical Engineering Laboratory Environment. Proceedings of 1994 Frontiers in education Conference, pp 506-510.
6. Thomas, R. and Neilson, I. Harnessing simulations in the service of education: The Interact Simulation Environment. Computers Educ, 25, 1995, pp 21-29.
7. Neilson, I., Thomas, R., Smeaton, C., Slater, A., and Chand, G. Education 2000 – Implications Of W3 Technology. Computers & Education, Vol. 26, No.1-3, 1996, pp 113-122.
8. Neilson, I., and Thomas, R. Designing Educational Software As A Reusable Resource. Journal Of Computer Assisted Learning, Vol.12, No.2, 1996, pp 114-126.
9. A.M.Close, Y.Rzhanov, H.M.Conner, Bringing Understanding to Electrical Signals Through Computer Representation and Display, International CoLoS conference, Maribor, Slovenia, 1998, pp 116-124.

**Клоус Алекс,
Университет Хериот-Ватт,
кафедра Информатики
и Электроники, Эдинбург,
Шотландия.**

**Томас Рут,
директор образовательных
программ компании
«Образные решения»
при Университете Хериот-Ватт.**

НАШИ АВТОРЫ