

# КОМП'ЮТЕРНІ ЗАСОБИ, МЕРЕЖІ ТА СИСТЕМИ

*Рассмотрены основы денотационной модели автоматизированного синтеза тренировочных задач в универсальных системах для привития обучаемым интеллектуальных навыков. Изложенные принципы модели могут также применяться при разработке универсальных систем индивидуального обучения детей в домашних условиях и для самообразования специалистов.*

© И.И. Верещагин, 2004

УДК 004.923

И.И. ВЕРЕЩАГИН

## АВТОМАТИЗИРОВАННЫЙ СИНТЕЗ ТРЕНИРОВОЧНЫХ ЗАДАЧ В ГИБКИХ ТРЕНАЖЁРАХ

Многочисленные современные образовательные среды, в том числе и в глобальной сети Internet, ориентированы лишь на приобретение обучаемыми, хотя и активное, но, тем не менее, знаний [1, 2]. Автоматизация развития у обучаемых собственно навыков происходит в слишком узких границах – понятия “тренажёр” и “навык” прочно связаны в обыденном сознании с имитациями технических средств для развития у человека моторных и близких к ним способностей уверенного обращения со сложными машинами. Однако исследования, проведённые в рамках тематики в Международном научно-учебном центре информационных технологий и систем НАН Украины, позволяют сделать вывод о перспективности автоматизации привития специалисту так называемых интеллектуальных навыков, в частности навыков оперативного, т. е. текущего управления на высших его уровнях [3, 4]. Автоматизированные системы подобного назначения мы назвали гибкими тренажёрами. Поскольку в настоящее время не существует систематического подхода к созданию тренажёров для выработки интеллектуальных навыков, научно-техническая задача разработки прикладной модели гибких тренажёров, инвариантной к реализациям, является актуальной. В данной работе излагаются основные положения той части модели гибких тренажёров, которая относится к автоматизированному синтезу самих тренировочных задач для последующего их решения на автономном персональном компьютере. Другая часть модели описывает эффективное решение созданных задач в

произвольных многопроцессных операционных системах. Хотя рассматриваемая модель построена с помощью денотационной семантики венского метода разработки программ [5, 6], в тексте указанная семантика почти не применяется из-за опасения потери прозрачности изложения в условиях ограниченного объёма работы. При описании основ модели, по мере возможности, будем избегать сослагательного наклонения.

Центральные понятия модели – тренировочная задача, план тренировки и цикл тренажа. Поэтому сначала дадим развёрнутые определения этих понятий и покажем их взаимную связь.

Разработчиками задачи, решение которой предполагает развитие интеллектуальных навыков, служат специалисты данного дела, в частности научно-педагогические кадры. Они на основе учебной темы, нормативных документов и собственной квалификации создают тренировочную задачу, или тренаж. Задача готовится разработчиками в гипотетической интегрированной среде на базе персонального компьютера. Такая среда складывается из специализированных инструментальных средств, предназначенных для визуальной разработки необходимых компонентов тренировочной задачи. Центральным компонентом тренажа служит формализованный план тренировки, многократное исполнение которого на персональном компьютере преследует цель выработки у тренируемого, или стажёра нужных навыков. Каждое решение тренировочной задачи по одному и тому же плану тренировки назовём циклом тренажа.

Тренировочная задача – система, состоящая из трёх связанных элементов-систем: мир задачи, мир явлений и виртуальный мир. Мир задачи представляет собой модель фрагмента или стороны реального или воображаемого мира. Такой мир существует, в основном, в сознании разработчиков задачи. Мир явлений это обобщённое описание процессов требуемого типа из мира задачи в явлениях и через информацию о явлениях от различных источников мира задачи. Именно в таком виде типовой процесс является тренируемому, который вовлечён в процесс в заданном образе и которому нужно предугадать развитие процесса, чтобы выполнить поставленное перед ним задание. Виртуальный мир генерируется гибким тренажёром по формализованному тексту задачи. В нём стажёру в циклах тренажа предъявляются процессы одного и того же типа в сценах, разыгрываемых на экране компьютера. В каждом цикле тренируемый решает задание одного типа, но, как правило, в более сложных условиях, например, в более сжатые сроки.

Основная цель, которая преследовалась при построении модели автоматизированного синтеза задач, было обеспечение полиморфизма текстов директив, скрытых от разработчиков в визуальной интегрированной среде. Такой полиморфизм осуществляется на трёх уровнях: уровень деталей картин, уровень сцен и уровень плана тренировки. Рассмотрим перечисленные уровни на конкретном примере.

Следующее предложение демонстрирует принципиальную структуру данных, сформированную в памяти компьютера в ходе визуальной разработки персонажа (детали) картины с именем “Главный бухгалтер”:

$$\begin{aligned}
& \text{Главный бухгалтер : } \mathbf{Должностное\ лицо} = & \mathbf{(НазвЦикла} \\
& \mathbf{— m} \rightarrow (s\text{-имя : ГлобИмя} \times s\text{-облик : Облик} \times & \\
& s\text{-знание : } \mathbf{(НазвУрЗнания} \mathbf{— m} \rightarrow \text{Текст})) \times & \\
& s\text{-поведение : } \mathbf{(НазвДействия} \leftarrow m \rightarrow s\text{-метод : (Аргумент*} \rightarrow \mathbf{[OUT]})} \times & \\
& s\text{-событие : } \mathbf{(НазвСобытия} \leftarrow m \rightarrow \text{НазвДействия}) & \mathbf{)} \times \quad (1)
\end{aligned}$$

Детали картин, как правило, владеют атрибутами облика (внешнего вида на экране) и знания в форме поименованных единиц текста. В примере (1), персонаж типа “Должностное лицо” с именем экземпляра “Главный бухгалтер” имеет в каждом поименованном цикле тренажа (НазвЦикла) своё имя по игре (s-имя) и внешний вид на экране компьютера (s-облик). В каждом цикле тренажа можно менять знания, которыми владеет персонаж (s-знание). Более того, в течении одного цикла уровнем знания можно варьировать посредством указания меток текста (НазвУрЗнания). Используя механизм событий (s-событие), стажёр всегда может получить доступ к знаниям, которыми владеет деталь картины в известный момент, выполнив, например, двойной щелчок мышью в границах её облика. Подчеркнём, что деталями картин, в разработанной модели, может служить и реквизит, скажем, телефон, кнопка вызова секретаря и т. п., а также книги, экономическая документация, схемы, рисунки, часы и другие вещи подобного рода. Инструменты стажёра для связи с виртуальным учителем и перемещения по сценам, которые рассмотрим ниже, тоже являются деталями картин.

Очередным уровнем в ходе разработки тренировочной задачи служит формирование сцен из ранее определённых деталей картин. Предположим, что необходимо создать простую динамическую сцену: “Должностное лицо входит в кабинет начальника с докладом”. Её принципиальная структура в памяти компьютера может быть такой:

$$\begin{aligned}
& \text{Входит в кабинет : } \mathbf{Динамическая\ сцена} = \\
& s\text{-переменные : } \langle \mathbf{(ИмяДетали} \times \text{“Должностное лицо”}) \rangle \times \\
& s\text{-картина : НазвТипаКартины} \times \\
& s\text{-действия : } \mathbf{ИмяДетали} \leftarrow m \rightarrow \langle \mathbf{(“Войти”} \times \text{ЗначАргумента*}), \\
& \mathbf{(“Представиться”} \times \text{ЗначАргумента*}), \mathbf{(“Рассказать”} \times \\
& \mathbf{ЗначАргумента*}) \rangle \quad (2)
\end{aligned}$$

Сцены всегда разыгрываются на фоне так называемых картин и программируются путём визуального размещения в картинах эталонов деталей требуемого типа – в целом процедура аналогична визуальному программированию сценариев анимационных фильмов [3]. В примере (2) “Динамическая сцена” – это название типа сцены с именем “Входит в кабинет” – содержит одну связанную переменную ИмяДетали с типом “Должностное лицо”. В плане тренировки, фрагмент которой рассматривается ниже, такая сцена может быть вызвана с любым экземпляром детали, имеющим заданный тип “Должностное лицо”. Подобную сцену может возбудить и стажёр, когда его действие предусмотрено разработа-

ками тренировочной задачи. В примере предполагается, что названия действий персонажа: “Войти, представиться, рассказать” содержатся в таблице поведения (s-поведение) должностного лица из примера (1). Значения аргументов для этих действий (ЗначАргумента\*) задаются при визуальном проектировании сцен и могут храниться в какой-либо несложной базе данных, например, в MS SQL Server [7]. Это справедливо для текстов, обликов, деталей картин, названий циклов тренажа и вообще для любых данных тренировочной задачи.

Разработка плана тренировки служит заключительным этапом в создании тренировочной задачи. Он базируется на ранее определённом множестве сцен. План тренировки разбивается на именованные периоды времени, называемые актами тренировки. Каждый акт, в свою очередь, может быть разделён на именованные секции, которые носят название действий. Особо подчеркнём, что все циклы тренажа выполняются по одному и тому же плану тренировки, но при разных значениях атрибутов для актов и действий. Разумеется, что облики деталей картин и самих картин, а также знания их персонажей и реквизита, как следует из вышеописанного, тоже подлежат модификации и изменению. Продолжая примеры (1) и (2), рассмотрим следующий акт плана тренировки:

**[Начало финансового кризиса предприятия] Акт :**  
**Сцена ( Кабинет директора )**  
**ПаузаОтносительная ( 5% )**  
**Сцена ( Входит в кабинет : Главный бухгалтер )**  
**[Отражение кризиса в балансе] Действие :**  
**ПаузаОтносительная ( 10% )**  
**СменитьЗнания ( Баланс на начало кризиса )**  
**[Отражение кризиса в остальных знаниях] Действие :**  
**СменитьЗнания ( Знания на начало кризиса ) (3)**

С каждым актом связаны два атрибута: продолжительность акта тренировки в минутах и переключатель – включить в план тренировки и исключить из плана. С действиями ассоциирован только один атрибут – переключатель. Теперь коротко рассмотрим гипотетические директивы. Первая директива Сцена() представляет на экране так называемую статическую сцену, картина которой просто отображает растр с именем “Кабинет директора”. Очевидно, что под таким именем в каждом цикле тренажа может скрываться иное изображение. Следующая одноимённая директива возбуждает динамическую сцену, рассмотренную в примере (2). Вместо связанной переменной ИмяДетали здесь подставляется экземпляр “Должностного лица” под именем “Главный бухгалтер” из примера (1). Сцена разыгрывается на фоне изображения кабинета директора предприятия. Директивы СменитьЗнания() задают текущий уровень знаний деталей картин, тексты которых помечены метками “Баланс на начало кризиса” и “Знания на начало кризиса” соответственно – в этой связи в структуре данных примера (1) обращаем внимание на ключ НазвУрЗнания.

Кратко рассмотрим остальные компоненты модели автоматизированного синтеза тренировочных задач. Разработчики тренажа при проектировании задачи в визуальной интегрированной среде оперируют, в основном, категориями сущностей и атрибутов сущностей. Поэтому создаваемый тренаж рассматривается ими как единство трёх сущностей: изображающего пространства, в частности экрана компьютера, тренируемого (стажёра) и виртуального учителя. Изображающее пространство складывается из сфер изображения, причём для плана тренировки, для стажёра и виртуального учителя целесообразно назначить свою, отдельную сферу изображения. В каждой сфере изображения представлена сцена, состоящая из картины с размещёнными в ней деталями картины, например, персонажами и реквизитом – под персонажем понимается деталь картины, наделённая качествами человека. Тогда в сфере изображения, скажем, виртуального учителя может быть представлена картина с элементами взаимодействия со справочной системой, и эти элементы будут выступать в качестве деталей картины в такой “служебной” сцене. В сфере изображения стажёра может размещаться картина с инструментами (детальями картины), позволяющими тренируемому перемещаться по статическим сценам и возбуждать сцены динамические. Подобным образом все сущности типизируются и образуют деревья типов. В качестве иллюстрации рассмотрим тип “Должностное лицо”, экземпляр которого уже фигурировал в примере (1):

$$\begin{aligned}
 &\text{Должностное лицо} = \\
 &s\text{-родовойТип} : \text{Человек Взрослый} \times \\
 &s\text{-имя} : \text{BOOL} \times s\text{-облик} : \text{BOOL} \times s\text{-знание} : \text{BOOL} \times \\
 &s\text{-поведение} : (\text{НазвДействия} \leftarrow m \rightarrow \text{Аргумент}^*) \times \\
 &s\text{-событие} : (\text{НазвСобытия} \leftarrow m \rightarrow \text{НазвДействия}) \quad (4)
 \end{aligned}$$

Каждая сущность снабжена уникальным названием. То же справедливо и для имён атрибутов. Способность сущности иметь некоторое качество называется атрибутом. К основным видам атрибутов относятся: атрибут существования (возможность образовывать экземпляр сущности или, то же самое, что экземпляр некоторого типа), облик сущности, звуковые (шумовые) атрибуты, информационные атрибуты, в частности данные и знания, атрибуты поведения (действий) и общения, например, через механизм событий. Сам атрибут служит элементарной или атомарной сущностью, следовательно, не включает других атрибутов, поэтому является чистым качеством.

В заключение очертим круг возможного применения изложенной денотационной модели. Кроме её основного назначения – описания инвариантных принципов автоматизированного синтеза тренировочных задач в универсальных системах для привития обучаемым интеллектуальных навыков, – разработанные подходы могут найти широкое применение при создании универсальных систем, предназначенных для индивидуального обучения детей в домашних условиях и для самообразования специалистов.

**Основные выводы:**

1. Автоматизированные системы на базе универсальных персональных компьютеров, предназначенные для привития обучаемому интеллектуальных навыков, целесообразно называть гибкими тренажёрами и противопоставлять такие системы тренажёрам, призванным вырабатывать навыки обращения со сложными машинами.

2. Научно-техническая задача разработки модели гибких тренажёров, инвариантной к реализациям, является актуальной, поскольку в настоящее время не существует систематического подхода к созданию универсальных автоматизированных систем для развития интеллектуальных навыков.

3. Инвариантная к реализациям модель гибких тренажёров состоит, в свою очередь, из модели автоматизированного синтеза самих тренировочных задач и модели эффективного решения разработанных задач на персональных компьютерах с произвольными многопроцессными операционными системами.

4. Рассмотрены основы инвариантной к реализациям денотационной модели автоматизированного синтеза тренировочных задач в гибких тренажёрах.

5. Изложенные принципы модели могут найти также применение при разработке универсальных систем индивидуального обучения детей в домашних условиях и для самообразования специалистов.

1. *Chunlin Li, Zhengling Lu, Layuan Li.* Design and Implementation of A Hybrid Agent Platform // Программирование. – 2003. – № 1. – С. 45–61.
2. *Шереметов Л., Усков В.* Виртуальные образовательные среды. Приложение // Информационные технологии. – 2002. – № 5. – 24 с.
3. *Верещагин И.И.* Организация анимационных фильмов в гибких тренажёрах // Компьютерні засоби, мережі та системи. – 2003. – № 2. – С. 158–163.
4. *Верещагин І.І., Динисовець В.Д.* Принципи архітектури гнучких тренажерів для вищих ланок управління // УкрІНТЕІ. – 2002. – № 1/2. – С. 10–13.
5. *Петренко А.К.* Венский метод разработки программ. Неформальное введение // Программирование. – 1991. – № 6. – С. 3–23.
6. *Fitzgerald John, Larsen Peter G.* Modelling Systems: Practical Tools and Techniques for Software Development. Cambridge University Press, 1998. – 486 p.
7. *Тихомиров Ю.В.* MS SQL Server 2000: Разработка приложений. – СПб.: БХВ – Петербург, 2000. – 368 с.

Получено 06.02.2004