

Исследование многокомпонентной модели обучения на ЭВМ

Р.В. Майер

Глазовский государственный педагогический институт, г. Глазов

Сформулирована основная задача дидактики, предложена многокомпонентная модель знаний, рассмотрены компьютерные программы, имитирующие процесс обучения, проанализированы получающиеся результаты.

Допустим, имеется n учеников, каждый из которых характеризуется набором параметров $\alpha_i, \beta_i, \gamma_i \dots (i = 1, 2, \dots, n)$ и m учителей, владеющих методами M_1, M_2, M_3 и т.д. **Основная задача дидактики** состоит в том, чтобы так организовать учебный процесс, то есть выбрать методы и распределить изучаемый материал в течение заданного промежутка времени, чтобы в конце обучения учащиеся справились с системой тестов $T = \{T_1, T_2, \dots\}$. Сформулируем **закон дидактики**: скорость увеличения знаний Z пропорциональна прилагаемым усилиям $F(t)$, эффективности методики обучения η , коэффициентам усвоения α и понимания Π : $dZ / dt = \alpha \Pi \eta F(t)$. Будем считать, что прилагаемые усилия F пропорциональны разности между уровнем требований U учителя и знаниями Z учащихся: $F = a(U - Z)$.

Как известно, процесс усвоения и запоминания сообщаемой информации состоит в установлении логических и ассоциативных связей между новыми и имеющимися знаниями. В результате приобретенные знания становятся более прочными и забываются значительно медленнее. Предлагаемая многокомпонентная модель обучения выражается системой уравнений:

$$dZ_1 / dt = k\alpha_1(U - Z) - k\alpha_2Z_1 - \gamma_1Z_1, \quad dZ_2 / dt = k\alpha_2Z_1 - k\alpha_3Z_2 - \gamma_2Z_2, \\ dZ_3 / dt = k\alpha_3Z_2 - k\alpha_4Z_3 - \gamma_3Z_3, \quad dZ_4 / dt = k\alpha_4Z_3 - \gamma_4Z_4,$$

где U — уровень требований учителя, равный сообщаемым знаниям Z_0 , Z — суммарные знания, Z_1 — самые “непрочные” знания первого типа с высоким коэффициентом забывания γ_1 , а Z_4 — самые “прочные” знания четвертого типа с низким γ_4 ($\gamma_4 < \gamma_3 < \gamma_2 < \gamma_1$). Коэффициенты усвоения α_i характеризуют быстроту перехода знаний $(i-1)$ -ого типа в знания i -ого типа. Пока происходит обучение, $k = 1$, а когда оно прекращается, $k = 0$. Коэффициент забывания $\gamma = 1/\tau$, где τ — время, в течение которого количество знаний i -ого типа уменьшается в $e = 2,72\dots$ раз. Результат обучения характеризуется суммарным уровнем приобретенных знаний и коэффициентом “прочности”: $Z = Z_1 + Z_2 + Z_3 + Z_4$, $Pr = (Z_2 / 4 + Z_3 / 2 + Z_4) / Z$. При изучении одной темы растет уровень знаний Z , одновременно происходит увеличение доли “прочных” знаний Z_4 и повышается прочность Pr .

```

Uses crt, graph; { PR-1 }
Const a1=0.01; a2=0.003; g1=0.005; g2=0.0001; dt=0.02; Mt=0.2; Mz=2.9;
Var t,U,Z1,Z2,Z3,Z4,Z,Pr,S: real; DV,MV,k: integer; { Free Pascal }
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi');
Repeat t:=t+dt; U:=0; k:=0; If (t<300) then begin U:=0.15*t+20; k:=1; end;
If (t>600)and(t<900) then begin U:=0.15*(t-600)+50; k:=1; end;
If (t>1200)and(t<1500) then begin U:=0.15*(t-1200)+90; k:=1; end;
Z:=Z1+Z2+Z3+Z4; Pr:=Z2/(Z+0.001);
Z1:=Z1+k*a1*(U-Z)*dt-g1*Z1*dt-k*a2*Z1*dt;
Z2:=Z2+k*a2*Z1*dt-g2*Z2*dt; circle(10+round(Mt*t),450-round(Mz*Z),1);
circle(10+round(Mt*t),450-round(Mz*(Z2)),1); circle(10+round(Mt*t),450-round(
Mz*(U)),1); circle(10+round(Mt*t),450-round(150*Pr),1);
until KeyPressed; CloseGraph;
END.

```

Для исследования двухкомпонентной модели обучения используется программа PR-1, результаты имитационного моделирования — на рис. 1.1. Учитель проводит три урока, в течение которых уровень требований растет пропорционально времени: $U = a(t - t_0) + b$. Видно, что во время перерывов и после обучения уровень “непрочных” знаний Z_1 быстро уменьшается, а “прочные” знания Z_2 забываются медленнее. При использовании четырехкомпонентной модели знаний получаются аналогичные результаты (рис. 1.2). Уровень требований в течение занятия остается постоянным.

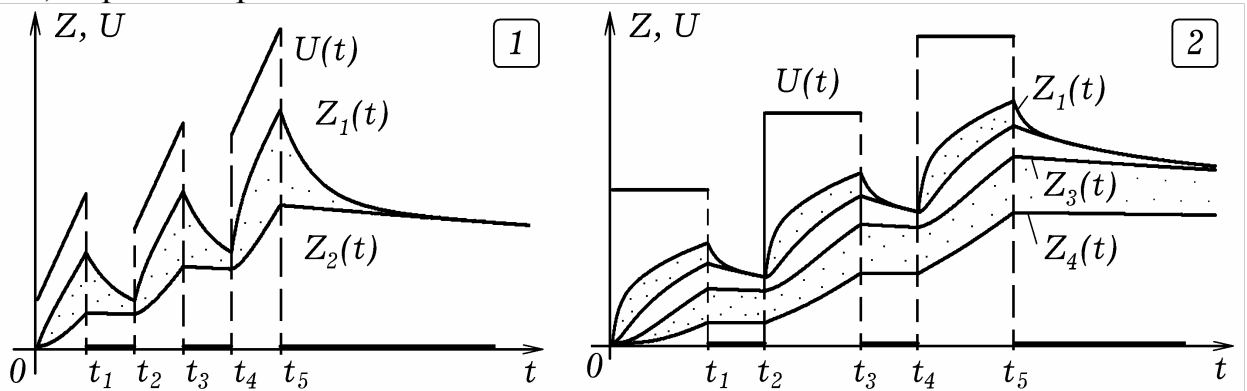


Рис. 1. Двух- и четырехкомпонентная модель обучения.

Обучение будет наиболее эффективным, когда уровень требований учителя превышает знания учащегося на максимально возможную величину, при которой у учащегося не пропадает мотивация к учебной деятельности. Такой режим обучения будем называть *согласованным*. Для нахождения эффективного пути обучения, соответствующего минимальным затратам энергии учителя и учащегося, в качестве целевой функции рассматриваемой оптимизационной задачи возьмем функционал:

$$S = \int_{t_1}^{t_2} (U(t) - Z(t)) dt \approx \sum_{j=1}^n (U_j - Z_j) \Delta t, \quad n = (t_2 - t_1) / \Delta t.$$

Нагрузка на учащегося должна быть более или менее равномерно распределена по всем занятиям так, чтобы не было переутомления. Поэтому для каждого i -ого занятия следует вычислять затраты энергии S_i и следить, чтобы они не превысили пороговое значение S_{\max} .

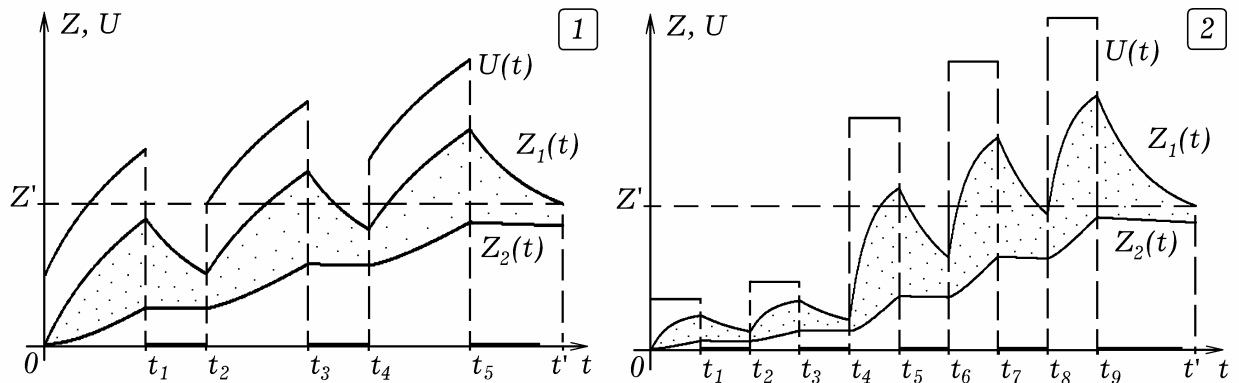


Рис. 2. Нахождение оптимальной организации обучения.

Пусть в режиме согласованного обучения проводятся три занятия, начинающиеся в фиксированные моменты времени $t_0 = 0$, t_2 и t_4 . Необходимо определить длительность занятий, при которой уровень знаний в момент t' будет равен $Z(t') = Z'$. Для решения этой оптимизационной задачи использовалась программа, содержащая цикл, в котором случайным образом изменяется длительность урока T_u , затем пересчитывается Z и выясняется, приблизился уровень знаний к требуемому значению Z' или нет. Если да, — изменения T_u принимаются, если нет, — отвергаются, и все повторяется снова. Результаты решения этой задачи представлены на рис. 2.1.

Рассмотрим другую ситуацию, в которой начала t_0 , t_2 , t_4 , t_6 , t_8 и длительности T_u пяти занятий фиксированы, а уровни требований U_i ($i = 1, 2, \dots, 5$) изменяются. Необходимо подобрать такие U_i , чтобы уровень знаний учащегося достиг заданного значения Z' . Для этого используется программа PR-2, результаты моделирования представлены на рис. 2.2.

```

Uses crt, graph; Const a1=0.01; a2=0.002; N=5; Ur=90; g1=0.005; g2=0.0001;
dt=0.5; Mt=0.3; Mz=2.1; Var min,UU,t,Z1,Z2,Z,Zn,Zn2,Pr,Sum,S1,g:real;
i,k,kk,DV,MV:integer; S,U,U1: array[0..N] of real; { PR-2 }
Procedure Raschet;
begin t:=0; Sum:=0; Z1:=0; Z2:=0; Zn:=0; For i:=1 to N do S[i]:=0;
Repeat t:=t+dt; k:=0; i:=round(t/10);
If t<200 then begin UU:=U[1]; k:=1; S[1]:=S[1]+k*abs(UU-Z)*dt; end;
If (t>400)and(t<600)then begin UU:=U[2];k:=1;S[2]:=S[2]+k*abs(UU-Z)*dt;end;
If (t>800)and(t<1000)then begin UU:=U[3];k:=1;S[3]:=S[3]+k*abs(UU-Z)*dt;end;
If(t>1200)and(t<1400)thenbegin UU:=U[4];k:=1;S[4]:=S[4]+k*abs(UU-Z)*dt;end;
If(t>1600)and(t<1800)thenbegin UU:=U[5];k:=1;S[5]:=S[5]+k*abs(UU-Z)*dt;end;
If UU>=Z then Z1:=Z1+k*a1*(UU-Z)*dt-g1*Z1*dt-k*a2*Z1*dt;

```

```

If UU<Z then Z1:=Z1-g1*Z1*dt-k*a2*Z1*dt;
Z2:=Z2+k*a2*Z1*dt-g2*Z2*dt; Z:=Z1+Z2; Sum:=Sum+k*abs(UU-Z)*dt;
until (t>2200)or(KeyPressed); Zn:=Z; Zn2:=Z2; end;
Procedure Draw;
begin t:=0; Z1:=0; Z2:=0; Z:=0; cleardevice;
Repeat t:=t+dt; k:=0; If t<200 then begin UU:=U[1]; k:=1; end;
If (t>400)and(t<600) then begin UU:=U[2]; k:=1; end;
If (t>800)and(t<1000) then begin UU:=U[3]; k:=1; end;
If (t>1200)and(t<1400) then begin UU:=U[4]; k:=1; end;
If (t>1600)and(t<1800) then begin UU:=U[5]; k:=1; end;
If UU>=Z then Z1:=Z1+k*a1*(UU-Z)*dt-g1*Z1*dt-k*a2*Z1*dt;
If UU<Z then Z1:=Z1-g1*Z1*dt-k*a2*Z1*dt;
Z2:=Z2+k*a2*Z1*dt-g2*Z2*dt; Z:=Z1+Z2;
circle(10+round(Mt*t),450-round(Mz*Z),1);circle(10+round(Mt*t),450-round(Mz
*Ur),1); circle(10+round(Mt*t),450-round(Mz*(Z2)),1); circle(10+round(Mt*t),
450-round(Mz*(UU)),1); until (t>2200)or(KeyPressed); end;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi'); Randomize; min:=1E+7;
For i:=1 to N do U[i]:=20+40*i;
Repeat For i:=1 to N do U1[i]:=U[i]; Raschet; S1:=Sum;
If Sum<min then min:=Sum;
For i:=1 to N do begin U[i]:=U[i]+random(800)/100-4;
If U[i]<0 then U[i]:=0; end; Raschet;
If (Sum>min)or(Zn<Ur)or(Zn2<0.6*Ur) then For i:=1 to N do U[i]:=U1[i]; g:=0;
For i:=1 to N do If S[i]>75*200 then g:=1; If (g=0)and(Sum<=min)and(Zn>Ur)
and(Zn2>0.6*Ur) then begin Draw; circle(10,round(min/200),3); end;
until Keypressed; CloseGraph;
END.

```

Теперь представим, что учитель должен обучить ученика решать 10 задач возрастающей сложности $\theta_i = \theta_i + i\Delta\theta$, которая равна количеству знаний, требующихся для решения i -той задачи. Учитель располагает задачи в порядке возрастания сложности и задает их ученику через равные промежутки времени Δt . Если общее количество знаний Z больше или равно θ_i , то ученик решает задачу. Суммарные знания Z увеличиваются, часть “непрочных” знаний становится “прочными”. После этого ученику предлагается $(i+1)$ -ая задача с более высоким уровнем сложности θ_{i+1} . Если у ученика знаний меньше чем θ_{i+1} , то он не может решить $(i+1)$ -ую задачу сразу. Учитель его обучает в течение времени Δt , а затем снова предлагает эту же или аналогичную задачу той же сложности θ_{i+1} . Занятия длительностью $T_u \gg \Delta t$ чередуются переменами продолжительностью $T_{II} \gg \Delta t$.

Для моделирования используется программа PR-3, в ней решение задачи рассматривается как случайный процесс, вероятность которого вычисляется по формуле Раша: $p_i = 1/(1 + \exp(-\lambda(Z(t) - \theta_i)))$. Результаты имита-

ционного моделирования обучения на трех и четырех занятиях представлены на рис. 3. Ступенчатая линия $u(t)$ показывает, как меняется сложность решаемых задач; графики $Z_1(t)$ и $Z_2(t)$ характеризуют динамику роста “непрочных” и “прочных” знаний. Эти кривые похожи на графики на рис. 1.1, соответствующие линейной зависимости уровня требований учителя от времени. Предложенные программы позволяют проанализировать различные ситуации, возникающие в процессе обучения (см. <http://rmajer.narod.ru>).

```

uses crt,graph; { PR-3 }
const a1=0.005; a2=0.002; g=0.0002; dt=0.2; b=0.03; Mt=0.1; N=35;
var DV,MV,i,k,r:integer; Z,Z1,Z2,U,t,p,x,Tu,Tp:real;
Z0: array[1..N]of integer; usl:boolean;
Procedure Test;
begin r:=1; p:=1/(1+exp(-b*(Z-Z0[i]))); x:=random(100)/100;
If (x<p)and(usl) then begin i:=i+1; r:=0; end; end;
BEGIN DV:=Detect; InitGraph(DV,MV,'c:\bp\bgi'); Randomize;
For i:=1 to N do Z0[i]:=10+15*i; i:=1; Tu:=1100; Tp:=700;
Repeat t:=t+dt; inc(k);
usl:=(t<Tu)or((t>Tu+Tp)and(t<2*Tu+Tp))or((t>2*Tu+2*Tp)and(t<3*Tu+2*Tp));
If usl then U:=r*(Z+40) else r:=0; Z:=Z1+Z2; If U>Z then Z1:=Z1+a1*r*
(Z0[i]-Z)*dt-g*Z1*dt-a2*r*Z1*dt else Z1:=Z1-g*Z1*dt;
If Z1>0 then Z2:=Z2+a2*r*Z1*dt-g*Z2*dt/10 else Z2:=Z2-g*Z2*dt/10;
If k mod 200=0 then Test; If i>N then i:=N; If k=1000 then k:=0;
circle(round(Mt*t),450-round(Z0[i]),1); circle(round(Mt*t),450-round(Z1+Z2),2);
circle(round(Mt*t),450-round(Z2),2);
until (KeyPressed); Closegraph;
END.

```

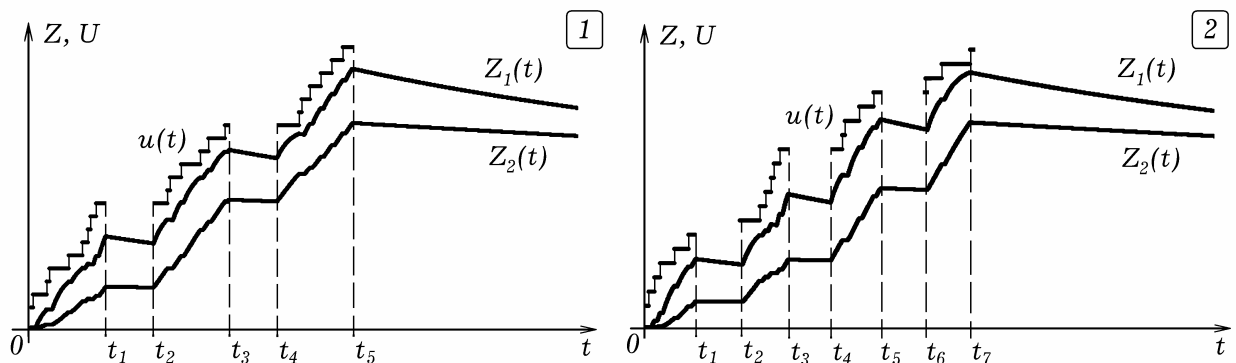


Рис. 3. Модель обучения путем решения задач возрастающей сложности.

Литература

1. Шеннон Р. Имитационное моделирование систем: искусство и наука. — М.: Мир, 1978. — 302 с.