

Об одном алгоритме построения проверочных вопросов в компьютерной обучающей системе C-tutor

Лю Юн (КНР), А. С. Строгалов

Введение

Одним из важных моментов в создании и последующем функционировании интеллектуальной обучающей системы (ITS — Intellectual Tutoring System) является определение уровня знаний обучаемого (ученика). Поэтому внутренняя модель знаний ученика в обучающей системе является компонентом, существенно влияющим на поведение системы в процессе обучения (интерактивного взаимодействия «ученик—система») [1, 3]. Можно даже утверждать, что удачно построенная модель знаний ученика делает поведение и реакции ITS на события, возникающие в процессе обучения, более осмысленными и позволяет приблизиться к решению задач индивидуального обучения. Эта же модель определяет ключевое различие между ITS и системами автоматизированного обучения (CAI — Computer-Aided Instruction [2, 4]). В данной работе мы описываем построение модели знаний ученика в системе C-tutor (ITS для изучения китайского как иностранного) и предлагаем алгоритм процесса организации проверочных вопросов, основанных на модели знаний ученика (образа обучаемого, модели ученика).

1. Модель ученика в системе C-tutor

Одной из распространенных конструкций для процесса моделирования образа ученика является конструкция «проекций (overlay) и

ошибок (bug)» [5]. Ее основой является то, что знания ученика рассматриваются как подмножество знаний эксперта и сравниваются с ними для того, чтобы определить, какие знания отсутствуют. Информация об ошибках базируется на библиотеке потенциальных ошибок, которые могут быть допущены в процессе обучения. Два этих соображения служат основой для построения семейства смешанных моделей, которые объединяют их сильные стороны, чтобы по возможности свести к минимуму жесткие ограничения исходной конструкции. Понятно также, что эффективность использования той или иной модели знаний ученика тесно связана с организацией процесса обучения на основе ITS.

Обобщенная схема C-tutor'a представлена ниже (рис. 1):

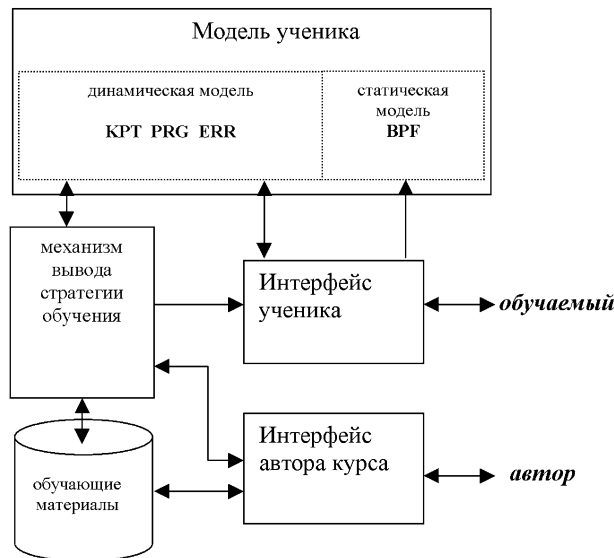


Рис. 1. Обобщенная схема C-tutor.

В системе C-tutor предлагается смешанная модель ученика, так как она более удобна для планирования процесса обучения и контроля уровня знаний, навыков и умений у ученика в области изучения начального курса китайского языка как иностранного.

Структура данных $ST_MODEL = \{BPF, KPT, PRG, ERR\}$ в этой модели состоит из четырех компонент:

Компонента $BPF = \{S_ID, S_NAME, PASSWD, NATION, NATIVE_LAN, REG_DATE, START_LINE\}$ определяет первоначальную информацию об ученике. Элементы этого множества соответственно означают: личный номер студента, имя, под которым он входит в систему (логин), пароль, национальность, родной язык, первую дату регистрации и начальный уровень языковой компетенции (0 — не знаю совсем; 1 — изучал менее одного года; 2 — изучал более одного года). Напомним, что это курс для начинающих и уровень 1 и 2 просто пока фиксируют некоторое время, затраченное на изучение языка, что позволяет сделать гипотезу о первоначальном уровне языковой компетенции, которая может быть позднее уточнена как в процессе тестирования, так и в процессе обучения.

Компонента $KPT = \{UNIT_ID, KPV, DATE\}$ — таблица уровня знаний. В этой таблице каждая запись состоит из трех пунктов: ID курса, набор, который обозначает языковую компетенцию (языковые навыки и умения), и дата, когда данный уровень навыков и умений был освоен. Набор KPV состоит из пяти компонент: $(k_id, v_1, v_2, v_3, v_4)$; Здесь k_id обозначает номер уровня знаний, v_1, v_2, v_3, v_4 означают различные уровни языковой компетенции на данном уровне знаний — узнавание, понимание, применение, анализ. Эти четыре значения могут быть вероятностными величинами, если их сумма нормирована на единицу — в общем случае это нечеткие переменные, которые в соответствии с определенной шкалой (не обязательно числовой) показывают значения уровня достигнутых компетенций ученика. Например, набор $KPV = (u1_k5, 0,1, 0,25, 0,65, 0,0)$ показывает, что студент получил оценку по уровню знаний 5 раздела 1, его/ее уровень навыка следующий: узнавание — с вероятностью 0,1, понимание — с вероятностью 0,25, применение — с вероятностью 0,65, анализ — с вероятностью 0. Приведем пример нахождения значений компонент набора $KVP = (k_id, v_1, v_2, v_3, v_4)$.

Итак, допустим, что при тестировании на уровне $k_id = K_i$ было задано четыре вопроса и результаты ответов показаны в таблице.

Таким образом, мы получили $KPV = (K_i, 0,5, 0,5, 0, 0)$.

Заметим, что в описании алгоритма (см. ниже) нахождение компонент набора KPV осуществляется с учетом истории прохождения учеником теста.

Вопросы на уровне K_i	Требования к уровню знаний				Ответ верен?
	узнавание	понимание	применение	анализ	
вопрос-1	✓	✓			да
вопрос-2			✓		нет
вопрос-3			✓	✓	нет
вопрос-4	✓	✓			да
Доля правильных ответов	$r_1 = 2/2$	$r_2 = 2/2$	$r_3 = 0/2$	$r_4 = 0/1$	
Компоненты KVP: $v_i = r_i / \left(\sum_{j=1}^4 r_j \right)$	0,5	0,5	0	0	

Компонента $PRG = \{CURR_KID, PRE_KSET, PRE_KSTATE, SUCC_KSET, SUCC_KSTATE\}$ есть индикатор прогресса процесса обучения. Здесь $CURR_KID$ — текущий уровень знаний, PRE_KSET и $SUCC_KSET$ — множество предыдущих и последующих значений уровней знаний; величины PRE_KSTATE , $SUCC_KSTATE$ отражают степень освоения данного уровня знаний. Информация из PRG помогает системе выбрать правдоподобные гипотезы при формировании дальнейших стратегий обучения.

Компонента $ERR = \{KID, ERR_COUNT, ERR_CLASS, DATE\}$ содержит информацию об ошибках ученика, возникших в процессе обучения. Она фиксирует ошибки, допущенные учеником на каждом уровне знаний ID , повторяемость ошибки (то есть сколько раз ошибка была допущена), класс ошибки и дату. Класс ошибки определяется уровнем сложности проверочных вопросов, который устанавливается с учетом значений компонент набора с v_1 по v_4 .

2. Алгоритм построения контрольных вопросов

Здесь мы приводим основную идею алгоритма и связанных с ним структур данных. Далее мы представим формальное описание алгоритма, который позволяет интеллектуализировать процедуру построения контрольных вопросов.

Задача алгоритма — осуществить автоматический выбор контрольных (экзаменационных) вопросов, соответствующих уровню знаний студента на заданном этапе обучения. Этот выбор осуществляется на основе установки более высоких приоритетов вопросам, соответствующим низким показателям уровня знаний студента. Для этого необходимо найти интегральный вес вектора уровня знаний, который ляжет в основу процедуры выбора вопросов из базы данных. Проиллюстрируем сказанное выше примером, из которого будет понятен механизм такого моделирования.

Зададим конкретные значения компонент набора $\mathbf{W} = (2^0, 2^1, 2^2, 2^3)$, интерпретируемых как веса требований к уровню знаний (УЗНАВАНИЕ, ПОНИМАНИЕ, ПРИМЕНЕНИЕ, АНАЛИЗ) для каждого этапа обучения. Для каждого контрольного вопроса значение весов требований указывается автором курса. Определим расчетный уровень знаний для этапа обучения как $\mathbf{e} = (v_1, v_2, v_3, v_4) \times \mathbf{W}$. Например, если $KPV = (u1_k5, 0,1, 0,25, 0,65, 0,0)$, то $\mathbf{e} = (0,1, 0,25, 0,65, 0,0) \times (2^0, 2^1, 2^2, 2^3) = 3,2$. Мы интерпретируем полученное значение как величину, отражающую обобщенный уровень знаний ученика на этапе $u1_k5$. Это значение будет использовано в качестве стандарта, основываясь на котором будут выбраны вопросы для этого этапа обучения.

Чтобы повысить приоритет процедуры улучшения невысоких показателей обучения ученика, при формировании списка новых контрольных вопросов, мы должны сделать следующее: для всех этапов обучения, входящих в тест, вычислить значение \mathbf{e} по KPV и отсортировать их в порядке увеличения, а затем с учетом этого порядка выбрать вопросы из базы данных.

Например, если ученик на уровне K_i имеет набор значений $KPV_{K_i} = (0,5, 0,5, 0,0, 0,0)$, а на уровне K_j имеет $KPV_{K_j} = (0,0, 0,5, 0,5, 0,0)$, то $\mathbf{e}_{K_i} = (0,5, 0,5, 0,0, 0,0) \times (2^0, 2^1, 2^2, 2^3) = 1,5$ и $\mathbf{e}_{K_j} = (0,0, 0,5, 0,5, 0,0) \times (2^0, 2^1, 2^2, 2^3) = 3,0$. Мы интерпретируем это так, что уровень K_j освоен лучше уровня K_i и, следовательно, при формировании следующего задания из базы данных вопросов должны быть выбраны в первую очередь задания для уровня K_i .

Ранжируя таким образом вопросы к очередным заданиям, мы можем надеяться на более равномерное повышение уровня знаний ученика в процессе обучения.

Формальное описание алгоритма выглядит следующим образом:

```

procedure create_test_paper (unit_id: integer; KPT:
prof_table_type; n_questions: integer; delta_w: integer)
{создать экзаменационный билет для содержания unit_id в
курсе, сумма вопросов= n,  $\max(|e - w|) \leq \text{delta\_w}$ }
begin
  {Шаг 1: создать список этапов обучения, которые попадут
в контрольную работу, список отсортирован в порядке
возрастания значения e}
  KL  $\leftarrow$  NULL; {инициализировать список этапов обуче-
ния KL}
  for (для всех k_nodei  $\in$  Unitu_id)
  begin
    find KPVi in KPT;
    ei  $\leftarrow$  KPVi  $\times$  W;
    if ei  $\leq$  0,85 then {когда ei > 0,85 мы предпо-
лагаем, что студент это уже усвоил}
      insert k_nodei into KL, отсортированным
      по возрастанию ei;
    end;
  }
  {Шаг 2: сформировать контрольное задание, чередуя вопро-
сы из базы данных, сравнивая вес требований w и общий
уровень знаний студента e, где  $|e - w| \leq \text{delta\_w}$ }
  count  $\leftarrow$  1; {Инициализируем количество вопросов}
  testpaper := NULL;
  p := KL  $\rightarrow$  head;
  while (count < n_questions) and (p  $\neq$  NULL)
  begin
    получить значение e из узла, обозначенного
    p; найти вопрос Qcount в базе данных. При
    условии,  $|e - w| \leq \text{delta\_w}$ ;
    вставить Qcount в контрольную;
    p := p  $\rightarrow$  next;
    count := count + 1;
  end;
end;
end;

```

Пример экрана автора курса (для создания упражнения на выбор правильного ответа) приведен на рис. 2.



Рис. 2. Составление вопроса с весом требований.

Список литературы

- [1] Van Lehn K. Student modeling // In M.C. Polson & J. Jeffrey Richardson (Eds.) Foundations of Intelligent Tutoring Systems. P. 55–78. Hillsdale, NJ: Lawrence Erlbaum Associates, 1988.
- [2] Carbonell J.R. AI in CAI: an artificial intelligence approach to computer-assisted instruction // IEEE Transactions on Man — Machine. 11. 1970. P. 190–202.
- [3] Stevens A.L., Collins A., Goldin S. Misconceptions in students' understanding // In D. Sleeman & J.S. Brown (Eds.) Intelligent Tutoring Systems. P. 13–24. London: Academic Press, 1982.
- [4] Carr B., Goldstein I. Overlays: a theory of modeling for computer-aided instruction. AI Lab Memo 406 (Logo Memo 40). Massachusetts Institute of Technology, Cambridge, MA, 1977.
- [5] Brown J., Burton R. Diagnostic models for procedural bugs in basic mathematical skills // Cognitive Science. 2. 1978. P. 155–191.