

KOGNITÍVNY TÚTOR PRE ZAČIATOČNÍKOV V PROGRAMOVANÍ

MGR. MARTINA KABÁTOVÁ

Katedra základov a vyučovania informatiky, Fakulta matematiky, fyziky a informatiky,
Univerzita Komenského, Mlynská dolina, 842 48 Bratislava,
tel. 0907/963 946, e-mail: kabatova@fmph.uniba.sk

Tvorba kognitívnych tútorov pre vyučovanie stavia na poznatkoch kognitívnej vedy a využíva moderné digitálne technológie. Tútoringový program predstavuje edukačnú pomôcku s potenciálom individualizovať a zefektívniť vyučovanie. Zároveň dokáže získavať cenné dáta o postupe študentov a ich spôsobe práce s výučbovým programom. Pomocou nástrojov CTAT (Cognitive Tutor Authoring Tools) sme vytvorili sériu tútorovaných úloh pre začiatok v programovaní a skúmali sme, ako budú študenti na takýto nástroj reagovať, ako ovplyvní vyučovanie a zvažujeme, ako do budúcnosti tútoring ďalej využívať.

Kľúčové slová: didaktika programovania, digitálne technológie, kognitívny tútor, ACT-R, CTAT

1. ÚVOD – KOGNITÍVNY TÚTOR

Kognitívny tútor je moderná interaktívna digitálna pomôcka pre podporu procesu učenia sa. Využíva poznatky z kognitívnej vedy zastúpenej najmä kognitívnou psychológiou a modelovaním mysle pomocou počítačov. Je to počítačová aplikácia, ktorá pri riešení zadávaných úloh poskytuje študentovi okamžitú odozvu – ohlasuje chyby a vysvetľuje ich, poskytuje na vyžiadanie nápovede a kontroluje správnosť riešenia.

Prvé tútoringové programy vznikli ako pomocné aplikácie teórie ACT-R (Adaptive Control of Thought - Rational), aby lepšie pomohli pochopiť niektoré aspekty ľudského myslenia a učenia sa (v zmysle nadobúdania procedurálnych zručností). Prvou oblasťou, ktorá bola pre kognitívneho tútora spracovaná, bola matematika – tútor riešil lineárne rovnice. ACT-R je teória, ktorá aspiruje na vysvetlenie ľudskej kognície pomocou modelovania umelého systému, ktorý bude vykazovať rovnaké schopnosti ako ľudská myseľ. Základné **princípy teórie ACT-R** sú [3]:

- existujú dva typy pamäti – deklaratívna a procedurálna, v deklaratívnej pamäti sú uložené vedomosti (jednotlivé fakty, napríklad – pes je cicavec), procedurálna pamäť obsahuje postupy ako niečo urobiť (napríklad ako sčítať dve veľké čísla),
- popis interakcie medzi oboma typmi pamäti a popis sprístupnenia deklaratívnych poznatkov procedurálnym zručnostiam sú kľúčové pre vysvetlenie kognitívnych prejavov a myslenia,
- existujú mechanizmy, ktoré umožňujú vznik nových procedurálnych zručností (pomocou imitácie, analógie, interpretácie),
- stratégie riešenia problémov sú zakódované ako pravidlá (typu „AK stav_y, TAK akcia_x“),
- procedurálne zručnosti a deklaratívne poznatky sa upevňujú používaním.

Z týchto princípov priamo vychádza koncepcia učenia sa, ktorá stavia na tom, že viaceré domény (matematika, programovanie aj iné), sa dajú popísať ako súbor pravidiel. Napríklad úspešné vyriešenie lineárnej rovnice predstavuje použitie viacerých pravidiel („AK sa neznáma vyskytuje v tvare n.x, TAK vynásob obe strany rovnice číslom 1/n“ a veľa iných). **Učenie sa** potom spočíva

- v zapamätaní si faktov a princípov, ktoré figurujú v pravidlách (tvorba a upevňovanie obsahu deklaratívnej pamäti),
- v tréningu použitia týchto pravidiel (upevňovanie obsahu procedurálnej pamäti) a
- v tvorbe nových súvisiacich pravidiel (tvorba obsahu procedurálnej pamäti).

Kognitívne tútoringové programy sa zameriavajú na upevňovanie a tvorbu obsahu procedurálnej pamäti, pretože predpokladajú, že deklaratívne poznatky sú získané inou cestou a v ich zápise do pamäti nie je žiaden principiálny problém. Naproti tomu procedurálne zručnosti je nutné prakticky precvičovať, nestačí o nich hovoriť [1]. V tomto bode sa prístup kognitívneho tútoringu zhoduje s papertovským konštrukcionizmom.

Pre ideálny postup je dobré dodržať tieto dve pravidlá: 1. dôkladne analyzovať a rozčleniť danú oblasť na jednotlivé komponenty (pravidlá), ktoré reprezentujú zručnosti, 2. zabezpečiť, aby študent dobre zvládol jednoduchšie zručnosti pred tým, než prejde k zložitejším [2].

Častá námietka znie, že učenie sa nejde rozdeliť na takto izolované komponenty, pretože je to zložitý proces, kde sa nedá celok brať ako súčet častí. Ak by sa takto postupovalo, učili by sme izolované zručnosti, čo nezodpovedá princípu učenia sa v kontexte. Kognitívny tútoring však nemá za cieľ vysvetľovať celý kontext vzdelávania v danej oblasti, poskytuje iba prostriedky pre tréning dobre definovaných malých úloh. Učenie sa využitiu týchto zručností v širšom kontexte by nemalo byť zrejme realizované pomocou tútora, ale inou formou

vyučovania. Treba priznať, že niektoré domény môžu byť svojou podstatou nevhodné pre akékoľvek využitie tútorov.

2. CTAT TÚTOR PRE DELPHI

Pre tvorbu tútora pre začiatočníkov v programovaní sme sa rozhodli využiť nástroje CTAT^{*} (Cognitive Tutor Authoring Tools) vyvinuté na pracovisku LearnLab na univerzite Carnegie Mellon v USA. Hlavnou motiváciou pre vytvorenie takýchto nástrojov bola požiadavka na zníženie časovej náročnosti tvorby tútora. Zároveň sa otvorila možnosť pripraviť tútora pre vyučovanie aj bez hlbších znalostí programovania, čo umožňuje vytvárať tútoringové aplikácie širšej skupine ľudí (v ideálnom prípade celému spektru učiteľov na ZŠ, SŠ a VŠ).

Pomocou CTAT sa dajú vytvoriť *dva typy tútorov*:

- tútor trasujúci príklady (example tracing tutor) – jeho príprava je jednoduchšia, ale treba definovať všetky správne aj nesprávne postupy pri riešení pre každú úlohu zvlášť,
- pravý kognitívny tútor – vyžaduje programovanie v jazyku JESS, kde je nutné popísať všeobecné pravidlá riešenia úloh jedného typu.

Vytvorili sme tútorov (trasujúcich príklady) pre sériu deviatich úloh zameraných na deklaráciu premennej a priradenie do premennej v jazyku pascal. Tvorba tútora pozostávala z troch krokov:

- **Príprava užívateľského rozhrania**

Užívateľské rozhranie sme vytvorili v programe Flash. Využili sme predpripravené komponenty nevyhnutné pre fungovanie tútora, ktoré komunikujú s jeho skrytými mechanizmami. Okrem týchto komponentov sme považovali za vhodné doplniť signalizáciu splnenia úlohy a prehľad o vyriešených úlohách (Obr. 1).

- **Naplnenie grafu riešení**

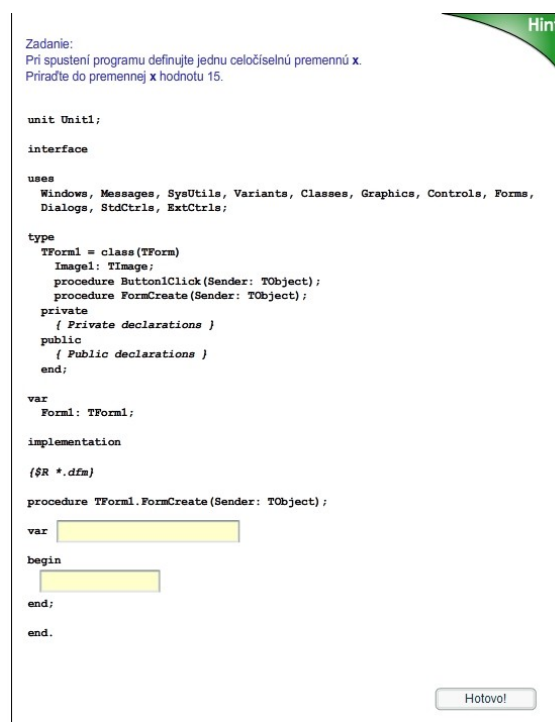
Graf riešení sa vytvára demonštráciou pomocou editora grafov, ktorý je súčasťou CTAT. V hotovom užívateľskom rozhraní (ktoré sa dá spustiť v ľubovoľnom internetovom prehliadači) sa predvádzajú správne aj nesprávne riešenia, ktoré sa zaznamenávajú do grafu. Predvádzať riešenie v tomto prípade znamená vpisovať do políček vstupy, vyberať z ponuky, stláčať tlačidlá a podobne. Priamo v zobrazenom grafe sa potom dopĺňajú aj texty nápovedy a texty chybových hlásení. Okrem toho je možné pomenovať jednotlivé kroky riešenia pre sprehľadnenie záznamu o interakcii študenta s tútorom

- **Sprístupnenie hotového tútora**

Pre zobrazovanie tútora cez internet a jeho funkčnosť treba umiestniť na server súbory Flash,

ktoré predstavujú užívateľské rozhranie, HTML súbory, ktoré zobrazia Flash súbory, textové súbory, v ktorých je zakódovaný graf riešení a podpornú aplikáciu, ktorá bude po celý čas na serveri spustená a bude zabezpečovať spoluprácu všetkých ostatných súborov. Potom je možné spúšťať tútora z akéhokoľvek počítača pripojeného na internet.

V našej prvej sérii sme pripravili dva typy úloh: 1. študent má vpisovať do prázdnych políček časti programu, 2. študent vyberá z niekoľkých možností správnu odpoveď.



① Obr. 1 Užívateľské rozhranie pre úlohu č.1

Treba poznamenať, že vlastnosťou tútora vytvoreného pomocou CTAT je to, že políčka sa nedajú vyplňať inak než postupne. Tútor nedokáže skontrolovať iba čiastočne správne riešenie, pretože riešenie je zakódované v stromovom grafe, v ktorom sa dá pohybovať po uzloch iba zhora nadol - tútor nepodporuje čiastočnú kontrolu niektorého podgrafu. Táto vlastnosť nie je ideálna z hľadiska didaktiky programovania, pretože niektoré typy úloh sa lepšie riešia nie riadok po riadku ale postupným vylepšovaním spočiatku „nesprávneho“ riešenia. Toto spolu s inými faktormi znemožňuje použiť takýto typ tútora pre zložitejšie úlohy z programovania. Treba si však uvedomiť, že to nie je všeobecná vlastnosť všetkých kognitívnych tútorov, naopak príklady z praxe dokazujú, že je možné vytvoriť veľmi kvalitný tútoringový program, ktorý zohľadní všetky požiadavky didaktiky programovania. CTAT predstavujú zjednodušenie

* Podrobnosti nájdete na stránke: <http://ctat.pact.cs.cmu.edu>

pre menej náročného tvorca, ktoré ide na úkor možností výsledného produktu, ale má iné výhody.

3. DÁTA ZÍSKANÉ PRI TESTOVANÍ TÚTORA

Tútor vytvorený pomocou CTAT dokáže zaznamenať všetky vstupy od študentov a všetky interakcie s prvkami ovládania. Tieto údaje spolu s časovou stopou a inými dátami sa ukladajú do textových súborov. Pre vytváranie korektného záznamu o práci študenta s tútorom je potrebné zabezpečiť, aby sa študenti mohli pred spustením identifikovať (potom je však nutné riešiť otázku spracovania osobných údajov), tým vieme priradiť jednému užívateľovi záznamy z jednotlivých úloh, ktoré riešil, a spracovať tak údaje o jeho postupe v celej sérii. Ak nepotrebujeme vedieť, ktoré dáta patria ktorému konkrétnemu študentovi, stačí, aby nejaký skript generoval unikátne identifikačné číslo pre každého, kto tútora spustí. Takto získané záznamy predstavujú cenný zdroj informácií o tom, ako študenti riešia zadané úlohy, ako interagujú s prvkami tútora, kedy otvárajú nápovede a podobne. Pre analýzu a spracovanie dát z tútorov existuje aplikácia DataShop, ktorú plánujeme v budúcnosti využiť.

Prvú sériu úloh zameranú na deklaráciu a priradenie do premennej sme testovali so skupinou študentov v prvom ročníku bakalárskeho štúdia učiteľstva všeobecnovzdelávacích predmetov v kombinácii s informatikou. Máme skúsenosti, že študenti týchto odborov prichádzajú často na univerzitu bez dostatočných skúseností s programovaním, veľa z nich sa s programovaním nikdy predtým nestretlo. S tútorom sme pracovali na nepovinnom cvičení za počítačmi, ktoré prebehlo v druhom týždni semestra a študenti teda mali za sebou minimálne jednu prednášku z tohto predmetu.

Počas pozorovania práce študentov s tútorom spolu s ďalšími tromi vyučujúcimi sme si všimli niekoľko javov:

- Pre prvákov je mátaúce ďalšie nové prostredie, ktoré sa musia naučiť ovládať a využívať. Ostáva na zváženie, či bude prínos tútora taký, aby túto nevýhodu vyvážil.
- Pre vyučujúceho je to ďalšia záťaž - tiež sa musí naučiť s tútorom pracovať, poznať jeho potenciál a obmedzenia.
- Niektoré technické problémy (tútor prestane kontrolovať správnosť vstupov, nutnosť vyplňať bunky postupne) a problémy s ovládaním programu (nemožnosť presúvať sa medzi políčkami iba pomocou klávesnice) znemožňujú jeho efektívne použitie.
- Študenti akoby nechceli veriť, že môžu používať nápovede v rámci tútora aj napriek tomu, že na možnosť nápovede bolo upozornené vopred. Radšej

sa pýtajú vyučujúcich, spolusediacich alebo študujú svoje poznámky z prednášky.

- Veľa študentov vyjadriло spokojnosť nad vyriešenou úlohou (dobré riešenie a splnenie úlohy je signalizované). Zrejme je dôležité, aby študenti vedeli, že majú správne riešenie a v akom štádiu riešenia celej série úloh sa nachádzajú. Považujeme to za dôležitý motivačný faktor zodpovedajúci filozofii „mastery learning“.

Ako študenti pracovali na úlohách zadávaných tútorom, všetky vstupy a akcie sa zaznamenávali do anonymných súborov (v tomto prípade sme nepoužili žiadnu identifikáciu študentov v tútorovi). Zo záznamových súborov sme zistili množinu častých nesprávnych riešení pre jednotlivé úlohy, tieto zapracujeme do novej verzie tútora a doplníme o príslušné chybové hlásenia. Okrem toho sme mali možnosť všimnúť si niektoré špecifické prístupy k riešeniu úloh v tútorovi a rôzne vnímanie ovládacích prvkov. Postupy pri riešení úloh by sa dali rozdeliť na „dobré“ a „zlé“ - dobré vedú pomerne rýchlo k splneniu celej úlohy, zlé priveľmi zvyšujú počet krokov, alebo nevedú k vyriešeniu úlohy (keďže všetci študenti prešli celou sériou úloh, je zrejme, že niektoré úlohy boli spustené jedným študentom viackrát, vo finále však úloha bola vždy vyriešená). Za „dobré“ postupy považujeme:

- **metóda „pokús-omyl“** - študent väčšinou nevie ihneď správne riešenie (typicky chýba bodkočiarka za príkazom), ale na niekoľko málo pokusov sa k nemu dopracuje, jeho chyba je veľa krát taká, že ju tútor rozozná a poskytne chybové hlásenie,
- **metóda „najprv nápoveda“** - bez toho, aby sa študent pokúšal o vlastné riešenie, vyžiada si hneď nápovedu, podľa ktorej spraví väčšinou správne riešenie, táto metóda je použitá iba zriedkavo,
- **metóda „nápoveda neskôr“** - po niekoľkých nesprávnych pokusoch si študent vyžiada nápovedu a podľa nej napíše správne riešenie, môže ísť o situáciu, keď študent vzdá metódu „pokús-omyl“ bez nápovede,

Použitie „zlého“ postupu môže byť zapríčinené nesprávnym pochopením ovládacích prvkov tútora, nepozornosťou alebo ignorovaním chybových hlásení. Za „zlé“ prístupy k riešeniu považujeme:

- **používanie tlačidla „HOTOVO“ ako kontroly správnosti** - veľa študentov stlačilo tlačidlo (ktoré slúži na ukončenie úlohy a presun na ďalšiu úlohu) napriek tomu, že tútor im v chybovej správe oznamoval, že ešte nemajú vyplnené všetky bunky, alebo že majú chybu v niektorom z políčok. Tlačidlo „HOTOVO“ takúto funkciu nemá a tútor v skutočnosti funguje tak, že kontroluje obsah daného políčka hneď ako sa dopíše vstup a označí sa myšou iné políčko. Je vhodné zvážiť, či by nemalo byť doplnené nejaké ďalšie tlačidlo (možno aj pre každú bunku zvlášť), ktoré by spĺňalo takúto funkciu. Iné vysvetlenie je, že chybné vstupy sú nedostatočne

označené, alebo že chybové hlásenia sú málo výrazné či nejasné.

- **Preskakovanie políčok** - niekoľko študentov sa snažilo vyplňať prázdne políčka v inom poradí, než bolo určené. Tútor nedokáže takéto riešenie vyhodnotiť, aj keby bolo čiastočne správne. Ak to bude technicky možné, považujeme za dôležité umožniť študentom vyplňať políčka riešenia v ľubovoľnom poradí.
- **Ignorovanie chyby vo vyššom políčku** – veľa nesprávnych riešení prebiehalo tak, že študent vyplnil všetky políčka, pričom niektoré z nich nesprávne, a označil úlohu za hotovú. Tútor označil prvé nesprávne políčko a požadoval od študenta, aby sa vrátil a opravil riešenie. Mnoho študentov nevedelo, ako pokračovať a často v takejto situácii opakovane žiadali nápoed', z ktorej však nevyčítali, kde sa stala chyba. Takýto postup je v rámci práce s tútorom (prihliadnuc na jeho obmedzenia) veľmi nevhodný, pokúsime sa to vyriešiť lepšou signalizáciou správnosti/nesprávnosti vstupu. Možno by bolo dobré, aby neboli aktívne tie políčka, ktoré tútor aktuálne nedokáže skontrolovať. Tým by sme však prišli o záznamy študentov, ktorí majú snahu takto úlohu riešiť, čo zatiaľ považujeme za hodnotnú informáciu.

Veríme, že viacero problémov by sa odstránilo, keby mali študenti možnosť pracovať s tútorom opakovane a zvyknúť si na niektoré špecifiká ovládania a funkčnosti. Naším cieľom však je, aby ovládanie bolo čo najintuitívnejšie a funkčnosť pochopiteľná pre začiatočníkov v programovaní.

4. NAVRHOVANÉ ZMENY A VYLEPŠENIA

Prednášky k predmetu - úprava a nový koncept

Pre efektívne využitie tútorov v kurze programovania by bolo vhodné upraviť texty prednášok a zadania povinných a nepovinných cvičení, tak aby spĺňali požiadavky prístupu k učeniu sa podľa teórie ACT-R. Predovšetkým by sa malo zabezpečiť, aby materiál k prednáške pokrýval na 100 % obsah cvičení a domácich úloh. Ďalším z najdôležitejších bodov bude upraviť cvičenia tak, aby boli úlohy vždy stupňované od najjednoduchších po najzložitejšie.

Úvodný kurz k programovaniu

Po zvážení situácie s novými študentami a v rámci tvorby tútorov pre vyučovanie programovania sme sa rozhodli pripraviť úvodný kurz programovania pre úplných začiatočníkov, ktorý by bol mimo semestra. Domnievame sa, že je to najvhodnejšie miesto pre využitie tútorov. Takýto kurz by pomohol zjednotiť úroveň študentov v prvom ročníku a tým umožnil ich kvalitnejšie vzdelávanie počas semestra.

Identifikácia študentov pre zber dát z tútora
Aby sme dokázali vyťažiť maximum z dát, ktoré

tútor zbiera a následne ich dokázali analyzovať pomocou aplikácie na to určenej, potrebujeme zabezpečiť identifikáciu jednotlivých študentov pred spustením tútora (pomocou zadania mena alebo kódu).

Úpravy prvej série úloh:

- doplnenie rozpoznávaných chýb a príslušných chybových hlásení podľa chybných vstupov získaných z prvého testovania,
- zvýraznenie správnosti/nesprávnosti vstupu,
- preklad chybových hlásení, ktoré sú v anglickom jazyku,
- oprava ovládania pomocou klávesnice,
- kontrola čiastočných riešení – ak bude možné upraviť nástroje na tvorbu tútorov, tak aby to podporovali.

5. ZÁVER

Vyskúšali sme si tvorbu série tútoringových programov pre vyučovanie. Aj keď nástroje na tvorbu tútorov sú pomerne obmedzené a vyžadujú od tvorcu viacero kompromisov, domnievame sa, že majú veľký potenciál v mnohých doménach vzdelávania. Digitálne technológie, ktoré môžu obohatiť vyučovanie je potrebné naďalej skúmať, pretože dávajú možnosť individualizovať vyučovanie a zároveň skúmať pravidelnosti v javoch, ktoré by nám inak ostali ukryté.

LITERATÚRA

- [1] Anderson, J.R., Corbett, A.T., Koedinger, K.R., Pelletier, R.: Cognitive Tutors: Lessons Learned. In *The Journal of the Learning Sciences*. 1995, 4(2), s. 167 - 207.
- [2] Corbett, A.T., Anderson, J.R.: Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge. In *User Modeling and User-Adapted Interaction*. 1995, 4, s. 253 - 278.
- [3] Anderson, J.R.: How Can the Human Mind Occur in the Physical Universe? New York : Oxford University Press, 2007. ISBN 978-0-19-532425-9
- [4] Anderson, J.R., Gluck, K.: What role do cognitive architectures play in intelligent tutoring systems? In Klahr, D., Carver, S.M. (eds.): *Cognition & Instruction: Twenty-five years of progress*. Mahwah, NJ : Erlbaum, 2001, s. 227 – 262. ISBN 0-89859-689-0
- [5] Koedinger, K.R., Alevan, V., Heffernan, N., McLaren, B., Hockenberry, M.: Opening the Door to Non-programmers: Authoring Intelligent Tutor Behavior by Demonstration. In *Proceedings of 7th Annual Intelligent Tutoring Systems Conference*. Maceio, Brazil, 2004.