AUTORSKÉ PROSTŘEDKY

Mgr. Jan Preclík

Abstrakt: Autorské prostředky jsou softwarové nástroje určené k tvorbě výukových aplikací. V příspěvku je zmiňován vývoj autorských prostředků, jejich klasifikace a typičtí zástupci jednotlivých kategorií.

Klíčová slova: Autorské systémy, programovací jazyky, autorské jazyky, výukové aplikace.

1 ÚVOD

Rozhodneme-li se používat počítač i při výuce "neinformatických" předmětů (biologie, zeměpis…), budeme k tomu potřebovat, mimo jiné, i vhodné ***výukové aplikace*** (anglicky *educational software* nebo *courseware*).

Nevyhovují-li nám prodávané programy (ať už z důvodu špatné kvality, nevhodného zpracování učební látky nebo kvůli chybějící lokalizaci[1]), můžeme se pokusit vytvořit program vlastní. Budeme k tomu potřebovat především nápad, odborné a pedagogické znalosti, spoustu času a nadšení a také některý z programů (softwarových nástrojů), který nám tvorbu umožní a usnadní. Programy určené pro tvorbu výukových aplikací se obecně nazývají ***autorské prostředky*** (*authoring tools*).

2 KLASIFIKACE A VÝVOJ AUTORSKÝCH PROSTŘEDKŮ

2.1 Programovací jazyky

Nejprve se pro tvorbu výukových programů používaly ***programovací jazyky*** *(programming languages*), například *Basic*, *Fortran*, *C/C++*, *Pascal*… Programovací jazyky jsou velmi silným nástrojem pro vývoj jakéhokoli typu aplikace, což je, pokud se týká vývoje výukových programů, jejich hlavní nevýhoda – jsou příliš obecné a všechno je v nich třeba programovat od začátku.

S příchodem moderních vývojových prostředí umožňujících ***vizuální programování*** (*Borland Delphi*, *Visual Basic*, *Visual C++*) mají programátoři dosti ulehčenou práci (především co se týče návrhu vzhledu aplikace, dialogových oken… – tzv. *front-end design*), ale stále je třeba znát syntaxi a sémantiku použitého programovacího jazyka. Pro ošetření jednotlivých ***událostí*** (*events*), např. stisku tlačítka nebo klávesy, pohybu myší, je stále nutno psát přímo programový kód.

Programovací jazyky ale nelze z hlediska tvorby výukových programů zcela zavrhnout. Většina moderních autorských systémů (viz dále) umožňuje volání externích programů (spustitelných exe souborů, knihoven *DLL* nebo *ActiveX* prvků) realizujících nestandardní funkce, jež nejsou autorským systémem přímo podporované.

---

[1] Lokalizovat program neznamená pouze přeložit veškeré texty do češtiny, ale také uzpůsobit chování programu národním zvyklostem (různé způsoby zápisu data a času, desetinný oddělovač …).

2.2 Autorské jazyky

Programovací jazyky nejsou pro nezkušené a začínající uživatele příliš vhodné, proto jejich zjednodušením a specializací vznikly ***autorské jazyky*** (*authoring languages*).

Autorské jazyky nejsou zcela obecné – nedá se v nich naprogramovat úplně vše. Obsahují méně příkazů, které jsou ale komplexnější a určené speciálně pro tvorbu výukových programů (např. výběrovou otázku lze většinou vytvořit jediným příkazem). Stále je zde nutno programovat (kódovat)[2], učit se příkazy jazyka a jejich použití.

Výhodou a současně nevýhodou autorských jazyků je jejich specializace – snadno v nich lze realizovat pouze to, k čemu jsou určeny. Jakékoli nestandardní věci je nutno programovat úplně od začátku a je to většinou obtížnější než v klasickém programovacím jazyce. Někdy je to zcela nemožné.

Jedním z prvních autorských jazyků byl jazyk *PILOT* (vyvinutý firmou IBM roku 1962, používal se v éře sálových počítačů, terminálů a znakového výstupu).

Příkladem českého systému založeného na použití autorského jazyka je *Interaktivní výuka* (http://www.volny.cz/edusoft). Základem je textový soubor (ASCII) nebo soubor formátu RTF (*Rich Text Format*) – je tedy možno používat obrázky a běžné formátování. Tento soubor obsahuje kromě textů otázek a odpovědí i řídící příkazy pro definování správné odpovědi, následující otázky…

Mezi autorské jazyky lze zahrnout i jazyk ***HTML*** (*HyperText Markup Language*) sloužící pro tvorbu WWW stránek.

2.3 Autorské systémy

Dnes se pro tvorbu výukových programů používají ***autorské systémy*** (*authoring systems*). Jsou to komplexní vývojová prostředí umožňující i uživatelům neznalým programování vytvářet vlastní aplikace. Pro tvorbu jednodušších aplikací se není třeba učit příkazy žádného jazyka, vše se odehrává interaktivně – výběrem z menu, pomocí ***průvodců*** (*wizards*) nebo používáním předpřipravených prvků (***komponent***), ze kterých se celá aplikace poskládá dohromady.

Autorské systémy většinou dovolují i programovat, často pomocí jednoduchého skriptového jazyka. Je vhodné, aby možnost programování v systému existovala, ale aby nebyla potřeba při tvorbě jednodušších aplikací.

Dle údajů uvedených v [3] zkracují autorské systémy dobu potřebnou k tvorbě výukového programu až na 1/8 v porovnání s dobou potřebnou k vývoji při použití programovacího jazyka. Je nutno si ale uvědomit, že toto urychlení se týká pouze doby potřebné k programování. Doba potřebná ke shromáždění a uspořádání potřebných materiálů nebo k jejich tvorbě zůstává stále stejná a u rozsáhlejších projektů tato doba několikanásobně převyšuje dobu nutnou k programování.

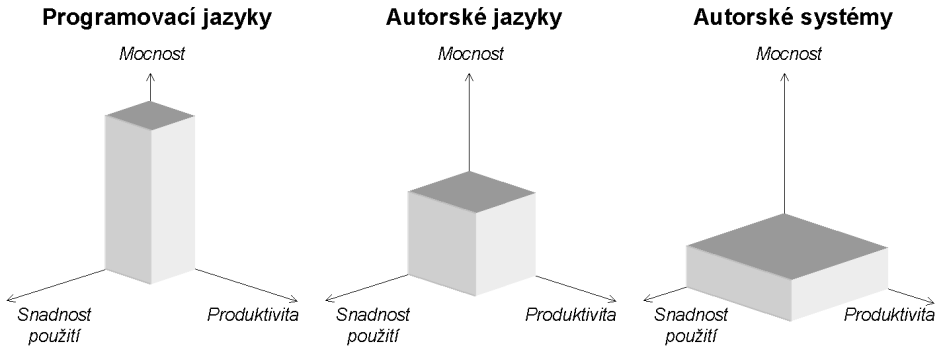
Použití autorských systémů je jednoduché i pro laika, na druhé straně náročnějšího autora do jisté míry omezují, neumožňují realizovat úplně vše, ale jen to, k čemu jsou určeny (podobně jako autorské jazyky). Proto je důležité, aby umožňovaly volání externích programů realizujících ty funkce, které nejsou autorským systémem přímo podporovány. Dosti velkou nevýhodou autorských systémů je i jejich cena pohybující se v desetitisících korun.

---

[2] Pod pojmem programovat (kódovat) se zde i v dalším textu rozumí zapisování příkazů programovacího jazyka v textové podobě.

3 POROVNÁNÍ AUTORSKÝCH PROSTŘEDKŮ

Popsané druhy autorských prostředků můžeme porovnat pomocí tří kritérií: ***Mocnost*** (*Power*) – co vše lze v systému realizovat, ***Produktivita*** (*Productivity*) – jak velkou část aplikace vytvoří jeden autor za jednotku času, ***Snadnost použití*** (*Easy-of-Use*) – za jak dlouho se lze se systémem naučit pracovat. Dostáváme tzv. ***PPE diagramy*** [4].



Obr. 1 – PPE diagramy

4 VÝVOJ AUTORSKÝCH SYSTÉMŮ

První autorské systémy vznikaly především s rozvojem grafických uživatelských prostředí – byly to v podstatě ***nadstavby autorských jazyků***. Myšlenka použít jako základ systému autorský jazyk určený pro zkušené autory a grafickou nadstavbu pro začátečníky není vůbec špatná. Ukazuje se, že začátečníkům vyhovuje spíše ovládání ve stylu Windows (*GUI – Graphic User Interface*, grafické uživatelské rozhraní), kdežto experti (zkušení uživatelé) dávají přednost zápisu příkazů v textové podobě (v textovém režimu lze mnoho věcí zautomatizovat – např. pomocí maker).

Modernější autorské systémy jsou založeny na ***hypertextu***. Patří sem např. *LinkWay* (jeden z prvních hypertextových systémů pro *MS DOS*), pro Macintosh známý a velmi oblíbený *HyperCard* (http://www.apple.com/hypercard) a jeho obdoba pro Windows *ToolBook* (http://www.asymetrix.com/products). Do této kategorie lze zahrnout i systémy pro tvorbu a vizuální návrh WWW stránek – například *MS FrontPage* nebo *Netscape Composer*.

Nejmodernější jsou ***objektově orientované autorské systémy***. Autor v nich má k dispozici množství předpřipravených objektů (nazývaných *prvky*, *komponenty*, *ikony*), ze kterých sestavuje jednotlivé části výsledné aplikace. U objektů lze nastavovat jejich ***vlastnosti*** (*properties*) – týkají se především vzhledu objektu, např. barva, písmo… a reakci na různé ***události*** (*events*) – např. co se má stát, když uživatel stiskne tlačítko.

5 DĚLENÍ AUTORSKÝCH SYSTÉMŮ

Podle filosofie, kterou používají při tvorbě aplikace, lze autorské systémy rozdělit na:

- ***Stránkově orientované*** (*card-based*): Aplikace se skládá z jednotlivých ***stránek*** (karet). Přechody mezi stránkami jsou často realizovány na principu hypertextových odkazů.
  Do této kategorie patří většina hypertextových systémů (již zmiňovaný *LinkWay*, *HyperCard*, *Toolbook*) nebo systém *Athelas* (http://www.gymnachod.cz/~preclik).

- ***Systémy založené na použití ikon*** (*iconic / flow control*) obsahují ***paletu ikon*** (*icon palette*), které se umísťují do ***postupového diagramu*** (*flow-chart*) znázorňujícího běh a strukturu programu. Ikony slouží k zobrazení objektu na obrazovce, k animaci vybraného objektu, k větvení běhu programu, ošetření reakce uživatele. Tvorba aplikace připomíná sestavování vývojových diagramů známých ze strukturovaného programování.
  Příkladem takového systému je jeden z prvních objektově orientovaných autorských systémů *Macromedia Authorware* (http://www.macromedia.com/software/authorware).
- ***Časově orientované autorské systémy*** (*time-based*) obsahují ***časovou osu*** (*time-line*), na které jsou vyznačeny začátky a konce různých akcí (zobrazení objektu, přehrávání zvuku, videa…). Tyto programy jsou zvláště vhodné pro tvorbu multimediálních prezentací. Příklad: *Macromedia Director* (http://www.macromedia.com/software/director).
- ***Generátory testů*** umožňují vytvářet a spravovat databáze otázek a odpovědí, ze kterých se generují (náhodně nebo přímým výběrem) otázky do jednotlivých testů. Vygenerované testy lze vypracovávat přímo na počítači nebo je vytisknout (i s klíčem) a použít klasickým způsobem. Příklad: český systém *DoTest* (http://www.dosli.cz).

6 ZÁVĚR

Autorské systémy představují velice silný a flexibilní nástroj pro vývoj výukových aplikací vhodný i pro začínající a méně zkušené uživatele. To s sebou nese i jeden negativní důsledek. Autorské systémy mohou zvýšit kvalitu vytvářených materiálů, ale nezaručují ji. Ukazuje se ([5]), že autorské systémy mohou přispívat k celkovému poklesu v kvalitě vytvořených výukových programů v posledních letech, neboť zpřístupňují tvorbu výukových programů každému bez ohledu na jeho zkušenosti, znalost a pochopení zpracovávané látky a pedagogické dovednosti. Výukových programů tedy vzniká více, ale jejich kvalita je v průměru horší. Dříve byl pro tvorbu výukového programu potřeba celý tým, což sice bylo nákladnější a programů se tvořilo mnohem méně, ale kvalita byla lepší.

Na druhé straně je jisté, že se dnes při vývoji moderní multimediální výukové aplikace bez nějakého autorského systému neobejdeme.

7 LITERATURA

[1] Brdička Bořivoj: *Učení s počítačem*; [online],
URL: http://omicron.felk.cvut.cz/cgi-bin/charset/~bobr/ucebnice.html

[2] Dean Christopher: *Authoring Systems for CBT and Interactive Multimedia*, Dean Associates & Technologies for Training Ltd, 1999; [online], URL: http://www.tft.co.uk/products/auth_systs.html

[3] Siglar Jamie: *Multimedia Authoring Systems FAQ*, 1999; [online],
URL: http://www.tiac.net/users/jasiglar/faq_index.html

[4] Newton J.: *Making the Right Choice*, IT Training, October/November 1992, str. 27

[5] Locatis Craig, Ullmer Eldon, Carr Victor, Banvard Richard, Lo Raulie, Le Quang, Williamson Natthew: *Authoring Systems*, LISTER HILL Monograph: LHNCBC 92-1, 1992; [online],
URL: http://cgsb.nlm.nih.gov

Mgr. Jan Preclík
KSVI, MFF UK
Malostranské náměstí 25, 118 00 Praha 1 – Malá Strana, ČR
tel: +420-2-21914217, fax: +420-2-21914281
e-mail: preclik@ksvi.mff.cuni.cz