

Žmogaus ir kompiuterio sąveikos dėstymo metodikų apžvalga

Kristina LAPIN (VU)

el. paštas: kristina.lapin@maf.vu.lt

Reziumė. Šiame straipsnyje yra nagrinėjamos fizinių ir socialinių mokslų bei inžinerijos mokymo metodikos žmogaus ir kompiuterių sąveikos dėstymo kontekste. Analizuojamos tradicinė, atvejais ir šablonais paremtos metodikos bei jų realizacijos žmogaus ir kompiuterių sąveikos kurse.

Raktiniai žodžiai: mokymo metodika, žmogaus ir kompiuterio sąveika.

1. Įvadas

Žmogaus ir kompiuterio sąveikos (ŽKS) dalyko įtraukimas į studijų programas paskatino domėjimąsi efektyviais šio dalyko mokymo būdais. ŽKS disciplina atsirado mokslų, inžinerijos ir dailės sankirtoje [5]. Kiek šių sričių nusistovėjusios mokymo metodikos gali būti naudojamos ŽKS dalyko mokyme? Kaip ŽKS turi būti dėstoma: kaip mokslų, inžinerijos ar dailės dalykas?

Šiame straipsnyje autorė apžvelgia mokslų, inžinerijos ir dailės metodinių palikimą ir analizuoja šių sričių pedagoginę patirtį ŽKS dėstymo kontekste.

2. Mokslų, inžinerijos ir dizaino metodinis palikimas

Svarbiausias ŽKS aspektas informatikos požiūriu yra sąsajų projektavimas. ŽKS vystosi kaip mokslo disciplina [7], nes esamos sąsajos yra stebimi, iškeliamos jų tobulinimo hipotezės, kurios yra vertinamos. Sukauptos žinios formuoja teorijas. Analizuojamos sąsajos yra skaidomos į pagrindinius elementus. Gaunami dėsniai, aprašantys šių elementų sąveiką. Nauji faktai yra analizuojami ir vėl yra keliamos hipotezės, numatomi vertinimo rezultatai.

ŽKS gali būti nagrinėjamas ir kaip inžinerijos dalykas. Inžinerinė disciplina nusako, kokia tvarka, kuo remiantis ir kokius sprendimus reikia priimti projektuojant ir konstruojant programų sistemas [2]. Atsižvelgiant į inžinerijos tikslus, pagrindiniai ŽKS siekiai yra sąsajų efektyvumas ir patikimumas [7]. Projektavimo dalykuose studentai yra mokomi spręsti uždavinius, remiantis nustatytais procedūromis ir analitiniais metodais.

Programų sistemų projektavimo racionalistinė tradicija yra paremta deterministiniu modeliu [13], kai pavienės idėjos, požiūriai, interesai ir jausmai nekeičia uždavinio esmės. Tačiau šis modelis yra nepakankamas ŽKS kontekste, nes žmogaus nuomonės ir jausmai yra iškeliami į tikslų lygmenį.

ŽKS turi dizaino elementų. Kūrybiškas dizaineris abstraktų šabloną pritaiko konkrečiai užduočiai, balansuodamas tarp įvairių ribojimų, būtent: kontrasto, spalvų derinimo, elementų išdėstymo. Šie ribojimai turi būti derinami su efektyviu informacijos perteikimu ir patogia naudotojo bei programos sąveika. Šiuo požiūriu dėstytojas turi išmokyti studentus suteikti sąsajai originalumą, teikti pasitenkinimą ir kelti teigiamą emocinį atsaką.

3. Tradicinė metodika

ŽKS mokymui yra keliamas tikslas suteikti studentams praktinių įgūdžių aibę bei daugybės abstrakčių sąvokų išsivadinimą. Dėl daugialypės ŽKS prigimties, šios sąvokos yra integruotos į dalyką pakankamai laisvai. Nors šios sąvokos atspindi svarbiausias ŽKS kryptis, jos yra nesusietos su projektavimo procesu [9].

Tradicinėje dėstyimo metodikoje dominuoja paskaitos, kuriose yra pateikiamos sąvokos, principai ir faktai. Jose taip pat nagrinėjami trumpi pavyzdžiai, kuriuose yra iliustruojamas abstrakčių sąvokų taikymas. Praktinių užsiėmimų metu studentai atlieka naudotojų analizės, sąsajos prototipo projektavimo, vertinimo ir realizavimo užduotis. Rezultatai yra dokumentuojami ir dėstytojas juos vertina [10,3].

Šis metodas turi kelis trūkumus. Prototipo kūrimas reikalauja daug pratybų laiko ir nukreipia dėmesį nuo ŽKS mokymo tikslų. Be realaus užsakovo sudėtinga atlikti kokybišką naudotojo analizę. Mokymo uždaviniuose yra keliami supaprastinti reikalavimai. Šie ribojimai verčia atsisakyti užbaigti projektus arba užbaigti juos, skiriant laiką programavimui ir nesigilinant į ŽKS sąvokų prasmę.

4. Atvejų metodas

Avejų (angl. *case-based*) metodas yra skirtas suderinti sąvokų išsivadinimo ir praktinių įgūdžių igijimo prieštaravimus. Jis yra vertinamas kaip perspektyviausias metodas ŽKS mokyme [7]. Atvejų metodas naudoja atvejų aprašus, skirtus padėti išmokti ir naudoti svarbiausias sąvokas [4]. Atvejų metodas dar nėra paplitęs informatikos mokyme, todėl panagrinėkime populiariausius šiuo metodu vedamų užsiėmimų realizacijos būdus:

Diskusijų metodas yra naudojamas verslo ir teisės mokyme. Studentams yra pateikiamos situacijos. Studentai kartu su dėstytoju identifikuoja problemą, įvertina galimus sprendimus ir veiksmų pasėkmes. Dėstytojas formuluoja nukreipiančius klausimus, o studentai analizuoja situaciją. Dėstytojas išdėsto įvadą, nukreipia diskusiją ir užsiėmimo pabaigoje apibendrina diskusiją. Optimalus studentų skaičius – nuo 20 iki 60 [4].

Debatų taip pat dominuoja socialinių mokslų mokyme, kai yra galimi du priešingi sprendimai. Dvi grupės po 3 narius ruošia argumentus. Debatų pradžioje trumpai pristatomos pozicijos. Klausytojai turi galimybę klausti ir reikšti savo nuomonę.

Viešieji klausymai yra tinkamas būdas išklausti daug skirtingų nuomonių. Nuomonės pristatomos klausančiai komisijai. Klausymai gali vykti keletą užsiėmimų. Komisijos nariai apklausia savo nuomonę ginančius studentus. Pabaigoje yra pateikiamas komisijos sprendimas arba rekomendacija.

Uždaviniais pagrįstas mokymas – tradicinis mokymo būdas medicinoje. Nedidelės studentų grupės nagrinėja uždavinius, kuriuos sieja nagrinėjama tema arba kylantis sudėtingumo lygmuo. Trijų susitikimų metu yra analizuojami duomenys, renkama informacija ir tokiu būdu artėjama prie diagnozės ir gydymo nustatymo. Šio metodo privalumas yra mąstymo, diskusijos ir informacijos paieškos interaktyvumas.

Moksliniai tyrimai apima duomenų rinkimą, apdorojimą ir apibendrinimą lentelėse ir grafikuose. Kita vertus, gali būti atliekami eksperimentai, analizuojami jų rezultatai. Jais remiantis, yra tobulinamas eksperimentas. Remiantis eksperimento rezultatais yra rašomas klasikinės struktūros mokslinis darbas, turintis įvadą, metodus, rezultatus, išvadas ir nuorodas. Grupės pasikeičia savo straipsniais ir recenzuoja juos. Straipsniai ir recenzijos yra pateikiamos dėstytojui įvertinti.

Grupinis mokymas buvo įvestas Oklahomos universitete. Nuo 5 iki 7 studentų grupės individualiai skaito naujos temos medžiagą apie 15 min. Toliau kiekvienas studentas sprendžia testą. Individualūs atsakymai yra aptariami grupėse ir rengiamas grupės atsakymas. Dėstytojas vertina grupės atsakymus, o studentai sprendžia uždavinius, taikydami naujas žinias.

Remiantis atlikta analize [9], rekomenduojama naudotis šiais praktikoje pasitaikiusių atvejų aprašų šaltiniais:

Einamosiosiose publikacijose yra išdėstomos sąsajų kūrimo naujovės, pavyzdžiui Nielseno tinklalapio praktiškumo vertinimas¹. Nors jose paprastai yra gausu technologinių ir teorinių subtilybių, jos paprastai sudomina studentus. Publikacijų silpnoji pusė yra faktas, kad jos nepilnai atspindi projektavimo veiklą. Jose dažnai nagrinėjami tik reikalavimai, ankstyvojo projektavimo sprendimai ir prototipai.

Profesionalūs rinkiniai yra atvejų bibliotekos, kurias rengia ekspertai, pavyzdžiui pripažintų sėkmingais atvejų biblioteka². Bibliotekose yra konkrečių projektų pilna projektinė dokumentacija. Šios medžiagos privalumas yra jos pilnumas. Studentams yra pateikiamos visų projekto stadijų sąvokos, kiekvieno etapo dokumentacija, tuo pačiu jie gauna projektinius sprendimus įtakojusius faktorius. Trūkumas yra medžiagos gausumas. Be papildomų paieškos galimybių šią medžiagą yra sunku aprėpti.

Pažįstamų sąsajų, būtent piešimo priemonių, pokalbių ir pašto programų, tekstų doroklių privalumas yra faktas, kad analizės objektas yra gerai pažįstamas. Trūkumas – jie gali pasirodyti studentams nuobodūs.

5. Šablonais paremtas mokymas

Projektavimo patirtis yra struktūrizuojama rekomendacijose arba šablonuose. Šablonais paremtas mokymas yra priimtas architektūros studijose. M. Kapor savo darbe [6] teigia, kad programų sistemų kūrėjas turi būti mokomas labiau kaip architektas nei kaip programuotojas. Naudotojo sąsajos kūrėjui, kaip ir architektui, gaminio estetiškumas yra labiau artimas nei programuotojui. Pastarajam labiau rūpi techniniai programų sistemų kūrimo aspektai.

¹http://www.useit.com/papers/1994_web_usability_report.html

²<http://ucs.cs.vt.edu>

Naudotojo sąsajos projektavimo patirtis yra apibendrinama rekomendacijose, kurių tikslas yra padėti projektuotojui priimti tinkamus sprendimus ir vengti projektavimo klaidų. Tačiau rekomendacijų taikymas nėra trivialus, jos dažnai yra per daug abstrakčios. Dėl jų gausos sunku pasirinkti, kuri rekomendacija tinka konkrečioje situacijoje. Ta pati rekomendacija viename kontekste gali būti teisinga, kitame – klaidinga [12].

Projektavimo šablonai yra skirti pradedančiajam. Priešingai nei rekomendacijos, šablonai yra konkretūs ir yra paremti gero projektavimo principais. Šablonai išsprendžia konteksto problemą. Juose išreikštiniu būdu yra nurodoma, kada, kaip ir kodėl tam tikras sprendimas turi būti taikomas [12].

Naudotojo sąsajų šablonų formatai yra panašūs į architektūros šablonų formatą. Šablonų nagrinėjimą palengvina jų bibliotekos, kaip antai Sari L. Laakso³ ir Jennifer Tidwell⁴ parengtos bibliotekos. Šablonų aprašui yra sukurtos aprašo kalbos, kaip antai, Tidwell šablonų aprašo kalba [11].

Mokymą, paremtą šablonais, galima realizuoti, sprendžiant nedideles užduotis. J. Borchers pripažįsta šablonų naudingumą projektavimo principų mokyme ir siūlo šablonų naudojimo mokymo būdus [1]. Jo manymu, šablonų naudojimas žymiai pagerina studentų atliekamų užduočių kokybę.

6. Išvados

Tradicinis mokymo būdas leidžia efektyviai perteikti struktūrizuotas, faktines žinias dideliame klausytojų skaičiui. Pratybų metu 4–6 asmenų grupes atlieka nedideles užduotis, remiantis projektavimo rekomendacijomis. Šis metodas turi rimtą trūkumą: nors abstrakčios sąvokos yra griežtai apibrėžiamos ir iliustruojamos pavyzdžiais, klausytojui dažnai jos lieka nusisietos su realaus gyvenimo patirtimi. Todėl klausytojai yra linkę atlikti projektavimo užduotis nevisiškai suvokiant atliekamos veiklos prasmės ir priežasčių.

Šablonai tiksliau pavaizduoja projektavimo žinias nei rekomendacijos, nes jose ryškiau parodytas kontekstas. Kadangi šablonai nusako kontekstą, jie labiau asocijuojasi su konkrečiu uždaviniu, todėl pradedančiajam juos yra lengviau naudoti nei rekomendacijas. Jau pirmieji studentų sąsajų kūrimo bandymai yra kokybiškesni nei įprastu metodu mokomų studentų darbai. Vienas iš šablonų trūkumų yra tai, kad jie neskatina kūrybiškumo ir originalumo. Mokymo, paremto šablonais, šalininkai sako, kad kūrybiškumą jie suvokia, kaip šablono pritaikymą naujam uždaviniui.

Atvejų metodas tinka tobulinti aukštesnio lygio samprotavimų, argumentacijos igūdžius. Jei atvejai dominuoja kurse, tuomet nespėjama apimti visos kurso medžiagos. Atvejų metodas tinka dėstant projektavimo, vertinimo bei socialinį aspektą turinčias temas, tačiau šio metodo taikymas yra brangesnis nei tradicinis. Jis riboja dalyvių skaičių ir reikalauja kiekvienai temai skirti daugiau laiko, nei būtų skirta tradicinėje paskaitoje. Atvejų metodas nėra tinkamas apžvalginėms temoms, kuriose yra daugybė principų, faktų ir brėžinių.

³<http://www.cs.helsinki.fi/u/salaakso/patterns/>

⁴<http://time-tripper.com/uipatterns/index.php>

Literatūra

1. J. Borchers, Teaching HCI patterns: experience from two university courses, position paper for "Patterns in Practice: A Workshop for UI Designers", in: *Proceedings of ACM CHI 2002 International Conference on Human Factors of Computing Systems*, Minneapolis, MI (2002), pp. 21–25.
2. A. Čaplinskas, *Programų sistemų inžinerija*, I dalis, MII, Vilnius (1996).
3. S. Greenberg, Teaching human-computer interaction to programmers, *ACM Interactions*, 3(4), 62–76 (1996).
4. C.F. Herreid, Case studies in science: a novel method of science education, *Journal of College Science Teaching*, 23(2), 221–229 (1994).
5. T.T. Hewett, R. Baecker, S. Card, T. Carey, J. Gasen, M. Mantei, G. Perlman, G. Strong, W. Verplank, *ACM SIGCHI Curricula for Human-Computer Interaction*, Association for computing machinery (1996).
<http://sigchi.org/cdg/index.html>
6. M. Kapor, A software design manifesto, *Dr. Dobb's Journal*, 16(1), 62–67 (1991).
http://www.kapor.com/homepages/mkapor/Software_Design_Manifesto.html
7. D.S. McCrickard, C.M. Chewar, J. Somervell, Design, science, and engineering topics? Teaching HCI with a unified method, in: *Proceedings of the Technical Symposium on Computer Science Education (SigCSE'04)*, Norfolk VA (2004), pp. 31–35.
<http://csgrad.cs.vt.edu/~jsomerve/papers/mcs-sigcse-04.pdf>
8. P. Purgathofer, Design and software engineer, *Exchange Online Journal*, 1 (2000).
http://www.media.uwe.ac.uk/exchange_online1/exch1_article6.php3
9. J. Somervell, C.M. Chewar, D. Scott McCrickard, Making a case for HCI: comparing materials on case-based teaching, in: *Proc. of the 34th ASEE/IEEE Conf. Frontiers in Education Conference: Expanding Educational Opportunities through Partnership and Distance Learning*, Savannah, Georgia (2004).
10. G.W. Strong, *A Report: New Directions in Human-Computer Interaction Education, Research and Practise*, Report of a project funded by the NSF (planning grant IRI-9322659) and ARPA (1994).
<http://www.sei.cmu.edu/community/hci/directions/>
11. J. Tidwell, *Common ground: A Pattern Language for Human-Computer Interface Design* (1999).
http://www.mit.edu/~jtidwell/common_ground_onefile.html
12. M. van Welie, G.C. van der Veer, A. Eliëns, Patterns as tools for user interface design, *International Workshop on Tools for Working with Guidelines*, France, 313–324 (2000).
<http://www.welie.com/patterns/literature.html>
13. J.R.G. Wood, A.T. Wood-Harper, Information technology in support of individual decision-making, *Journal of Info Systems*, 3, 85–101 (1993).

SUMMARY

K. Lapin. Survey of the teaching methods of human computer interaction

The paper surveys teaching methods of both sciences and engineering that are implemented in the Human Computer Interaction courses. It deals with traditional lecture-based, case-based and pattern-based teaching.

Keywords: teaching methodology, HCI.