



COMPUTERS & LEARNING

2019/1

Volume: 2

Number: 1

Papers

Applications of human-computer interfaces and related psychological-educational aspects

Ember-számítógép interfészek alkalmazásai és ezek kapcsolódó pszichológiai-oktatási vonatkozásai
Gergely Sziládi 1

Advantages of Computer Network Simulation Environment in Education

Számítógép hálózati szimulációs környezet előnyei az oktatásban
László Greben 15

Importance of project-based learning in software development

A projekt alapú tanulás fontossága a szoftverfejlesztésben
György Schrauf.....27

Implementing a Java EE based information system as a student self-study task

Java EE alapú információs rendszer megvalósítása önálló tanulói feladatmegoldás során
Csaba Csizmadia 40

Editor-in-Chief

Attila Kővári, University of Dunaújváros

József Katona, University of Dunaújváros

Associate Editors

Bálint Nagy, University of Dunaújváros

Tibor Ujbányi, University of Dunaújváros

Publisher: Attila Kővári, 2400 Dunaújváros, Thököly Imre utca 75.

Publishing frequency: annually

<http://c&l.eu>

Applications of human-computer interfaces and related psychological-educational aspects

Gergely Sziládi

University of Dunaújváros, Táncsics M. út 1/A, Dunaújváros, 2400, Hungary, sziladig@uniduna.hu

Abstract

Computers play an important role in many areas of life and can also play an important role in supporting education and learning. Properly designed human-computer interaction can also improve the efficiency of human-computer communication. In human-computer interaction, higher-level cognitive functions play a key role in enabling people to effectively use computer tools and interpersonal forms of communication with them. This article discusses the potential, psychological, and behavioural implications of human-computer interfaces, which provide important information for the proper application of technology.

Keywords: human-computer interfaces; psychology; education;

Ember-számítógép interfészek alkalmazásai és ezek kapcsolódó pszichológiai-oktatási vonatkozásai

Sziládi Gergely

Dunaújvárosi Egyetem, Táncsics M. út 1/A, Dunaújváros, 2400, Magyarország, sziladig@uniduna.hu

Absztrakt

A számítógépek fontos szerepet játszanak az élet számos területén, és az oktatásban is fontos szerepet tudnak betölteni az oktatás-tanulás támogatásában. Az ember-számítógép interakció megfelelő kialakítása az ember-gép kommunikáció hatékonyságát is javíthatja. Az interakcióban a magasabb szintű kognitív funkciók kulcsszerepet játszanak abban, hogy az ember képessé váljon a számítógépes eszközök és a velük folytatott interperszonális kommunikációs formák hatékony használatára. A cikk az ember-számítógép interfészek lehetőségeit, pszichológiai vonatkozásait és viselkedésre gyakorolt hatását tárgyalja, mely fontos információkat nyújt a technológia megfelelő alkalmazásához.

Kulcsszavak: ember-számítógép interfészek; pszichológia; oktatás;

1. Bevezető

A tanulás hatékonysága számos tényező függvénye, azonban a modern technológiák megfelelő használata egy további tényező, mely a tanulás hatékonyságát támogathatja (K. A., 2019a). A korszerű eszközök alkalmazásával, az oktatás gördülékenységével, innovatív módszereivel ösztönözzük arra a tanulókat, hogy érdeklődésüket felkeltsük és akár önálló tanulást is ösztönözzük. A modern eszközök többek között segítséget nyújthatnak abban, hogy a ma fontosabbnak tartott gyakorlati tapasztalatokat minél hamarabb és intenzívebb módon szerezhessék meg a tanulók (Molnár, Szűts, 2019). Napjaink tendenciái azt mutatják,

hogy a modern, jól felszerelt, mind a tanárok, mind pedig a diákok szemében modernnek számító tankörnyezet szükséges a versenyképes oktatáshoz.

Ennek pszichológiai háttere sem elhanyagolható, hiszen maga a tanulás folyamata sem választható el az emberi gondolkodástól, a személyiségtől, ami közvetlenül meghatározhatja a tanulás eredményének minőségét (Balogh, 2006). Emellett figyelembe kell venni a manapság igen népszerű eszközök alapvető, emberre gyakorolt pszichológiai hatásait is, hiszen egyre többet hallhatunk internet vagy játékfüggőségről (Neils, Shavaun, 2012). Ráadásul a közösségi oldalak is nagy hatást gyakorolnak ránk az azonnaliságukkal, a világot áthidaló kapcsolattartás lehetőségével. A következmények pedig egyértelműek: a mai diákok és fiatalok szociális élete egyre inkább az internetre terelődik, sőt van, amikor már szinte az összes emberi kapcsolatukat számítógépen vagy telefonon bonyolítják (Szebedy, 2005).

A cikk az ember-számítógép interfészek (ESZI) egyes pszichológiai, és ezen keresztül kapcsolódó oktatási vonatkozásait tárgyalja.

2. Ember-számítógép interfész / interakció

Az ember-számítógép interakciók részletezése és azok pszichológiai, társadalmi és szociális hatásainak részletezése előtt szükséges annak meghatározása, hogy mit is tekintünk ember-számítógép kapcsolatnak és milyen módon zajlik ez a folyamat. Maga az interfész informatikai meghatározás olyan felületet, kapcsolódási pontot képez, ahol megvalósul a rendszer különböző komponenseinek fizikai és logikai összekapcsolása, amin keresztül ezek a komponensek kommunikálni tudnak egymással, amin keresztül a rendszer stabil működése valósul meg. Fontos kiemelni, hogy az interfész működéséhez feltételezni kell azt, hogy az interfész minden csatlakoztatott komponens számára értelmezhető adatokat továbbít.

Kommunikációs szempontból az interfészt az alábbi módon határozhatjuk meg: Olyan kapcsolódási felület és protokollgyűjtemény, amely szabályozza, hogyan csatlakozhat két számítógép, két program, hogyan használhat egy program egy berendezést, hogyan használhat a felhasználó egy programot vagy a számítógép egészét. Ez utóbbi a felhasználói interfész (GUI), mely, jelentősen befolyásolja a számítógépes feladatmegoldás betanulási idejét, munkavégzés minőségét. Manapság szinte minden, általános felhasználásra készült operációs rendszer rendelkezik grafikus felhasználói felülettel, amely segítségével a felhasználók könnyedén képesek használni az alkalmazásokat.

2.1. *Interfész kapcsolódásai formák*

Fontos kiemelni azt a tényezőt is, amely jelentősen meghatározza az interfész tulajdonságait, a számítógépes rendszer működését. Ez a környezethez való kapcsolódási formák kategóriáit a következők szerint határozom meg.

2.1.1. Technológiai megközelítés

Technológiai szempontból a kapcsolatot valamilyen kezelhető fizikai eszközön keresztül – ilyen például a billentyűzet, egér, monitor stb. – alakítja ki a rendszer a számítógép és az azt használó ember között. A lényeges momentuma a kapcsolatnak az, hogy valamely, az emberi érzékszervek számára – leggyakrabban látás, illetve hallás – értelmezhető jeleket fogadjon, illetve továbbítsa az interfész.

2.1.2. Környezeti megközelítés

Más számítógépekkel és rendszerekkel való kapcsolódás esetén az előző csoportosításhoz hasonlóan technológia a megoldás, azonban az információ továbbítás és értelmezése sokkal gyorsabb, hiszen egyszerű elektromos jelek feldolgozásával valósul meg a kommunikáció.

2.1.3. Emberrel való kapcsolatteremtés

Az emberrel való kapcsolatteremtés már a technológiai megközelítésnél is meg lett határozva, ám az ESZI-k fejlődése a régmúlttól (Licklider, 1960) napjainkig jelentős fejlődést mutatnak és sokkal közvetlenebb kommunikációt tesznek lehetővé abban az értelemben, hogy itt az ember maga információ forrás és cél is lehet. Ebben az esetben az ember a szó szoros értelemben véve nem egy fizikai interfészt – pl.: billentyűzet – manipulál, hanem egy technológiai eszköz segítségével ő maga válik a kapcsolatot kiépítő interfész részévé. Ez igaz a virtuális valóság rendszerekre is, ahol az ember mozgását, hangját szenzorokkal rögzítik, amire a rendszer az előre programozott módon reagál.

A technikai fejlesztések egyre inkább bővítik ezen interfészek és rendszerek lehetőségeit, így manapság az ember-számítógép interfészek és interakciók egyre fontosabb fejlesztési és kutatási témák alapjai. Ezen kutatások fontos lépcsők abban, hogy megérthessük saját magunk agyi folyamatait, gondolkodását és reakcióit, hiszen ezek a rendszerek a közvetlenebb kapcsolat kialakításával könnyebbé tették az ember, mint biológiai lény felépítésének és működésének vizsgálatát. A következő fejezetekben, alfejezetekben kerülnek

leírásra az ESZI-k, mint eszközök, azok emberre gyakorolt hatásai, az ezt vizsgáló tudományos vizsgálati módszerek és eredményeik.

2.2. Ember-számítógép interfészek a mindennapokban

Az ember-számítógép interfészek (Human-Computer Interfaces), válthatják és váltják fel a jelenleg használt mechanikus beviteli eszközöket – mint az egér és a billentyűzet – vagy a szöveges és grafikus interfészeket – utóbbi grafikus interfészt sokszor kombinálják ESZI eszközökkel –, ezzel együtt pedig a gépeket is felruházzák az emberi testbeszéd megértésének képességével.

2.2.1. Myo kézmozgásszenzor

Az egyik ilyen például a Myo armband (Myo, 2019) – amely az úgynevezett elektromiográfia (sEMG) (SZTE - Orvosi Fizikai és Orvosi Informatikai Intézet) vizsgálati módszer alapján működik. Ennek lényege, hogy a bőr felszínén kialakuló potenciálingadozást mérjük, amely a bőr alatti izommozgás hatására alakul ki. Ezt az ingadozást a bőrre helyezett elektródákkal lehet kimutatni (1. ábra).



1. ábra Myo kézi gesztusérzékelő egység (Myo, 2019)

A potenciálingadozások mérése lehetővé teszi különböző – az ingadozások intenzitása és pozíciója alapján következtetve az izmok mozgásában bekövetkezett – állapot változások rögzítését, majd ezekhez a jelfeldolgozó szoftver gesztusokat – a számítógép vagy a digitális eszköz számára értelmezhető jelsorozatokat – társít. Ezen gesztusok lehetnek előre definiáltak, illetve a felhasználók által beprogramozottak, függően a vezérlőszoftver funkcionalitásától.

2.2.2. Leap Motion kéz érzékelőegység

Hasonló, kézi gesztusok értelmezésére kifejlesztett eszköz a Leap Motion érzékelő egység is (Leap Motion Inc, 2019) (2. ábra).

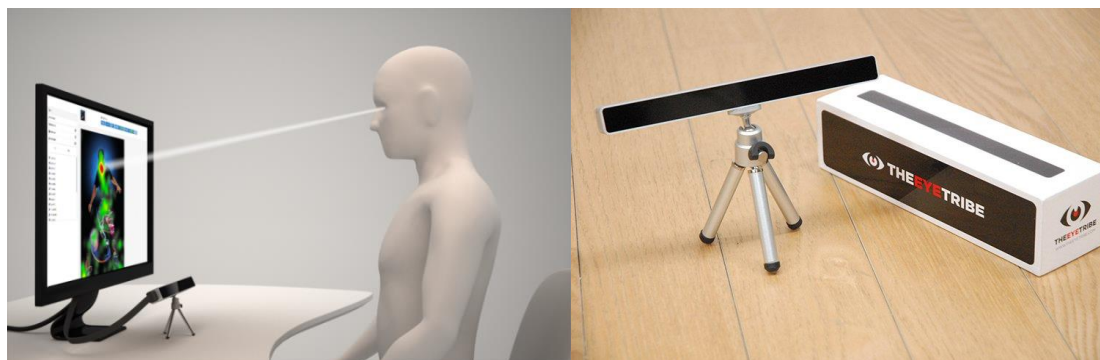


2. ábra Leap Motion kontrollert felépítése és használata (Leap Motion Inc, 2019)

A kontrollert a Leap Motion Inc. hozta létre és fejleszti 2010 óta. Az egység üzembehelyezése egyszerű: USB kábelen keresztül csatlakozik a számítógéphez vagy Mac-hez, amelyen keresztül a tápellátása és az információátvitel is történik. A kapott adatokat a cég által fejlesztett szoftver végzi a számítógépen, valós időben. Az eszköz nem csak a kézfej, hanem minden egyes ujj mozgását külön érzékelni tudja. A kéz és ujjak érzékelését az egység szélén elhelyezett monokróm infravörös kamera és három infravörös LED teszi lehetővé. Az egység érzékelési tartománya egy körülbelül 1 méter átmérőjű, nagyjából félgömb alakú térrész. A LED-ek folyamatosan infravörös fényt bocsátanak ki, két kamera pedig 300 FPS sebességgel képeket készít a LED-ek által megvilágított területről. Az elkészült képeket a vezérlő a számítógépünkre továbbítja, ahol a Leap Motion szoftvere matematikai algoritmusok segítségével feldolgozza a két kamera képét, és így a 2D-s forrásokból 3D-s, a kéz- és ujjpozíciókra vonatkozó értékeket tud kiszámítani, amely alapján a vezérlés történik.

2.2.3. Eye Tribe szemmozgáskövető szenzor

Az Eye Tribe (Eye Tribe Developer Team, 2016) szemmozgás követő egység hasonlóan a Leap Motion technológiájához infravörös LED-eket és kamerákat használ a felhasználó szemmozgásának megfigyelésére (3. ábra). A LED megvilágítja a felhasználó arcát, a kamera pedig követi a felhasználó szemmozgását amennyiben megfelelően kalibrálták az érzékelőegységet – saját tapasztalat alapján tudom, hogy a szenzor sajnos szemüveges felhasználók számára pontatlanul működik. Az eszköz 30 Hz és 60 Hz-es frekvenciájú mintavételi eljárással képes működni. Az optimális működés érdekében 45 és 75 cm-es távolságra kell ülni a szenzortól a jó megvilágítás és érzékelhetőség miatt.



3. ábra Eye Tribe szemmozgáskövető egység (Eye Tribe Developer Team, 2016)

2.2.4. Agyi tevékenység vizsgálatára alkalmas eszközök

Az agyi tevékenység számítógéppel történő vizsgálatára alkalmas eszközt agy-számítógép interfésznek – angolul Brain-computer interface (BCI) – nevezzük, amely egy olyan alternatív kommunikációs csatorna, amely támogatásával az emberi agyban végbemenő neurális aktivitás során keletkezett agyi bioelektromos jelek érzékelhetők. Ennek segítségével az ember képessé válik a külvilággal való kapcsolat kialakítására és működtetésére pusztán az agyhullámok segítségével. A BCI rendszer képes EEG jelfeldolgozás útján az agyi aktivitás megfigyelésére. A BCI rendszerek használati módja szerint megkülönböztünk invazív and noninvazív eljárásokat, attól függően milyen módon mérjük az agy elektromos aktivitását. Amennyiben az érzékelő szenzort az emberi koponyán belül az agy felszínén kerül elhelyezésre, úgy invazív BCI-ről, ha a koponyán kívül a fejbőrön helyezzük el az elektródákat a BCI rendszert a non-invazív osztályba soroljuk.

A NeuroSky által fejlesztett MindWave EEG headset egy komfortos kialakítású, a homlokon elhelyezkedő száraz típusú elektródájával az agy prefrontális kérgi tevékenységének megfigyelésére alkalmas (4. ábra). Az emberi figyelem, mely a tanulás hatékonyságában is fontos szerepet játszik, az agy prefrontális tevékenységén keresztül megfigyelhető. Az eszköz a mért és feldolgozott agyhullám információkat vezeték nélküli kommunikáción továbbítja. A Neurosky által fejlesztett ThinkGear technológiára épülő chip által meghatározott fontosabb jellemzők a figyelem vagy koncentráció értéke, a meditációs érték, pislogás és az agyhullám értékek a 0,5 - 50 Hz tartományban. Tápellátását AAA típusú akkumulátor biztosítja. Az eszköz a homloklebenyen elhelyezett érzékelő segítségével a fülhöz csatlakozó referenciaponthoz képest méri az agyjelek nagyságát. Az eszköz közvetlenül képes az agyhullámok erősségeinek és a figyelem értékének meghatározására az eszközbe épített mikrovezérlőn futó algoritmus segítségével (Katona, 2018).

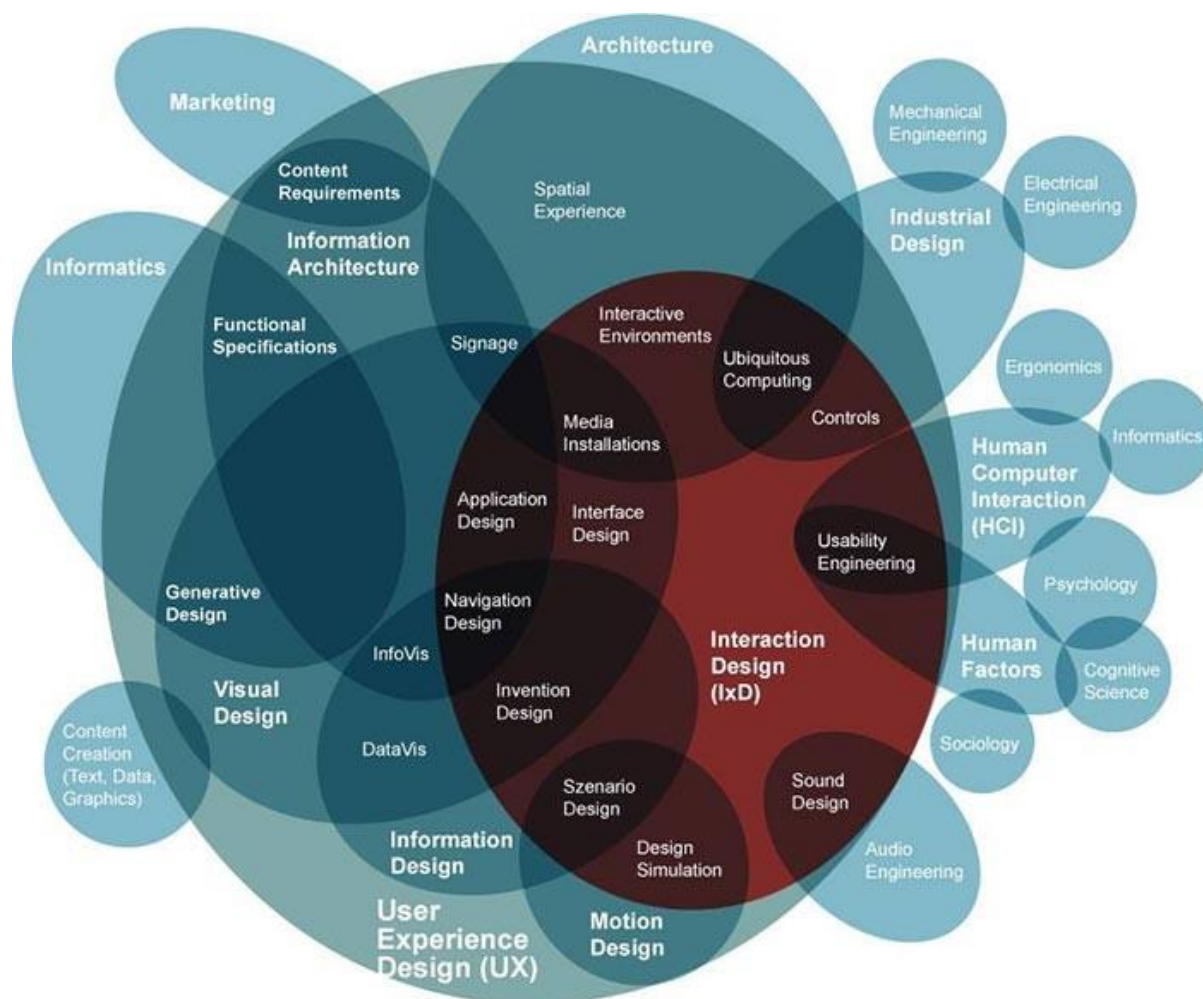


4. ábra MindWave EEG Headset (Katona, 2018)

3. Ember-számítógép kapcsolatok pszichológiai jellegzetességei

Minden ember eltérően reagál egy-egy élethelyzetre, ahogyan a számítógépekkel való kommunikáció változásaira is. Az idősebbek kevésbé igénylik a számítógéppel való interakciót a ma fiataljaival ellentétben. Azonban bármely korosztályt is vesszük alapul, az ember feladata, szerepe az ember-számítógép interakciókban alapjaiban nem változik. Minden ilyen típusú interakcióra jellemző az, hogy az emberi felhasználó egy számítógéppel és az azt magában foglaló technikai, szociális és interperszonális környezettel kerül kapcsolatba. Ez a rendszer felépítését tekintve igen összetett (5. ábra) és – bár az ember építette és őt szolgálja ki – ezzel együtt követelményeket is támaszt a felhasználókkal szemben. Ezen követelmények teljesítéséhez (kommunikációs jelek, protokoll értelmezése, használatának képessége stb.) az emberi agy pszichológiai változásokat eszközöl, melyek magukba foglalják a kognitív, azaz a megismerési, az emocionális, motivációs és az akarattal kapcsolatos folyamatokat összesítik.

A megismerés, felismerés folyamata nélkülözhetetlen az interakcióhoz, hiszen a grafikus interfészen megjelenített képet, a hangszóróból hallható hangjelzéseket vagy az egyéb visszajelzéseket, amiket a rendszer küld, tudni kell értelmezni és a későbbi interakciók miatt el is kell raktározni az agynak. Ezek a magasabb szintű kognitív funkciók kulcsszerepet játszanak abban, hogy képessé váljon az ember a számítógépes eszközök és a velük folytatott interperszonális kommunikációs formák használatára (Paolo Montuschi, 2014) (Carroll, 2017). Más aspektusát vizsgálva az ember és számítógép által alkotott rendszereknek világossá válik az is, hogy ha az ember, mint a rendszer aktív működésében kulcsszerepet játszó entitás nem képes maradéktalanul teljesíteni a rendszer által támasztott követelményeket, akkor az egész működés – a hibátűrő berendezések számától és minőségétől függően – bizonytalan és hibás lehet.



5. ábra Ember-számítógép interakcióhoz és interfészekhez kapcsolódó területek (Carroll, 2017)

3.1. A technológia és az ember-számítógép interfészek hatása a viselkedésre

Érdekes változás, amelyet az ESZI-k használata okoz, jól megfigyelhető az emberek viselkedésében. Ha korunk technológiai vívmányainak csak egy elemét, például az internetet emeljük ki, már ott is szembetűnők az eltérések az átlagos viselkedéshez képest.

Az internet az 1960-as évektől fejlesztett, eleinte katonai, majd később polgári felhasználású globális kommunikációs csatorna, amely a világon létező számítógépes hálózatokat tömöríti magában, ezzel lehetővé téve akár a valós idejű kapcsolattartást a bolygó szinte bármely két pontja között (Comer, 2006). A 70-es és 80-as években csak katonai, akadémiai és kutatóintézetek férhettek hozzá az eleinte misztikusnak tűnő globális hálózathoz, amely mára a Föld lakosságának nagy része által elérhető. Hatalmas mennyiségű információ áramlik keresztül rajta nap, mint nap (Cisco, 2017), amely segítségével szinte bármilyen emberi

tevékenység támogatásához szükséges adat beszerezhető és felhasználható az adott feladat minél gyorsabb és sikeresebb kivitelezéséhez. Emellett a kapcsolattartáshoz is kiemelkedően fontos, hiszen manapság a katonai, kormányzati, valamint orvosi és kutató intézetek közötti azonnali információ áramlás és kommunikáció sok esetben létfontosságú lehet. Ennek az azonnaliságnak, nagyfokú rendelkezésre állásnak és legfőképp az egyre kényelmesebb használatnak köszönhetően az felhasználóknál kialakulnak bizonyos szokások, amelyek sok esetben negatív tényezőként jelenhetnek meg a mindennapi életben.

Az internet használatakor sokszor furcsa viselkedési formákat használ az ember. Vannak pozitívumai az internetes kommunikációnak – például az azonnaliság, a földrajzi távolságtól való függetlenség –, ugyanakkor sok negatívummal is szembesülhetünk „online életünk” során. Ez lehet egy zaklatottabb lelkiállapotban írt vélemény egy közösségi oldalon vagy egy félresikerült fotó egy szórakozóhelyről. Legyen azonban bármely lehetőség is, egy dolgot nem szabad elfelejtenünk – amit sajnos sokan mégis megtesznek: az internet az emberekkel ellentétben sosem felejt. Bármilyen információt is küldünk el vagy töltünk fel, azt valamely adatközpontban mentik le, amelyet évekkel, évtizedekkel később is elő lehet keresni.

Az online, digitális világ és az ahhoz kapcsolódó eszközök használatakor sokszor úgy tűnhet, hogy az ember irányít, ám ez csak féligazság. A pszichológiai kutatások már évtizedekkel korábban rámutattak már az emberi viselkedés szélsőségeire. Gyakran a legjózanabb ember is meggondolatlanul cselekszik, ha bizonyos fokú befolyást szerez valami vagy valaki felett, amelyet korábban nem gyakorolt. Akik nem használják ezeket a technológiákat, azok közül sokan negatív véleményt alkotnak. Ez igaz az idősebb generációra is, valamint azokra is, akik nem értik teljes mértékben a fent említett technológiákat. Természetesen, mint minden ember által elkészített eszköznek vagy kitalált szellemi terméknek vannak hibái, mivel mi magunk sem vagyunk tökéletesek: naponta olvashatunk különböző balesetekről, bűncselekményekről, amelyeknél szerepet játszott a világháló vagy az okoseszközök. Sokan a technológiai fejlődést is okolják, pedig nem a gép, hanem az ember az, aki döntést hoz. A számítógép csak egy eszköze ennek a végrehajtásához.

Bár a számítógépek és az internet kapcsán végzett pszichológiai kutatások még nem mindennaposak, ugyanakkor számos pszichológiai változás és viselkedésminta leírása már rendelkezésre áll a korábbi évtizedekből, amelyet alkalmazhatnak. Ezért nem kell teljesen új fogalmakat alkotni, mindössze új vizsgálatkora és megfigyelésekre van szükség. Más, az ember által megszokott társadalmi helyzet – társalgás közbeni hirtelen agresszió;ingerszegény környezetben történő elszigetelődés stb. – vizsgálata alapján az új, digitális

környezetben előforduló jelenségekkel párhuzam vonható. Ezek segítségével stabil alapokon indulhat el a kutatás, hogy megérthessük, milyen hatással vannak ránk korunk technikai vívmányai. (Centola, 2010) (Lo, Hsieh, & Chiu, 2013)

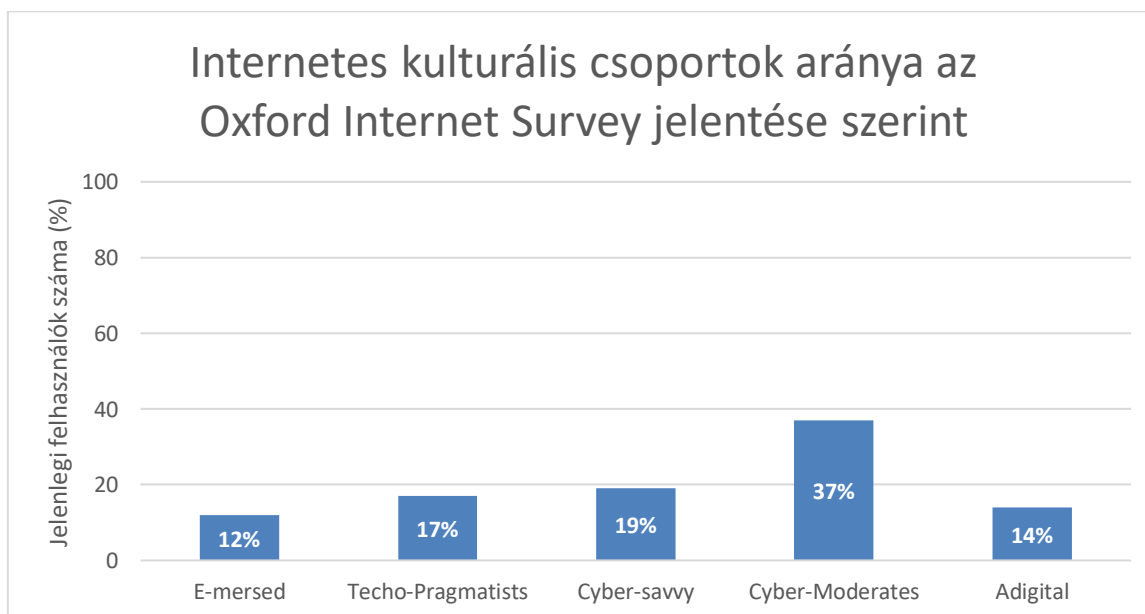
3.1.1. Alternatív személyiségek

A ma digitális eszközei alig két évtized alatt meghódította a világot. A 90-es évek végén az internet lassan terjedt és a számítógépek is sokszor csak az oktatási intézmények és kutatók számára volt megfizethető. Éppen ezért sokáig privilégiumnak számított az, ha valaki emailt küldhetett vagy éppen egy adatbázisban kereshetett információt. Abban az időben mindez csak egy új, kevesek számára elérhető lehetőséget jelentett. Ekkor még a legtöbben személyesen találkoztak és kapcsolódtak ki egy-egy hosszú nap után. Mára ez jelentősen megváltozott.

Manapság az arctalanság, az online világ fátyla elhomályosítja a valóságot, amit digitális létünk során tapasztalunk. Egyre több kutatás foglalkozik az ESZI-k a személyiségre és a viselkedésre vetített váltoásaival kapcsolatban (Rahmani & Lavasani, 2011). Az Oxfordi Egyetemen egy külön intézet foglalkozik az internetes kutatásokkal és megfigyelésekkel. Több jelentést is készítettek az elmúlt években (William H. Dutton G. B., 2011) (William H. Dutton G. B., 2013), amelyek világosan körvonalazzák, miként kapcsolódik az ember a digitális világhoz és az hogyan változtatja meg mindennapi életünket. Az intézet jelentésében az írók öt csoportot, szubkultúrát különböztetnek meg a britek társadalmában (6. ábra):

- E-mersed: Ezt a csoportot az abszolút elfogadás jellemzi az infokommunikációs eszközök irányában, sőt, igénylik is azokat a mindennapokban. Boldogok, ha online lehetnek, hogy elérhetik őket és hogy ők is elérhetnek bárkit vagy bármilyen információt, amellyel az emberiség rendelkezik. Ők a szabadságot, az általuk manipulálható világot látják az internetes technológiákban, ahol szabadon cselekedhetnek, ahol egy szabad nemzetközi közösséghez tartozhatnak. A jelentés a brit lakosság 12%-át sorolja ebbe a kategóriába.
- Techno-pragmatists : Hasonlóan befogadók az új technológiákkal szemben, mint az E-mersed tagjai, de máshogy tekintenek az online világra. Ők azok, akik a munkájuk, az életük megkönnyítésére használják fel a digitális technológia vívmányait. Nem tekintik ezeket menedéknek és nem jellemző rájuk, hogy kizárólag szórakozásra céljából használják infokommunikációs eszközöket. A statisztika szerint a brit internetes felhasználók 17%-a képezi ennek a csoportnak a tagjait.

- Cyber-savvy : Ez a csoport összetett képet mutat az internet használatával kapcsolatban. Egyrészt szeretik kihasználni a gyors információ keresés és közlés adta lehetőségeket, emellett a szórakozáshoz és kapcsolattartáshoz is használják az online világot. Másrészt sokszor úgy érzik, hogy az internet, a digitális technológiák részben irányítják az életüket és sok esetben időpocsékolásként élik meg ezek használatát. Körültekintők az internet használata terén, ugyanakkor nem adják fel a használatával járó kényelmet.
- Cyber-moderates : A szokásaikban legkönnyebben leírható csoportot alkotják ők, részben az E-mersed tagjaihoz hasonlóan jó időtöltésnek, hatékony információ forrásnak tartják a világhálót, ugyanakkor mellőzik a Cyber-savvy-k félelmeit. A jelentés őket tekinti az átlagos felhasználóknak, akik a brit internetfelhasználók 37%-át teszik ki.
- Adigitals: Az előzőekben felsorolt csoportok közül ők a legszeptikusabbak. Számukra az internet mások által irányított média, nem találják hatékonynak a használatát. Leginkább ez a csoport érzékeny a negatív kritikákra az infokommunikációs technológiákkal kapcsolatban. Gyakran érzik magukat kirekesztettnek, ha nehezen kezelhető kihívással találják szemben magukat az internetes technológiák használatakor. A csoport a brit felhasználók 14 százalékát teszi ki.



6. ábra - Internetes felhasználói csoportok Angliában (William H. Dutton G. B., 2013)

A 2013-as jelentés ismerteti a különböző csoportokra vonatkozó életkori, élethelyzeti és szociális körülményeket, amelyek a saját tapasztalataim és a kor tendenciáinak megfelelően azt mutatják, hogy a fiatalabb korosztálynak rendszeresen szüksége van az online világra. Ők jellemzően az E-mersed, a Techno-Pragmatists és a Cyber-savvy csoport tagjai, míg az életkor előrehaladtával egyre több jellemzőt és viselkedési formát találunk, amelyek a Cyber-moderates, majd később az Adigital kategóriákra illik. Bár kapcsolódó kutatás és felmérés az angliai lakosságra vonatkozik, általánosságban ugyan ezek a jelenségek figyelhetők meg Magyarországon is (William H. Dutton G. B., 2011) (William H. Dutton G. B., 2013). Ezt a változást az oktatási rendszerre is terhet ró, amit rövid időn belül orvosolni kell.

4. Összefoglalás

A korszerű eszközök alkalmazásával a tanulás hatékonysága is növelhető, a jövőben az ember-számítógép interfészeknek is fontos szerepük lehet ebben. A modern eszközök többek között segítséget nyújthatnak abban, hogy a ma fontosabbnak tartott gyakorlati tapasztalatokat minél hamarabb és intenzívebb módon szerezhessék meg a tanulók.

Azonban az ember-számítógép interfészek előnyeinek kihasználása szempontjából tudni érdemes a mögöttes pszichológiai folyamatokat, hiszen maga a tanulás folyamata sem választható el az emberi gondolkodástól, a személyiségtől, ami közvetlenül meghatározhatja a tanulás eredményének minőségét. A cikk ezen szempontokat, az ember-számítógép interfészek (ESZI) egyes pszichológiai, és ezen keresztül kapcsolódó oktatási vonatkozásait tárgyalta.

A megismerés, felismerés folyamata nélkülözhetetlen az interakcióhoz, hiszen a grafikus interfészen megjelenített képet, a hangszóróból hallható hangjelzéseket vagy az egyéb visszajelzéseket, amiket a rendszer küld, tudni kell értelmezni és a későbbi interakciók miatt el is kell raktározni az agynak. Ezek a magasabb szintű kognitív funkciók kulcsszerepet játszanak abban, hogy képessé váljon az ember a számítógépes eszközök és a velük folytatott interperszonális kommunikációs formák használatára. Másrészt az ember-számítógép interfészekon keresztül az ember, mint a rendszer aktív működésében kulcsszerepet játszó entitás nem képes maradéktalanul teljesíteni a rendszer által támasztott követelményeket, akkor az egész működés – a hibatűrő berendezések számától és minőségétől függően – bizonytalan és hibás lehet. Ezért is fontos ismerni a technológia pszichológiai hátterét és ennek megfelelően kell meghatározni a technológia helyes alkalmazását.

Irodalomjegyzék

- Balogh L. et al (2006). *Pedagógiai pszichológia az iskolai gyakorlatban*. Urbis Könyvkiadó.
- Carroll, J. M. (2017). *The Encyclopedia of Human-Computer Interaction*, 2nd Ed.
- Cisco. (2017). *The Zettabyte Era: Trends and Analysis*. Forrás: Cisco: <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.html>
- Centola, D. (2010). The spread of behavior in an online social network experiment. *Science*, 329(5996), 1194-1197.
- Clark Neils, Scott Shavaun (2012). A játékfüggőség jelenségéről. *alkalmazott pszichológia* 2012(2), 91–95.
- Comer, D. (2006). *The Internet book*. Prentice Hall.
- Eye Tribe Developer Team. (2019). *Eye Tribe Developer Website*. Letöltés dátuma: 2019. 11 23, Forrás: <http://aws-website-theeyetribe-lbmoo.s3-website-us-east-1.amazonaws.com/dev.theeyetribe.com/dev.theeyetribe.com/general/index.html>
- Katona J. et al (2018). Electroencephalogram-Based Brain-Computer Interface for Internet of Robotic Things. In *Cognitive Infocommunications, Theory and Applications*, Springer. 253–275.
- Kővári, A. (2019a). A felnőttoktatás 4.0 és az az ipar 4.0 kihívásai az életen át tartó tanulásban. *PEDACTA*, 9(1), 9–16.
- Kővári A. (2019b). *Human-Computer Interfaces I*. Subotica: Subotica Tech.
- Kővári, A. (2019b). *Ember-számítógép interfészek I*. Subotica: Subotica Tech.
- Leap Motion Inc. (2019). *Leap Motion technology* . Forrás: Leap Motion Website: <https://www.leapmotion.com/>
- Licklider, J. C. (1960). Man-Computer Symbiosis. *IRE Transactions on Human Factors in Electronics*, HFE-1, 4–11.
- Lo, S.-K., Hsieh, A.-Y., & Chiu, Y.-P. (2013). Contradictory deceptive behavior in online dating. *Computers in Human Behavior*, 29(4), 1755-1762.
- Molnár, G., Szűts, Z. (2019). Modern ikt és hálózatalapú tanulástámogatási lehetőségek a nonformális és informális tanulási folyamatban. *LÉTÜNK*, 2019(1.), 189–198.

Myo. (2019). Myo Armband introduction. Letöltés dátuma: 2019. 11 23, forrás: myo.com
Website: www.myo.com

Paolo Montuschi, A. S. (2014). Human-Computer Interaction: Present and Future Trends.

Forrás: Computing Now:

<https://www.computer.org/web/computingnow/archive/september2014>

Rahmani, S., & Lavasani, M. G. (2011). The relationship between internet dependency with sensation seeking and personality. *Procedia - Social and Behavioral Sciences*, 30, 272-277.

Szebedy, T. (2005). A pedagóguspálya sajátosságai és a foglalkozási ártalmak. *Új Pedagógiai Szemle*, 18-32.

William H. Dutton, G. B. (2011). *Oxford Internet Surveys Reports - Report 11*.

William H. Dutton, G. B. (2013). *Oxford Internet Survey Reports - Report 13*.

Advantages of Computer Network Simulation Environment in Education

László Gereben

Ózdi SzC Bródy Imre Szakgimnáziuma, Szakközépiskolája és Kollégiuma, Petőfi út 20., Ózd, 3600, Hungary, lacos4a@gmail.com

Abstract

Computer applications are also increasing the potential for education. The Packet Tracer computer network simulation software provides several easy-to-use features for debugging and testing, helping both educators and students. This article presents the results of a survey about Packet Tracer simulator software that can be used to teach the topic of computer networks. The results of the survey show that the use of the simulation environment simplifies the solution of the computer network problems, the possibilities provided by the simulation software allow the visual learning of students, and the simulation environment facilitates the teacher's work.

Keywords: computer network knowledge; simulation; survey;

Számítógép hálózati szimulációs környezet előnyei az oktatásban

Gereben László

Ózdi SzC Bródy Imre Szakgimnáziuma, Szakközépiskolája és Kollégiuma, Petőfi út 20., Ózd, 3600, Magyarország, lacos4a@gmail.com

Absztrakt

A számítógépes alkalmazások egyre több lehetőséget adnak az oktatás terén is. A Packet Tracer számítógép hálózati szimulációs szoftver a hibakeresés, tesztelés szempontjából is jól használható lehetőségeket ad, ezzel segítve mind az oktató, mind pedig a tanulók munkáját. A cikk a számítógép hálózatok témakörének oktatása szempontjából előnyösen használható Packet Tracer szimulátorral összefüggésben végzett felmérés eredményeit mutatja be. A felmérés eredményei azt mutatják, hogy a szimulációs környezet alkalmazása leegyszerűsíti a feladatmegoldást, a szimulációs szoftver adta lehetőségek a tanultak vizuális rögzítésére adnak lehetőséget, valamint a szimulációs környezet megkönnyíti a tanári munkát.

Kulcsszavak: számítógép hálózati ismeretek; szimuláció; kérdőíves felmérés;

1. Bevezető

A számítógépes alkalmazások egyre több lehetőséget adnak az oktatás terén is, a cikk is egy a számítógép hálózatok témakörének oktatása szempontjából előnyösen használható szimulátorral összefüggésben végzett felmérés eredményeit mutatja be. A szimuláció nem csak a tervezés, hanem a tesztelés és hibakeresés szempontjából is nagyon jó lehetőségeket ad, mely nagyban segíti a diákok problémamegoldó gondolkodását, és a hibakeresésben a

megértést, értelmezést. A hálózati ismeretek elméleti oktatásához rendelkezésre áll néhány, interneten elérhető jegyzet, vagy a Cisco CCNA (Cisco Certified Network Associate, a Cisco által kiadott IT tanúsítvány) szakmai agyaga, azonban a gyakorlati oktatás során kénytelenek vagyunk saját módszertant kifejleszteni, azt alkalmazni a gyakorlatban (Lammle, 2011). A Packet Tracer számítógép hálózati szimulációs szoftver a hibakeresés, tesztelés szempontjából is jól használható lehetőségeket ad, ezzel segítve mind az oktató, mind pedig a tanulók munkáját.

A számítógép hálózatok téma szerves részét képezi az informatikus szakemberek szakmai ismereteinek (Horváth, 2014). A szakmai érettségi vizsga komplex gyakorlati és szóbeli vizsgatevékenységén egyaránt előtérbe kerül (OH, 2018).

A cikk a Hálózati ismeretek témakörben alkalmazott szimulációs környezet vonatkozásában végzett kérdőíves felmérésről és annak eredményeit mutatja be.

2. A kutatás célja, módszere, eszköze

Általánosságban elmondható, hogy kérdőíves vizsgálattal rövid idő nagy adatmennyiség gyűjthető be. Ugyanakkor a kérdőívek hitelessége gyakorta erősen kétséges lehet, valamint a kérdések megfogalmazása, a kérdések sorrendje is befolyásolhatja a kapott válaszokat. A névtelenség és az önálló kitöltés lehetősége az őszinte válaszokat valószínűsíti. Azonban a megkérdezettek legjobb szándéka ellenére is, egyes válaszok eltérhetnek a valóságtól.

2.1. A kutatás célja

A Hálózati ismeretek gyakorlatát csupán harmadik éve tanítom. Az első tanévben került bevezetésre az ágazati szakmai érettségi, amely bizonyos mértékben megelőzte az előkészítő tevékenységet, ugyanis az első gyakorlati mintafeladat a Hálózati ismeretek témakörből félév előtt jelent meg. A kezdeti nehézségek ellenére úgy gondolom, hogy kellő rutint szereztem a téma oktatásában, és elérkezettnek látom az időt, hogy a szimulációs környezet használóit megkérdezzük annak hatékonyságáról, szükségességéről.

A szakdolgozatomban végzett felmérés célja, hogy megismerjük a szimulációs környezet felhasználóinak véleményét, hogy a szimulációs módszer alkalmazásának a gyakorlati oktatás során mennyire nyújt támogatást a szükséges szakmai ismeretek elsajátításában.

2.2. *A kutatás módszere*

A Hálózati ismeretek gyakorlati oktatásában a Packet Tracer nyújtotta szimulációs környezetet használjuk iskolánkban. A középszintű ágazati érettségi gyakorlati feladatsorának Hálózati ismeretek című része is ezen a környezet használatára került összeállításra. (A szimulációs környezet alkalmazásának módszertanáról az előző fejezetben esett szó részletesen.) Ebben a fejezetben arra vagyunk kíváncsiak, hogy a szimulációs szoftver alkalmazása mennyire eredményes a gyakorlatban, mennyire támasztja alá, illetve segíti az elméletben megismertek megértését, rögzítését, és milyen mértékben segíti a tanári munkát.

A felmérésben két kérdőív elemzését tűztem ki célul: az egyiket olyan tanulók töltötték ki, akik valamilyen tapasztalattal rendelkeznek a szimulációs környezet alkalmazásával kapcsolatban, a másikat azok a tanár kollégák, akik alkalmazzák a szimuláció adta lehetőségeket és a módszert a téma gyakorlati oktatásában.

2.3. *A kutatás eszköze*

A kérdőívek összeállításánál elsősorban a zárt kérdéseket részesítettem előnyben, ugyanis azoknak a kiértékelése könnyebb, szemléletes diagramokat készíthetünk, másrészt nem ad lehetőséget a válaszadóknak nem oda illő vagy értékelhetetlen válaszok (nemválasz) megadására. Mindkét (tanulói és tanári) kérdőívben az utolsó kérdés nyitott kérdés, amely az egyik esetben a témakörön belül a nehézséget jelentő ismeretekre, a másik esetben a szimulációs környezettel kapcsolatban kíváncsózó további fejlesztési javaslatokra vonatkozik.

A tanulói kérdőív húsz kérdésből áll, amelyből az első nyolc személyes jellegű, (életkor, nem, lakhely, stb) a többi pedig a szimulációs program használatával kapcsolatos. A személyes kérdéseknél az utolsó a tanuló szorgalmával kapcsolatos egytől ötig terjedő skálán. A szimulációs módszerre vonatkozó kérdések – az utolsó nyitott kérdés kivételével – szintén ötös skálán történő értékelést tesznek lehetővé.

A tizenöt kérdést tartalmazó tanári kérdőív első öt kérdése szintén személyes jellegű (lakhely, intézmény, szakmai tapasztalat, stb), a többi a szimulációs környezetre vonatkozik. A szimulációs módszerrel kapcsolatos kérdések – az utolsó nyitott kérdés kivételével – ötös skálán jelzi a kolléga véleményét az adott kérdéskörben.

A kérdőíveket a válaszadók online töltik ki, azokat az <http://online-kerdoiv.com> teljesen automatizál kérdőívkészítő rendszerrel készítettem. Az online kérdőív előnyei, hogy olcsó, könnyen elérhető és megosztható, egyszerre többen is tölthetik, könnyen kiértékelhető, gyors,

és azonnal látjuk az eredményt. Hátrányai a szociokulturális háttér, valamint a technikai tényezők befolyásoló hatása, a nyitott kérdések kiértékelése nehezebb, és az értékelhetetlen válaszok.

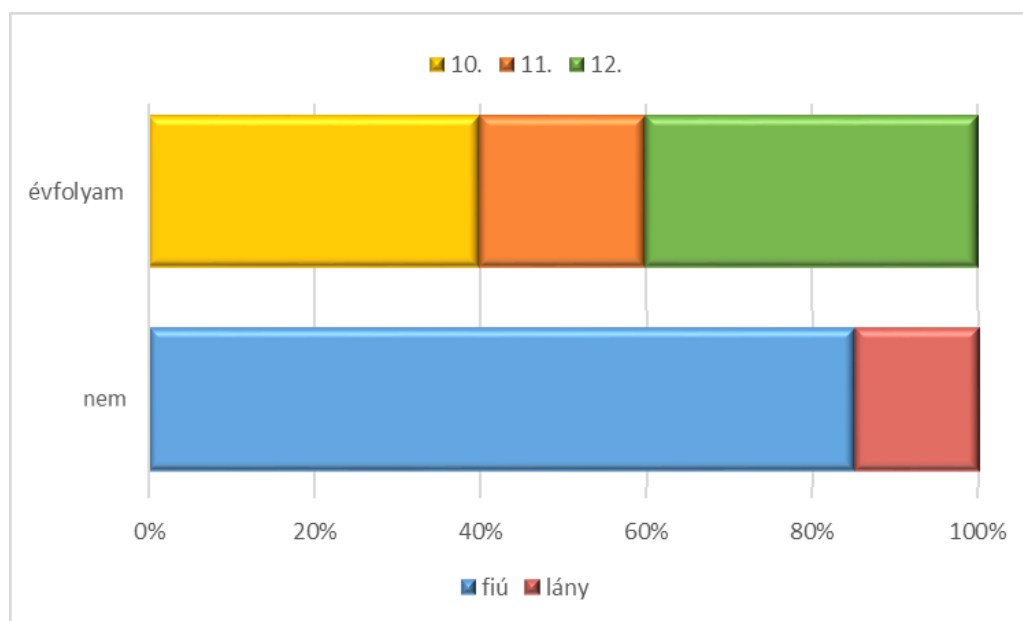
3. Hipotézisek

3.1. A kérdőíves felméréssel a szimuláció a hálózati ismertek gyakorlatában történő alkalmazása terén az alábbiak szerint kerültek megfogalmazásra:

- a) eredményesen leegyszerűsíti a feladatmegoldást,
- b) hatékonyan segíti az elméletben tanultak megértését,
- c) a tanári munkát nagymértékben megkönnyíti.

4. Kutatás eredményei

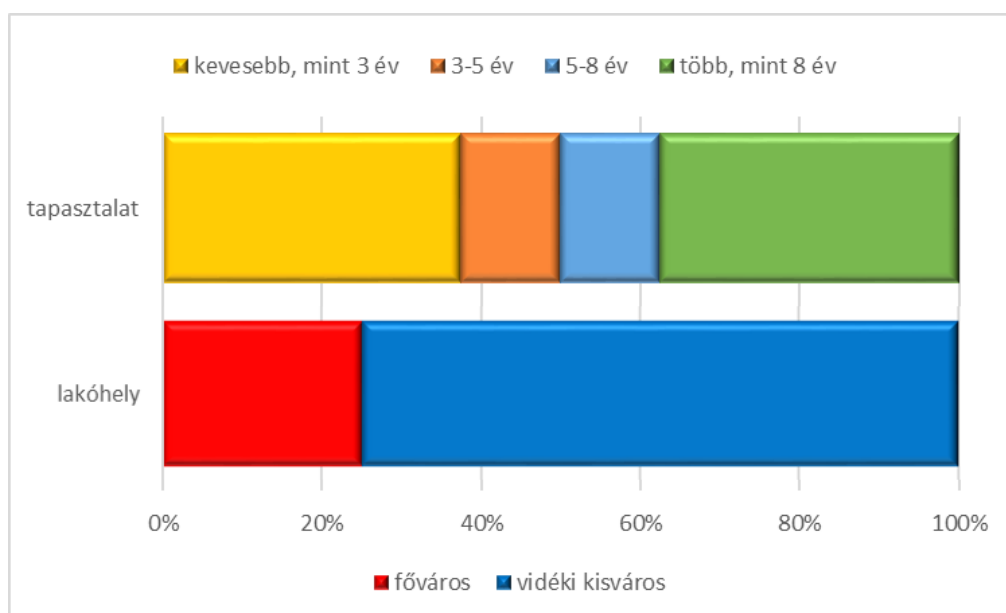
Ahogy azt már korábban említésre került, kétféle kérdőív került használatra a felmérés során. Az egyik kérdőív célcsoportja azok a tanulók, akiknek a szimulációs környezet alkalmazásával tanítom a hálózatok gyakorlatát. A minta tagjai 10-12. évfolyamos, nappali tagozatos szagimnáziumi tanulók. 10. évfolyamon 16 fő (40%), 11. évfolyamon 8 fő (20%), 12. évfolyamon 16 fő (40%). Ez összesen 40 fő, amiből 6 fő leány (15%). A tanulói minta évfolyam- és nemeloszlását az 1. ábra mutatja (kérdőív 1-8 kérdései).



1. ábra A megkérdezett tanulók évfolyam- és nem szerinti eloszlása

A lakóhelyet megvizsgálva a tanulók döntő többsége ózdi, csupán 15%-uk jár be a környező településekről. Ha az előző tanév végi tanulmányi átlagot tekintjük, azt mondhatjuk, hogy 65% közepes tanuló, 20% jó, 5% jeles, és sajnos 10%-a a mintának nem érte el a 2,5-es átlagot. Ezt az adat is alátámasztja, miszerint a tanulók több mint 70%-a csak napi fél vagy 1 órát, esetleg annál kevesebbet tölt otthon intenzív tanulással, noha 90%-uk legalább közepes szorgalommal jellemezte önmagát.

A tanári kérdőívet sajnos sokkal kisebb célcsoportra sikerült alkalmazni, összesen 8 fő vett részt a felmérésben. Közülük 2 fő (25%) a fővárosban él, a többi 6 fő (75%) vidéki kisváros lakosa. A 8 megkérdezett pedagógusból 3 fő kevesebb, mint 3 éve tanítja a hálózati szimulációt, 1 fő 3-5 éve, 1 fő 5-8 éve és 3 fő több mint 8 éve. A tanári minta tapasztalat és lakóhely szerinti elosztását a 2. ábrán látható.



2. ábra A megkérdezett tanárok tapasztalat és lakóhely szerinti eloszlása

A tanár minta nagyrésze középfokú oktatási intézményben tanít, a megkérdezettek közül csupán egy fő tanít ipari környezetben. A felmérésben részt vevő tanárok 62,5%-a használja a szimulációs programot a tanórák 30-50%-ában, a többiek esetében ez csak az idő 10-30%-át teszi ki. A gyakorlati feladatok előkészítésre 62,5% szán legfeljebb napi egy órát, mások 1-2 órát szánnak előkészületre napi szinten.

A tanulói kérdőív esetén a 9-19. kérdésekre válaszolhattak a megkérdezettek ötfokozatú skálán. Kérdések az alábbiak voltak:

9. Mennyire gondolod fontosnak a hálózati ismeretek elsajátítását a jövőbeli szakmai munkáddal összefüggésben? (1-5)
10. Mennyire támogatja az eszközök kiválasztása a hálózati készülékekről tanultak megértését? (1-5)
11. Mennyire támogatja a kapcsolatok kialakítása az átviteli közegekről tanultak megértését? (1-5)
12. Mennyire támogatja az IP címek, alhálózati maszkok megadása a témáról tanultak megértését? (1-5)
13. Mennyire okoz nehézséget a CLI felület használata, a parancsok megtanulása? (1-5)
14. Mennyire támogatja a forgalomirányítás beállításával összefüggő gyakorlati feladat a témáról tanultak megértését? (1-5)
15. Mennyire támogatja a vezeték nélküli hálózat konfigurálásával összefüggő gyakorlati feladat a vezeték nélküli hálózatok működésének megértését? (1-5)
16. Mennyire könnyíti meg a szimulációs mód a hibakeresést? (1-5)
17. Összességében milyen mértékben segíti a hálózatok tervezésének, konfigurálásának elsajátítását a szimulációs szoftver? (1-5)
18. Mit gondolsz, milyen mértékben segíti egy szimulációs környezet az önálló tanulást, a hálózati ismeretek alkalmazásával összefüggő önálló gyakorlást? (1-5)
19. Mennyire jellemző, hogy a hálózati ismeretek témakörben egyéb online elérhető forrásokat (például: weboldalak, youtube oktató- vagy bemutató videó) is felhasználasz a tanulás során? (1-5)

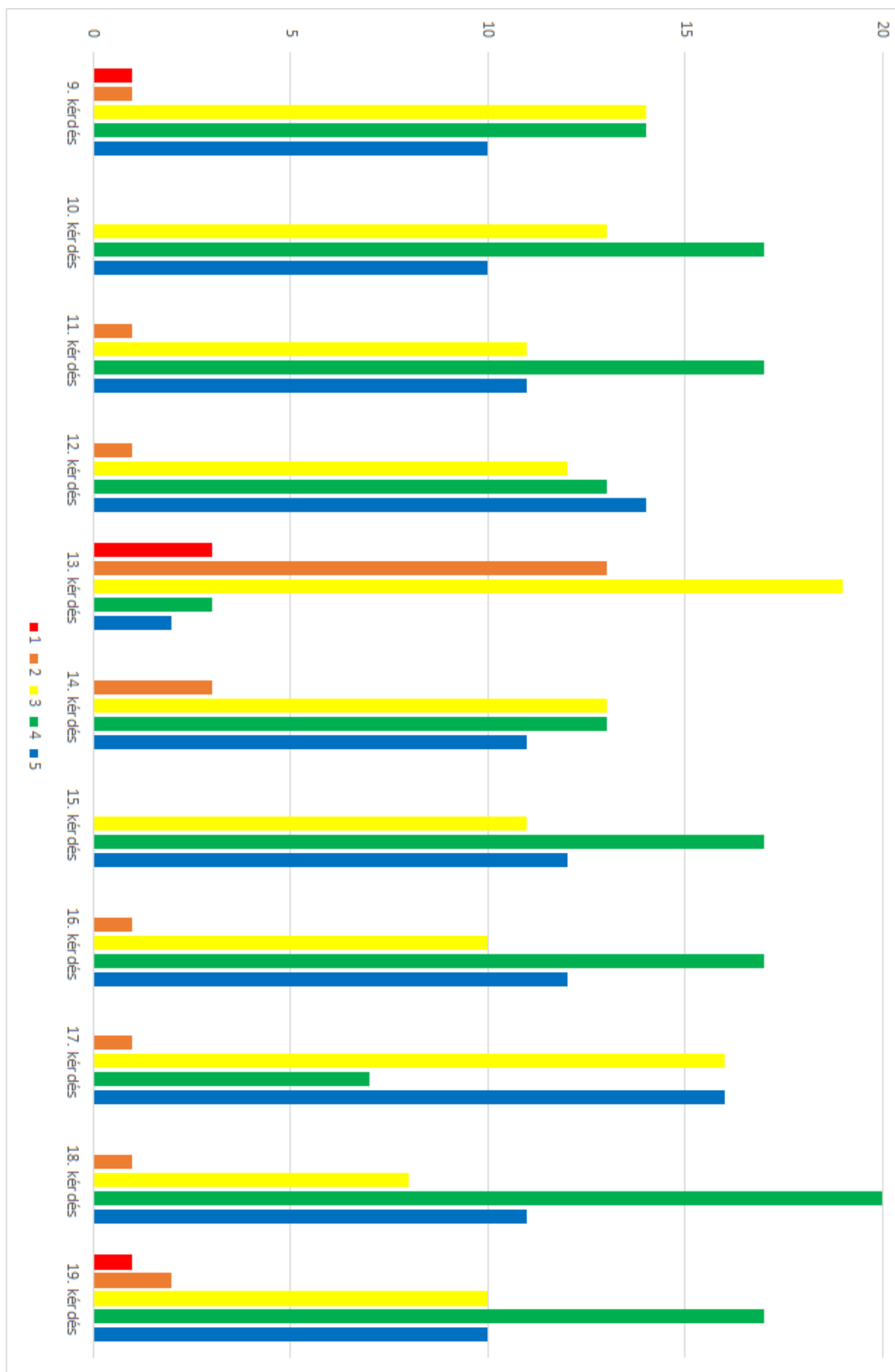
A kérdésekre adott válaszok alapján készítettem egy táblázatot, amelynek adataiból készült a 3. ábra. Az ábrát tanulmányozva megállapítható, hogy a kérdések többsége esetében a válaszok a 3-4-5 felé szórnak, 1-2 alig akad. Az utolsó kérdésben a további gyakorlásra, magyarázatra szoruló területekre kérdeztem rá. Ezek százalékos előfordulásuk sorrendjében az IP címek, a CLI felület és a DHCP, amely szintén a CLI-hez kapcsolódik, de szorosan összefügg az IP címekkel is.

A tanári kérdések közül a 6-14. kérdésekre lehetett válaszolni ötfokozatú skálán (Az első öt kérdést az előző pontban tárgyaltuk):

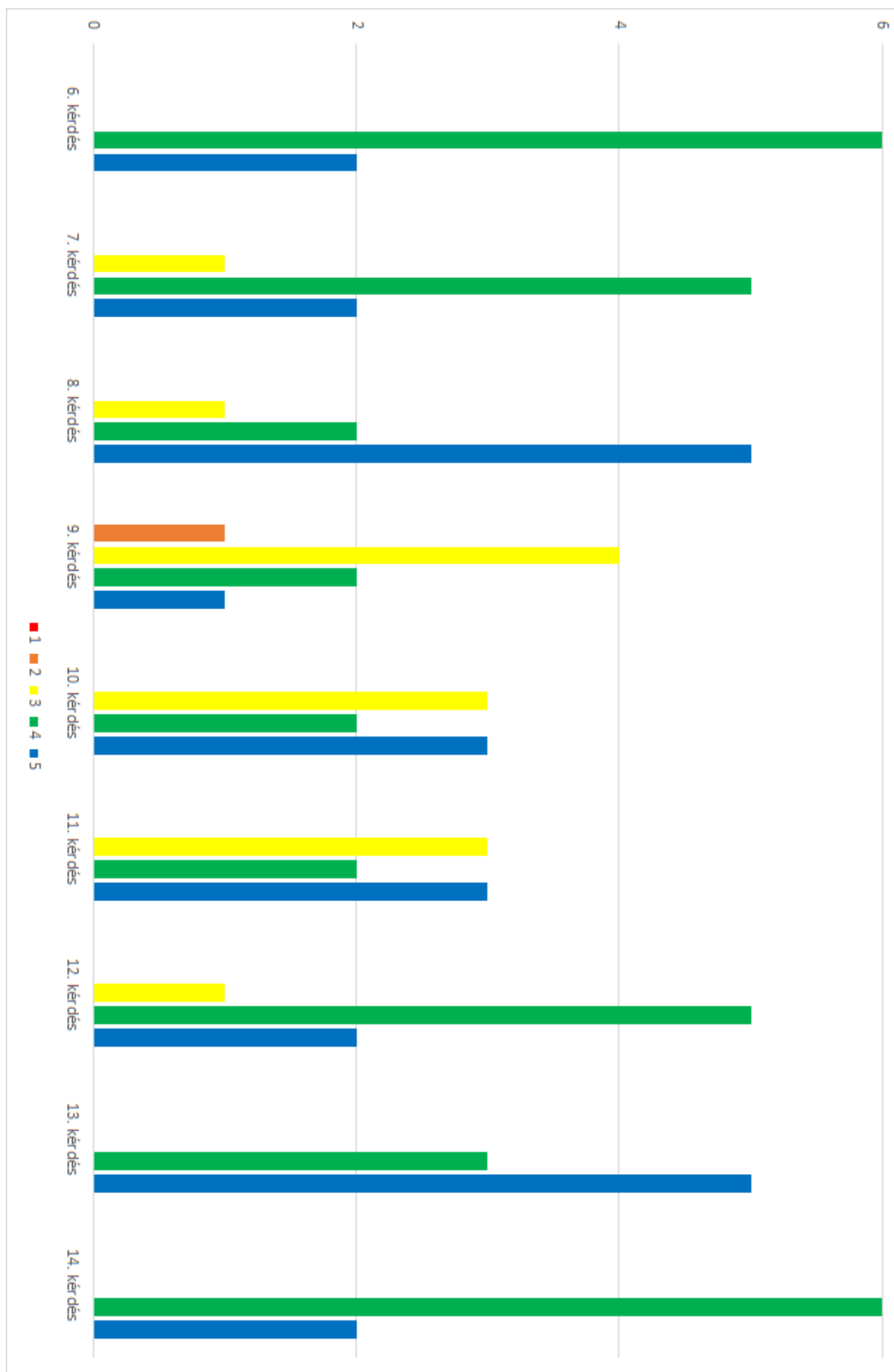
6. Véleménye szerint mennyire támogatja az eszközök kiválasztása a hálózati készülékekről tanultak megértését? (1-5)
7. Véleménye szerint mennyire támogatja a kapcsolatok kialakítása az átviteli közegekről tanultak megértését? (1-5)
8. Véleménye szerint mennyire támogatja az IP címek, alhálózati maszkok megadása a témáról tanultak megértését? (1-5)
9. Ön szerint mennyire okoz nehézséget a tanulóknak a CLI felület használata, a parancsok megtanulása? (1-5)
10. Ön szerint mennyire támogatja a forgalomirányítás beállításával összefüggő gyakorlati feladat a témáról tanultak megértését? (1-5)
11. Ön szerint mennyire támogatja a vezeték nélküli hálózat konfigurálásával összefüggő gyakorlati feladat a vezeték nélküli hálózatok működésének megértését? (1-5)
12. Véleménye szerint mennyire könnyíti meg a szimulációs mód a hibakeresést? (1-5)
13. Összességében milyen mértékben segíti tanári munkáját a szimulációs szoftver? (1-5)
14. Mennyire jellemző, hogy a hálózati ismeretek témakörben egyéb online elérhető forrásokat (például: weboldalak, youtube oktató- vagy bemutató videó) is felhasznál a tanítás során? (1-5)

Az eredmények a 4. ábrán láthatók.

Az utolsó kérdés egy nyitott kérdés, amely a szimulációs program fejlesztési javaslataira vonatkozik. Ebben a vonatkozásban a jobb grafikus megjelenítés, szintaktikai hibák javítása és a valós hibák szimulálása került előtérbe.



3. ábra A tanulói kérdőív kérdéseire adott válaszok gyakoriságai

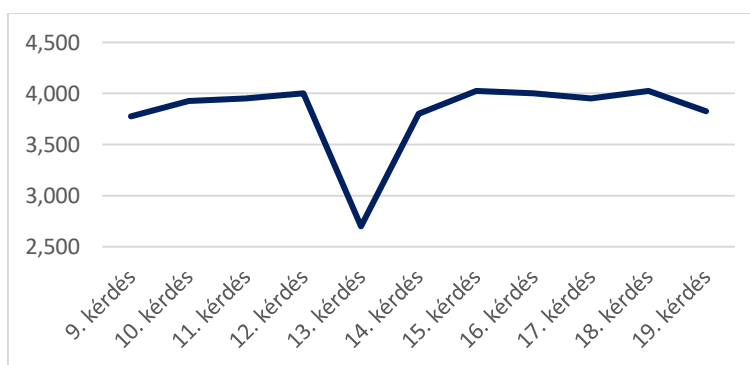


4. ábra A tanári kérdőív kérdéseire adott válaszok gyakoriságai

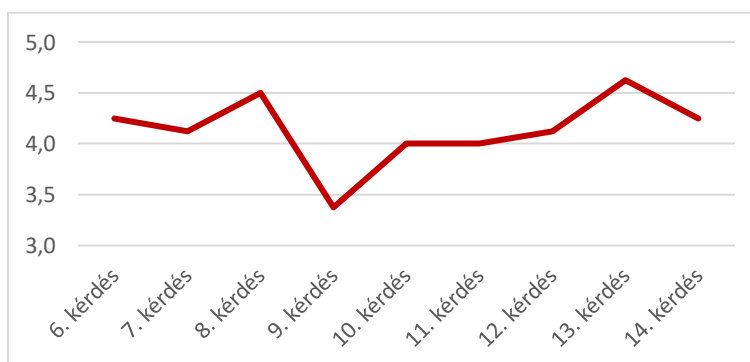
5. Hipotézisek vizsgálata

Az kérdőíves felmérés eredménye alapján elmondhatjuk, hogy a szimulációs környezet igen hasznos módszer a számítógép-hálózatok gyakorlati oktatásában. Módszer is és eszköz is egyben.

Az egyes kérdésekre adott válaszok alapján készítettem egy-egy diagramot, amely a válaszok súlyozott átlagát mutatja (5 és 6. ábrák). Amint az leolvasható a tanulók esetében az átlagpontok többsége a 4 körül található, a tanári kérdőívénél szintén a 4-es szám körül van a legtöbb. A tanulói kérdőív alapján az átlagpontszám 3,816, a szórás 0,363. A tanárok által adott átlagpontszám 4,139, a szórás, 0,336.



5. ábra A tanulói kérdőív kérdéseire adott értékek átlagai



6. ábra A tanári kérdőív kérdéseire adott értékek átlagai

A tanulók körében évfolyamonként is készült összesítés, valamint az alapján diagram (40. ábra). A diagram karakterisztikája nagyon hasonlít a . ábrán láthatóhoz. Szembetűnő, hogy a

13. kérdésnél a vonalnak minimum értéke van. (A 13. kérdés a CLI felület használatával kapcsolatos.)

1. A szimulációs környezet alkalmazása leegyszerűsíti a feladatmegoldást, mert nincs szükségünk több milliós eszközállomány beszerzésére, nem szükségesek valós kábelek, szerszámok, bármilyen számítógépteremben oktatható. Továbbá nem keletkezik hulladék, ami környezetvédelmi szempontból előnyös.

2. Megkönnyíti az elméletben tanultak gyakorlatban való alkalmazását úgy, hogy a grafikus felület és a szimuláció adta lehetőségek a tanultak vizuális rögzítésére adnak lehetőséget. Egyszerűen tudunk megtervezni és elkészíteni egy hálózatot.

3. Megkönnyíti tanári munkát, hiszen gyakorlati feladatok könnyen előkészíthetők, az esetleges károk (pl. hibás kábelezés miatt tönkremegy a berendezés) minimálisak, és a szerszámhasználatból adódó balesetveszély is elkerülhető.

6. Összefoglalás

A számítógép-hálózatok témakörének oktatása elengedhetetlenül fontos a korszerű informatikai szakképesítést nyújtó oktatási intézmények életében. Az interneten nagyon sok tudásanyag található ezzel a témával kapcsolatban is, de azok egyike sem kifejezetten a középszintű ágazati érettségire való felkészítés jegyében készült. Az egyik esetben csak a hálózatok témakör csak egy kis részét fedi le, a gyakorlatias ismeretek részben háttérbe szorulnak. A Packet Tracer számítógép hálózati szimulációs környezet nyújtotta lehetőségek a gyakorlati ismeretek elsajátítása szempontjából nagy segítséget nyújtanak. A cikkben bemutatott kérdőíves felmérés eredménye alátámasztja a szimulációs környezet létjogosultságát.

Irodalomjegyzék

Horváth József (2014). Az informatikaoktatás kihívásai a középfokú szakképzésben. *EDU Szakped*, 7(2), 116-125.

Lammle, T. (2011). *CCNA Cisco Certified Network Associate Deluxe Study Guide*. John Wiley & Sons.

Oktatási Hivatal (2017). Ágazati szakmai érettségi vizsgatárgyak. [Online] https://www.oktatas.hu/kozneveles/erettsegi/2017oszi_vizsgaidoszak

Ujbányi T. et al (2017). ICT Based Interactive and Smart Technologies in Education - Teaching Difficulties. International Journal of Management and Applied Science, 3.(10.), 72–77.

Importance of project-based learning in software development

Gy. Schrauf

MH vitéz Szurmay Sándor Budapest Helyőrség Dandár, Budaörsi street 49-53, Budapest 1118, Hungary, mh.bhd@hm.gov.hu

Abstract

This article presents a software development project that can be used in large companies. Design, problem solving, decision making and creating a project product were essential to solving the specified requirements. Using the constructivist pedagogical method, it was possible to implement an application which makes information flow easier and serves its communication in point of employee. In this type of development clearly shows well that the knowledge which acquired in the university, how can be supplemented in accordance with the requirements of the labor market.

Keywords: constructivist pedagogical method; project task; SharePoint

A projekt alapú tanulás fontossága a szoftverfejlesztésben

Schrauf Gy.

MH vitéz Szurmay Sándor Budapest Helyőrség Dandár, Budaörsi út 49-53, Budapest, 1118, Magyarország, mh.bhd@hm.gov.hu

Absztrakt

A cikkben egy nagyvállalati környezetben is felhasználható szoftverfejlesztési projekt munka kerül bemutatásra. A feladat megoldása során elengedhetetlen volt a tervezés, a problémamegoldás, a döntéshozás és egy projekttermék létrehozása. A konstruktivista pedagógiai módszert alkalmazva lehetőség adódott arra, hogy egy olyan alkalmazás kerüljön megvalósításra, amely alkalmas az információ áramlás megkönnyítésére és közlésére egy dolgozó állomány tekintetében. Az ilyen típusú fejlesztés jól mutatja, hogy a munkaerőpiac elvárásainak megfelelően milyen módon egészíthető ki az iskolában szerzett tudás.

Kulcsszavak: konstruktivista pedagógia; projektfeladat; SharePoint

1. Bevezető

Napjainkban a pedagógia tudományában egyre inkább megjelenik az iskolában szerzett és a munkaerőpiacon elvárt tudás különbségének vizsgálata. A munkaadók leginkább olyan munkaerőt keresnek, akik professzionálisan dolgoznak, ismerik a vállalati kultúrát, jó kommunikáció készséggel rendelkeznek, magasszintű problémamegoldó készséggel rendelkeznek és hatékonyan tudnak csapatban dolgozni (Kővári, 2019). A projekt alapú szoftverfejlesztés megoldást jelenthet az elméleti és alkalmazott tudás különbségének áthidalására. (Márta H., 2010) (Ilter, 2014) (Boaler, 2002)

Egy felmerült igény szerint a nagyvállalti környezetben hatékonyabb működést eredményezhet, ha egy központi felületen lehetne elérni az összes osztályhoz tartozó információt. A vállalatok számára nagy probléma, hogy a havi, vagy akár a napi munkavégzésükhöz szükséges információk, adatok lassan érnek el egyik helyről a másikra. Az információáramlást szeretnék felgyorsítani, központosítani. További elvárásként megfogalmazható a levelező rendszerek tehermentesítése is, mivel körlevelek küldésekor, a levélhez csatolt dokumentumok akár több kilobájt vagy akár több megabájt méretűek is lehetnek, végsősoron az üzenet több ezer példányban haladhat át a vállalat hálózati eszközein, infrastruktúráján. A több ezer levélhez csatolt dokumentumoknak összértéke így több gigabájt méretben terheli a levelező szervereket és a hálózatot.

Korábban kialakításra került egy szolgáltatás, Hirdetmények 1.0 megnevezéssel, amit be is vezettek egy 2010-es SharePoint farmon. Ezt a szolgáltatást szeretnék tovább fejleszteni és átmigrálni egy újabb, gyorsabb, fejlettebb 2013-as SharePoint farm környezetre. A Hirdetmény 1.0 szolgáltatás azonban nem a legoptimálisabban lett kialakítva. Az információáramlást gyorsította, a levelező szolgáltatást pedig tehermentesítette ugyan, ellenben nem teljesen automatizált a folyamatok végrehajtása. A 2013-as SharePoint farmra való átállást nagymértékben hátráltatja a dokumentációk, a folyamatábrák, a telepítő scriptek hiánya.

A Hirdetmény 1.0 szolgáltatás folyamatainak megismerésében és kialakításában részt vett informatikus állománnyal átgondoltan újra terveztük az új szolgáltatást, amit a továbbiakban Hirdetmény 2.0 néven ismerhetünk meg. A Hirdetmény 2.0 keretein belül a régi szolgáltatásnak megfelelően, de már az új 2013-as SharePoint farmon, egy központi webhelyen kerül kialakításra három lista a felhasználók felé közzétehető információkkal.

1.1. Osztály adatok lista

A lista célja, hogy a korábbi egyedi megoldások kiváltásaként, minden osztály naprakész adatokat szolgáltat az osztályról (teljes név, osztály címe, vezető neve, beosztása) és centralizáltan szerepelteti egy központi, mindenki által elérhető helyen. Előnye a listának, hogy a korábban redundánsan szereplő adatokkal ellentétben ezen a felületen könnyen hozzáférhetők és szükség szerint frissíthetők. A közzétett adatok rögzítését, módosítását csak az osztály részéről a központi webhelyen beregisztrált tartalomszerkesztő és tartalomjövőhagyó (továbbiakban együtt tartalommegosztó) végezheti. Ezen személyek egy listában, a szerkesztő és jövőhagyók listában kerülnek regisztrálásra az üzemeltetők által.

Ezzel egyidejűleg a rendszer automatikusan e-mailben értesítheti a felhasználókat az osztály adatainak változásairól, amennyiben sürgős az információ azonnali megosztása.

1.2. Szerkesztő és Jóváhagyók lista

Tulajdonképpen ez a lista szolgál arra, hogy a központi webhelyen regisztrálja a tartalommegosztókat, akik hirdetményeket helyezhetnek el és publikálhatják azt. Illetve az osztály adatok listában az osztályukhoz tartozó listaelemet módosíthatják és publikálhatják mindenki számára. Másodlagos célja, hogy a háttérben futó folyamatok e lista alapján döntenek el, hogy van-e jogosultsága az aktuális felhasználónak új hirdetmény feltöltésére, illetve az osztály adatok listában az osztályhoz tartozó listaelem módosítására. Előnye a listának, ha egy felhasználó szeretne hirdetményt publikáltatni, akkor a lista segítségével könnyedén megtalálja az osztályához tartozó személyeket, akiknek kérheti a segítségét.

1.3. Hirdetmények lista

A lista elsődleges célja a központi levelező szolgáltatás tehermentesítése. Korábban a „hirdetmények” közlését körlevelek formájában továbbították a felhasználók felé, ami túlzott hálózati forgalmat generált, különösképpen a nagyobb méretű csatolmányok miatt. A Hirdetmény 2.0 biztosítja minden osztály részére az osztályszintű, osztály keretein túlmutató, esetleg a teljes állományra vonatkozó fontos információk azonnali közzétételét minden felhasználó számára hirdetmények formájában. A túlzott hálózati forgalom elkerülése érdekében, a hirdetményeket elsősorban szöveges formátumban kell elhelyezni. A hirdetmények rögzítését, módosítását csak az osztály részéről a szerkesztő és jóváhagyók listában beregisztrált tartalomszerkesztő végezheti. Az új, illetve a módosított hirdetmények közzététele pedig csak regisztrált tartalomjóváhagyó jóváhagyásával történhet. Előnye a listának, hogy egyetlen tárhelyről, erőforrás takarékos módon elérhetővé teszi az információkat, melyeket korábban redundáns módon, minden felhasználó felé egyedileg közvetítettek. Másik előnye, hogy a csatolt dokumentumok egyetlen példányban kerülnek publikálásra, a hirdetményhez csatolva.

2. Tervezés

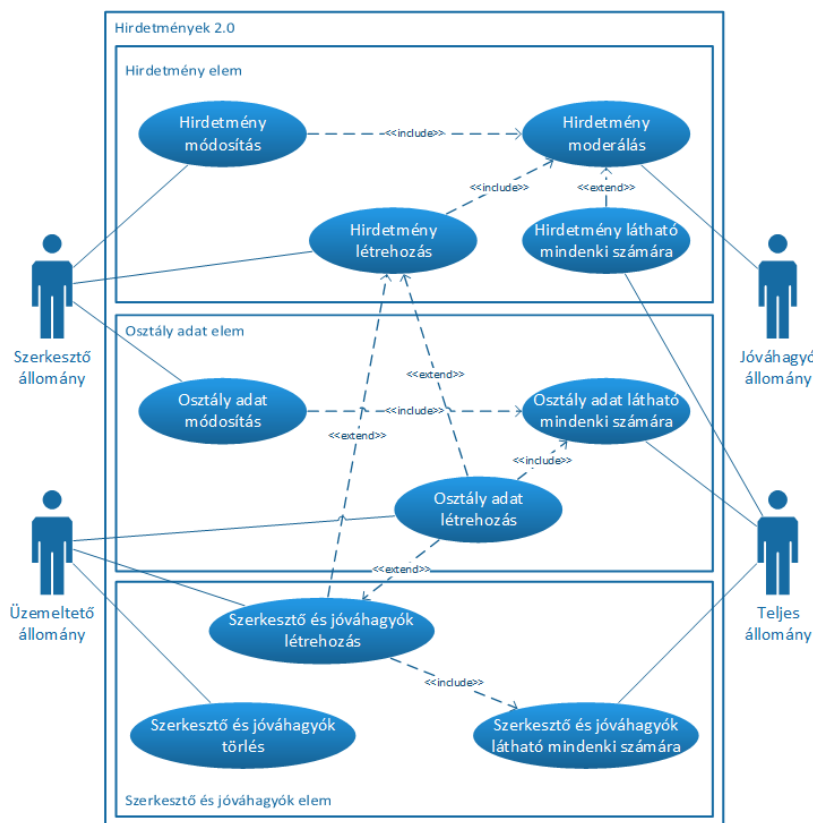
A SharePoint adta beépített funkciók segítségével nem lehet a szolgáltatást automatizáltabbá, felhasználó barátibbá tenni. Hiába a magasabb 2013-as verzió környezet, a SharePoint e verziója sem képes egyszerű beépített munkafolyamatokkal az elvárásokat teljesíteni. Így a

SharePoint egy magasabb szintű tudást igénylő, az esemény figyelések (event receivers) C# nyelven keresztül történő programozását választottuk. Eseményfigyelések használatával a Microsoft SharePoint lehetővé teszi, hogy egyéni kóddal, programozottan reagálhassunk, amikor egy adott esemény bekövetkezik egy SharePoint objektum változtatásakor. Visual Studio-ban úgy nevezett SharePoint 2013 projektet hozunk létre, amiben SharePoint névterek (namespace) segítségével automatizáltabb és magasabb szintű működést érhetünk el, mint az egyszerűbb beépített funkciókkal.

Elsődleges szempontunk, hogy a Hirdetmények 2.0 felhasználó barát, könnyen kezelhető és áttekinthető legyen mind a felhasználók, mind pedig az üzemeltetők számára. A különböző jogosultsági szintek és csoportok segítségével könnyedén meg tudjuk határozni, hogy a Hirdetmények 2.0 az aktuális felhasználónak (szerepkörét tekintve) milyen adatokat, információkat jelenítsen meg.

2.1. A rendszer használat felőli nézete, UML alapú használati-eset diagram tervezése

Az 1. ábra használati eset diagram segítségével szemléltetjük a rendszer szerepkörök szerinti működésének különböző eseteit, amiket a tervezés során feltártunk.



1. ábra: A rendszer használat felőli nézete

A szolgáltatás használat felőli működése szempontjából hat esetet különböztethetünk meg:

- Osztály adat létrehozás eset: amikor tartalommegosztóként szeretne egy osztály megjeleni a központi webhelyen, akkor az üzemeltető állomány egyik tagja - aki jogosult új elem felvételére - az osztály adatok listában felveszi az osztály adatait új elemként. A teljes állomány számára pedig láthatóvá válik az új osztály adat.
- Szerkesztő és jóváhagyók létrehozása eset: amikor az üzemeltető állomány egyik tagja az előzőekben már regisztrált osztályhoz felveszi az általuk kijelölt tartalomszerkesztőt és annak tartalomjóváhagyóit. A teljes állomány számára pedig láthatóvá válik az új szerkesztő és jóváhagyók.
- Osztály adat módosítás eset: amikor egy tartalommegosztóként regisztrált osztály adataiban változtatást kell alkalmazni. Első körben az osztályhoz tartozó, beregisztrált, szerkesztő és jóváhagyók feladata az osztályuk vonatkozásában a módosításokat végrehajtani. Második körben az üzemeltető állomány is végre tudja hajtani az osztály adatainak módosítását, a háttérben való támogatás végett. A teljes állomány számára pedig láthatóvá válik a módosított osztály adat.
- Szerkesztő és jóváhagyók törlése eset: amikor az osztályhoz tartozó szerkesztő, vagy jóváhagyók egyikének, vagy mindannyiuknak megszűnik a tartalommegosztói feladata osztálya vonatkozásában. Az üzemeltető állomány egyik tagja törli a szerkesztő és jóváhagyók listában a regisztrált tartalommegosztókhoz tartozó elemet.
- Hirdetmény létrehozás eset: amikor a tartalommegosztóként regisztrált osztály fontos információkkal szeretné tájékoztatni a célközönséget, vagy a teljes állományt. Az előzőekben a szerkesztő és jóváhagyók listában már regisztrált szerkesztő létrehoz egy új elemet (hirdetményt) a hirdetmények listában. A teljes állomány számára pedig láthatóvá válik az új hirdetmény.
- Hirdetmény módosítás eset: amikor a tartalommegosztóként regisztrált osztály módosítani szeretné a már előzőekben létrehozott hirdetményt. A hirdetmények listában, a regisztrált szerkesztő megkeresi az általa létrehozott módosítandó hirdetményt és módosítja az adatokat benne. A teljes állomány számára pedig láthatóvá válik a módosított hirdetmény.

3. Implementálás és integrálás

Első körben a webhely oszlopokat és a listákat kell létrehozni, majd a listákhoz a webhely oszlopokat kell hozzácsatolni. PowerShell scriptek alkalmazásával gyorsabban létre tudjuk

hozni a kívánt webhelyet, listát, oszlopokat vagy más egyéb SharePoint objektumot. A mi esetünkben nem elég a PowerShell parancsértelmezőt megnyitni, mert nem tartalmazza a SharePoint farmhoz tartozó névtereket. Erre külön parancsértelmezőt fejlesztett ki a Microsoft, még pedig SharePoint Management Shell néven. Ez a rendszerhéj csak a SharePoint szervereken érhető el, ezért minden esetben be kell lépni egy SharePoint szerverre, ha ilyen jellegű parancssort vagy scripteket akarunk futtatni.

A kódbázis két külön részből áll. Az első része SharePoint Management Shell scriptekből áll, ami a webhelyen létrehozza és bekonfigurálja a szolgáltatás alapját. A másik rész pedig Visual Studio programmal létrehozott SharePoint 2013 projekt fájl lesz (továbbiakban: solution), ami tartalmazza az esemény figyeléseket és ezek hatására bekövetkező folyamatokat.

3.1. A kódbázis felépítése

Maga a kódbázis öt jól elkülöníthető részcsoporthoz áll:

- Elem esemény figyeléshez létrehozott osztályok
- Listák, csoportok és jogosultsági szintek részére létrehozott osztályok
- SharePoint Server Side Object Modell (SSOM) részére létrehozott osztályok
- SharePoint Client Side Object Modell (CSOM) részére létrehozott osztályok
- Üzenet küldésnek létrehozott osztályok

3.2. Az alkalmazásban megvalósított főbb funkciók

SharePoint listákat többféleképpen lehet deklarálni szerver oldali objektumként. Legtöbb esetben a lista megjelenítési nevével hivatkoznak rá. Ezt a lehetőséget szerettük volna elkerülni annak biztosítása érdekében, hogy ha a későbbiekben a lista megjelenítési nevét meg akarják változtatni, azt gond nélkül megtehesse az üzemeltető állomány. Ezért kerestünk egy olyan lehetőséget, amivel a lista url paraméter megadásával tudjuk deklarálni a lista objektumot. Azért döntöttünk ezen megvalósítás mellett, mert tapasztalataink szerint az elemek url elérését ritkán változtatják meg és ékezetes betűket sem alkalmaznak benne, így az bizonyult a legjobb választásnak. A létrehozott funkciónak átadandó paraméterek az SPWeb szerver oldali objektum és egy szöveg típusú adattag:

```
public static SPList ListByUrl(SPWeb web, string url)
{
    return web.GetList(SPUrlUtility.CombineUrl(web.ServerRelativeUrl, url));
}
```

A fejlesztő programban a függvény meghívása:

```
SPList list = GetSO.ListByUrl(web, "/List/hirdetmenyek");
```

A tervezés során arra a következtetésre jutottunk, hogy az elemnek egyedi engedéllyel kell rendelkeznie. Alapértelmezetten a SharePoint objektumok létrehozásuk során öröklik a szülőtől az engedélyeket, így első lépésben meg kell törni az öröklést, majd beállítani a hozzáféréseket. Az öröklés megtörésére `BreakRoleInheritance()` metódust lehet használni alapértelmezetten az `SPIItem` objektumon is. A metódust használva be lehet állítani, hogy a szülőtől kapott engedélyek megmaradjanak az elemen vagy sem. Mivel minden esetben törölni kellene, ezért ezzel a metódussal töröljük az örökölt engedélyeket, így nem kell létrehozni a tisztításhoz szükséges eljárást. Két logikai értéket kell megadni a metódusnak. Az első érték igaz állítása esetén lemásolja a jogosultságokat a szülő elemtől és megtartja őket a megtört elemen, hamis állítás esetén nem hajtja ezt végre. A másik pedig, hogy az elemünk gyermek elemeiről szintén törölje ezeket az örökölt engedélyeket. Az esetünkben az első logikai érték mindig *false* a második pedig *true*. Viszont ehhez a beállításhoz már több jogosultsággal kell rendelkeznie az aktuális felhasználó nevében futtatott szolgáltatásnak. A regisztrált szerkesztők, jóváhagyók nem rendelkeznek ilyen engedéllyel, ezért előtte magas jogosultságkezelést kell alkalmaznunk az elemen, majd ezután tudja a szolgáltatás is végrehajtani az öröklések leállítását:

```
public static void Inheritance(SPListItem listitem)
{
    SPListItem elevatedListItem = GetSO.ElevatedLitsItem(listitem);
    elevatedListItem.BreakRoleInheritance(false, true);
}
```

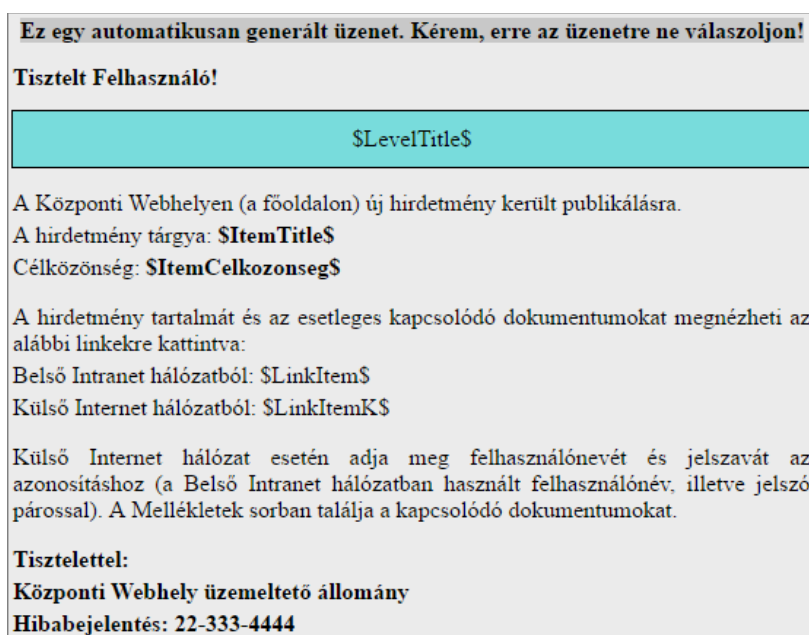
3.3. *Az alkalmazás legfontosabb kódbázisai*

A kódbázisnak minden egyes része fontos, de ha ki kellene emelni néhányat, akkor a szerver oldali objektum modellben alkotott magas jogosultságkezelés, mint legfontosabb függvény, és az üzenet küldésének létrehozott osztályok kerülnének kiemelésre. A magas jogosultság kezelés lehetőség a forráskód megírása közben sok problémát oldott meg, amikor a felhasználó nevében futó szolgáltatás olyan listákhoz is hozzáfért, amelyhez nincs köze, és így a szolgáltatás problémamentesen tudott tovább futni. Többször is alkalmaztuk ezt az eljárást: amíg a felhasználó rendelkezik az összes olyan engedéllyel, amivel még tud dolgozni az elemen vagy elemeken, addig a felhasználó nevében futott a szolgáltatás. Mihelyst már nem voltak elegendők a felhasználó engedélyei, akkor forgattuk át az elemet emelt jogosultsági szintre és folytatta tovább a szolgáltatás az utasítások végrehajtását. Ennek

köszönhetően nem akadt meg a szolgáltatás hozzáférési hiba miatt. Ellenben így sokszor újra kellett tervezni az algoritmust, hogy meddig legyen úgymond a felhasználó kezében a szolgáltatás működése, és mikor kapjon teljes hozzáférést ezzel az eljárási lehetőséggel.

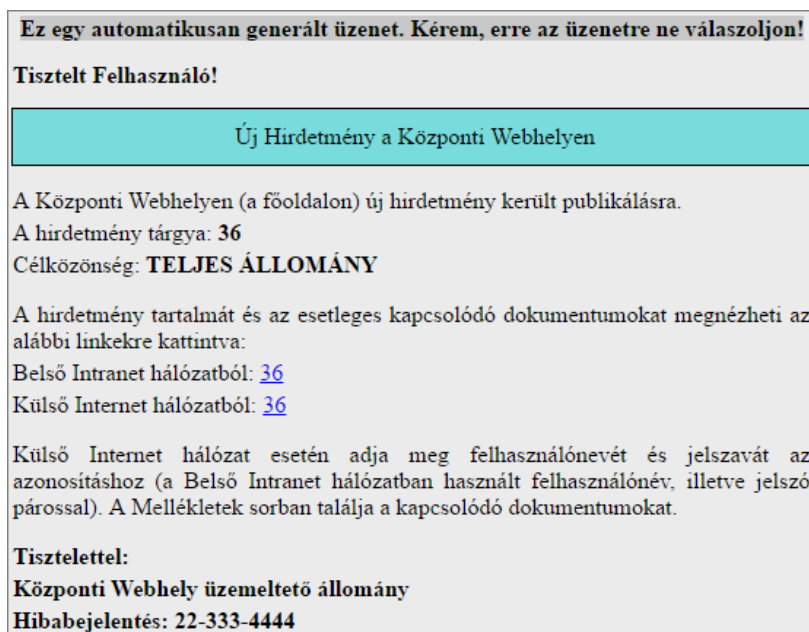
Az üzenet küldésének létrehozott osztályok és metódusok csoportja egyike a legnagyobb, legösszetettebb, külön elszeparálható szigetrésze a kódbázisnak. Ennek köszönhetően bármikor felhasználható más szolgáltatások tervezésekor és alkalmazható más projektekben. Ebben a részben alkalmazzuk a levél sablonok listában a token cserét. Vegyük azt az esetet, amikor új hirdetményt publikáltak és a felhasználókat azonnal akarják értesíteni. A forráskódban elérkezünk ahhoz a részhez, ahol a levél sablonok listából megkeressük az új hirdetményhez tartozó levél sablon törzsét.

A szövegrész egyes részei \$ jelek között helyezkednek el, mint például a \$LevelTitle\$, vagy a \$LinkItem\$. Ezek lesznek azok a tokenek, amiket dinamikusan tudunk változtatni. Most nézzük meg egy web böngészőben, hogy is néz ez ki 2. ábra:



2. ábra: Sablon üzenet token csere előtt

Jól látható, hogy így már áttekinthetőbb és a megjelenítése is szebb, viszont még hiányoznak információk a hirdetményről. Minden létrehozott hirdetménynek eltérő lesz a tárgya (ami a hirdetmény címe), a célközönsége, és a legfontosabb, a hirdetmény elérési url linkje. A tokeneknek nevezett dinamikusan változtatható értékeket a kiküldés előtt még meg kell változtatni az aktuális hirdetmény adataival. Ezeket az információkat a levél sablon kiolvasása után egy Replace() metódussal és további függvények segítségével kicseréljük és a 3. ábra látható módon fog kinézni az üzenetet.



3. ábra: Sablon üzenet token csere után

4. A program tesztelése

A tesztelés fontos része a szoftverfejlesztésnek és fő célja a hibák bekövetkeztének csökkentése, valamint programhibák megtalálása. A tesztelést úgy kell megtervezni, hogy a lehető legtöbb hibát észleljük. A forráskód írása közben folyamatosan, egyes modulok megalkotása után hibakereséseket végeztünk. A Visual Studio fejlesztő program segítségével futási időben tudunk a forráskódon soronként végig haladni és figyelemmel kísérni az egyes részek helyes vagy helytelen működését. Szerencsére nem kell mindig az elejétől kezdve soronként végig mennünk a forráskódon, a fejlesztő program segítségével elég töréspontokat elhelyezni. A fejlesztő program egyből az első töréspontra ugrik és innen már lépésenkénti végrehajtással is tovább nézhetjük a forráskódot. A lépésenkénti végrehajtásban az egyes lépések lefutása előtt lehetőségünk van elemezni a program működését. Egy hiba felismerését és kijavítását követően újra tesztelni kell, így meggyőződhetünk róla, hogy a hiba eltűnt (ellenőrző teszt).

Lehetséges tesztelési módszerek (Ammann,, Offutt, 2016) (Myers, Sandler, Badgett, 2011):

- Egység- és rendszer teszt
- Fekete és fehér doboz tesztelés
- Verifikáció és validáció

Mivel egyik módszerrel sem lehetséges a kimerítő tesztelés, el kell fogadnunk, hogy nem tudjuk egyetlen program hibamentességét sem szavatolni. A tesztelés során feltárt hibák

helyének megállapítása a hiba keresése. Általában a program növekedésével együtt növekszik a hibák száma is.

4.1. *Egység- és rendszer teszt*

Egységteszt során a rendszer elemeit, moduljait teszteljük külön-külön, egymástól függetlenül. Az általunk deklarált kisebb, nagyobb egységek például:

- új hirdetmény esetén létrehozó felhasználó jogosultságának ellenőrzése
- a szerkesztő és a hozzá tartozó jóváhagyók adatainak megállapítása
- tartalommegosztókhöz tartozó osztály elem keresése
- extranet eléréséhez szükséges adatok megállapítása
- hirdetmény másolása extranet webhelyre
- jogosultság öröklésének megtörése
- jogosultságok beállítása
- azonnali üzenet küldésének megállapítása
- célközönség email címének megállapítása
- értesítő üzenet összeállítása
- értesítő üzenet küldése

4.2. *Fekete és fehér doboz tesztelés*

A dinamikus tesztelési módszerek alapelve, hogy a programot működés közben vizsgáljuk. Két esetét ismerjük a fekete és a fehér doboz tesztelést.

Fekete doboz tesztelés (adatvezérelt tesztelés) esetén csak a működési rendellenességek felderítésére van lehetőségünk, a hibák helyének lokalizálására nincs. Ebben az esetben a tesztelő nem rendelkezik információval a szoftver belső struktúrájáról, tehát a forráskód ismeretlen számára. Csak a bemenő és kimenő adatok kapcsolatát figyeljük. Az ilyen tesztek jellemzően nem a fejlesztő csapat végzi.

- Az értesítő beállítása listán a teljes állomány számára nincs felvéve terjesztési lista email cím. Ilyenkor, ha az osztály adatok listában új elemet vesznek fel és a felhasználókat is értesíteni akarják, akkor nem kap értesítő üzenetet a teljes állomány. Ellenben a szolgáltatásnak hiba nélkül kell tovább működnie.
- Az értesítő beállítása listán az osztályhoz nincs felvéve a hozzá tartozó terjesztési lista email címe. Ilyenkor, ha a hirdetmények listán új hirdetményt publikálnak és a

célközönséget is értesíteni akarják, akkor a célközönségnek beállított osztály nem kap értesítő üzenetet. Ellenben a szolgáltatásnak hiba nélkül kell tovább működnie.

- A felhasználó valamilyen oknál fogva bekerült vagy beragadt a szerkesztő csoportba, és a szerkesztő és jóváhagyók listában nincs regisztrálva, továbbá a felhasználó új hirdetményt vesz fel a hirdetmények listán. Ilyenkor a szolgáltatásnak törölnie kell a hirdetményt, értesítenie kell a felhasználót, hogy nincs tartalommegosztói jogosultsága a hirdetmény publikálásához.

Fehér doboz tesztelés (logika vezérelt tesztelés) esetén figyelembe vesszük, hogy ismerjük a forráskódot és tudjuk, hogy mikor minek és hogyan kell történnie. Tehát a tesztelőnek pontosan ismernie kell a szoftver belső szerkezetét. A teszteseteket a program struktúrája alapján választjuk, annak tudatában, hogy nem lehet minden végrehajtási sorrendre kipróbálni. Az utasítás lefedése, hogy minden utasítás egyszer hajtódik végre. Döntés lefedése, hogy minden elágazás minden ágát legalább egyszer körbejártuk.

4.3. *Verifikáció és validáció*

- Nulladik tesztelésként határoztuk meg a PowerShell scriptes telepítés utáni ellenőrzést, hogy minden oszlop, lista, csoport és jogosultsági szint a megadott paraméterekkel, elnevezésekkel szerepel-e a központi webhelyen. Valamint mindezek megfelelően vannak-e felkonfigurálva. Ugyanis, ha a strukturált osztályok adataiban bármelyik elem neve vagy tulajdonsága akár egy karakterrel is másként szerepel, mint a szolgáltatás kódjában, akkor a működési folyamatok hibával megállnak és a szolgáltatás helytelen működését okozhatja.
- Az üzemeltető csoportba felvett felhasználó mindegyik listát meg tudja-e nyitni, új elemet felvenni, meglévő elemet módosítani, törölni tud-e. Ugyanis a hirdetmények lista kivételével mindegyiken képes ezekre. Új hirdetmény felvételekor egy értesítő üzenetet kell kapnia, hogy nem rendelkezik tartalommegosztói jogosultsággal, valamint az elemet törölnie kell a szolgáltatásnak. Az üzemeltető felhasználó az olvasó és szerkesztő csoportba képes-e új felhasználót felvenni, és meglévőt kivenni a csoportból.
- Az osztály adatok listán új elem felvételekor vagy létező elem módosításakor és a felhasználók értesítését aktiválva a teljes állomány meg kapja-e az értesítő üzenetet és az információs adatok megfelelnek-e benne. Elem törlésekor nem kap-e egyetlen felhasználó sem értesítést.

- A szerkesztő és jóváhagyók listában az üzemeltető felhasználó által felvett új tartalommegosztók megkapják-e az értesítő üzeneteket és az információs adatok megfelelnek-e benne. A szerkesztőnek kijelölt felhasználónak a tartalomszerkesztői, a jóváhagyónak pedig a tartalomjóváhagyói üzenetet.
- A regisztrált szerkesztő képes-e az osztályához tartozó osztály elemet szerkeszteni, de törölni nem. Képes-e új hirdetményt létrehozni, csak a saját hirdetményét módosítani, de a saját, illetve mások hirdetményét törölni nem. A hirdetmény elutasításáról vagy jóváhagyásáról kap-e értesítő üzenetet a megfelelő adat tartalommal.
- A regisztrált jóváhagyó képes-e az osztályához tartozó osztály elemet szerkeszteni, de törölni nem. Képes-e csak a hozzá tartozó hirdetményt moderálni, de a saját, illetve mások hirdetményét törölni nem. Új hirdetmény felvételekor kap-e értesítő üzenetet a moderálandó hirdetményről, a jóváhagyó és az információs adatok megfelelnek-e benne.
- Új hirdetmény publikálása után a teljes állomány képes-e megnyitni azt. Ha a célközönség számára értesítést kértek az új hirdetménnyel kapcsolatban, akkor kapott-e a célközönség értesítő üzenetet és az információs adatok megfelelnek-e benne. Továbbá az extranet oldali webhelyen a publikált hirdetmény megjelenik-e összes csatolmányával együtt a teljes állomány számára.
- A törölt szerkesztő képes-e a volt osztályához tartozó osztály elemet szerkeszteni, képes-e új hirdetményt létrehozni. Törlést követően kapott-e értesítő üzenetet tartalomszerkesztői feladatainak elvételéről és az információs adatok megfelelnek-e benne.
- A törölt jóváhagyó képes-e a volt osztályához tartozó osztály elemet szerkeszteni. Képes-e a hozzá tartozó hirdetményt moderálni. Törlést követően kapott-e értesítő üzenetet tartalomjóváhagyói feladatainak elvételéről és az információs adatok megfelelnek-e benne.

5. Összegzés

A projekt résztvevői a Hirdetmények 2.0 létrehozása során nagyon sokat tanulhattak a SharePoint kulisszái mögötti területekről. A felhasználói felületen kialakítani egységesen működő szolgáltatást egyre nehezebbnek bizonyult, nagyon sokszor olyan igények merültek fel, amit már így nehezen lehet megoldani. A szolgáltatás megírásával világossá vált, hogy a SharePoint-on az igényeket létrehozni, alkalmazni, működtetni sokkal könnyebb, mint

felhasználói felületen összehozni. Természetesen itt is vannak nehezen megvalósítható igények, de a lehetőségek tárháza messzemenően meghaladja a felhasználói felület adta lehetőségeket.

A projekt alapú oktatás és tanulás több műszaki területen is jelenthet hatékonyabb tudás szerzést, jobb problémamegoldó készséget és hatékony kommunikáció fejlődést. (Katona et al, 2016).

Irodalomjegyzék

Ammann, P., & Offutt, J. (2016). *Introduction to software testing*. Cambridge University Press.

Boaler, J. (2002). Learning from teaching: Exploring the relationship between reform curriculum and equity. *Journal for Research in Mathematics Education*, 33(4), 239–258.

Ilter, Í. (2014). A study on the efficacy of project-based learning approach on Social Studies Education: Conceptual achievement and academic motivation. *Educational Research and Reviews*, 9(15), 487.

Kővári, A. (2019). A felnőttoktatás 4.0 és az az ipar 4.0 kihívásai az életen át tartó tanulásban. *PEDACTA*, 9(1), 9–16.

Katona J et al (2016). A Brain–Computer Interface Project Applied in Computer Engineering. *IEEE Transactions on Education*, 59(4), 319-326.

Márta, H. (2010). Projektmódszer a 21. században II. *Új pedagógiai szemle*, 1, 2.

Myers, G. J., Sandler, C., & Badgett, T. (2011). *The art of software testing*. John Wiley & Sons.

Implementing a Java EE based information system as a student self-study task

Cs. Csizmadia

MÁV Szolgáltató Központ ZRT, Könyves Kálmán avenue 54-60, Budapest 1086, Hungary, jabbaco@gmail.com

Abstract

One of the highest levels of self-study is the problem solving, which primarily means solving a practical task using existing knowledge. During my studies, this development was the most complex learning process, since the knowledge, which had already been acquired, was not enough to solve the problem, moreover I had to take several problems. I had a challenging job to develop an information system for the agents of various insurance companies, since their previous administrative and client support methods include a lot of imperfection. The use of their technology is difficult, and it involves numerous possibilities for error, and is not up to date at all. Therefore, the exact method relies on the personal inventiveness and preparedness of the agents, which has an effect on the achievement of the agents and on the exterior opinion about the company. Thus, it also affects the competitiveness. In the process of acquiring knowledge, it has become essential to organize learning and manage time efficiently.

Keywords: self-study; information-system; agency

Java EE alapú információs rendszer megvalósítása önálló tanulói feladatmegoldás során

Csizmadia Cs.

MÁV Szolgáltató Központ ZRT, Könyves Kálmán krt 54-60, Budapest, 1086, Magyarország, jabbaco@gmail.com

Absztrakt

Az önálló tanulás egyik legmagasabb szintje a problémamegoldás, amely során elsősorban a már meglévő ismereteket felhasználva oldunk meg egy-egy gyakorlati feladatot. A képzésem során ez a fejlesztés valósult a legösszetettebb tanulási folyamatnak, mivel a problémamegoldáshoz nem volt elegendő a már megszerzett ismeret, több problémaszituációval is szembe kellett nézmem. Kihívást jelentő feladatom volt egy információs rendszer fejlesztése különféle biztosítási cégek ügynökei részére, mivel azok adatkezelési, ügyféltámogatási módszerei esetenként kívánnivalót hagynak maguk után. Technológiájuk használata sokszor nehézkes, számos hibalehetőséget is magában hordoz, és így cseppet sem korszerű. Ezáltal a pontos metodika az ügynökök egyéni leleményességén, felkészültségén múlik, amely a biztosítási cég teljesítményére, külső megítélésére és ezekből következően a versenyképességére is negatív hatást gyakorol. Az ismeretszerzés során elengedhetlenné vált a tanulás megszervezése és az idővel való hatékony gazdálkodás is.

Kulcsszavak: önálló tanulás; információs rendszer; ügynök

1. Bevezető

Az önálló hatékony tanulás útján megvalósított információs rendszer fejlesztés számos lehetőséget rejt a tanórán kívüli nem formális tanulás környezetben (Váczy, 2012). A nem

formális tanulás során a tanórákon túlmutató új ismeretekre lehet szert tenni, amelyek feldolgozhatók és beépíthetők a tanulási stratégiába, másrészt a szakmai kompetenciák fejlődéséhez is több szempontból hozzájárul. (Sagitova, 2014) (Barbosa et al, 2017) A nem formális tanulási módszerek fontos alapot adnak az életen át tartó tanulás megalapozása tekintetében is (Kővári, 2019). Mindezek mellett korábbi tanulmányok és tapasztalatok is segítették az önálló feladatkidolgozás során szükségessé vált problémák feltérképezését, megoldását.

1.1. Probléma ismertetése

Egy biztosítási cég ügynökei az eredményes munkásságuk érdekében számos nyilvántartást kénytelenek naprakészen vezetni. Nem elég csupán az ügyfelek adatait tárolni, hanem az azokkal kapcsolatos kisebb-nagyobb teendőket is szinkronban kell tartaniuk. Mindez a gyakorlatban 2-3 táblázat és/vagy határidőnapló használatát jelenti elektronikus, vagy papíralapú formában. A bennük lévő adatok egy része így természetesen redundáns, ezért ezek állandó naprakészen tartása kényelmetlen feladatot jelent, és számos hibalehetőséget is magában hordoz. Ráadásul eme módszer korunk kívánalmainak sem tud eleget tenni, hiszen keresésre, összesített adatok lekérdezésére csak nagyon korlátozott mértékben használható. Miközben sem navigációs lehetőséget, sem közösségi hálózatokhoz való csatlakozást nem biztosít. Így az adatok nyilvántartása és kezelése az ügynökök egyéni leleményességén, felkészültségén múlik, amely így végső soron a biztosítási cég teljesítményére, versenyképességére is hatással van.

A felvázolt helyzet megoldására egy szoftverfejlesztő cég vállalkozott, más feladatok elvégzése mellett és viszonylag nagyobb határidővel. Személyes kapcsolatom révén én részt vehettem a munkálatok során, pontosabban a fenti probléma megoldását teljes egészében magamra vállaltam. Cserébe a cégvezetés hasznos információkkal látott el a biztosítási ügynökök ügymenetéről, valamint a biztosítók lehetséges elvárásairól. Így ilyen összefüggésben e szoftverfejlesztő cég lényegében a megrendelőnek tekinthető, ezért a későbbiekben így is hivatkozom rájuk.

1.2. Elérendő cél kitűzése

A cél elérése érdekében számos feladatot kell sikerrel teljesíteni. Előbb magát a problémát kell behatóbban tanulmányozni, majd a lehetséges megrendelők ezzel kapcsolatos igényeit kell feltárni. Ezt követően biztosítani kell – a formálódó elképzelések alapján – a szükséges

adatok tárolását, megjelenítését és a velük kapcsolatos különféle szolgáltatásokat, funkciókat is. Előbb csupán gondolati síkon, majd később a fizikai valóságban is. Lényeges, hogy az információk közlésén és igény szerinti kezelésén túl az adatellenőrzést és a felhasználói azonosítást is meg kell oldanom. Majd a folyamat végeztével egy mindenre kiterjedő tesztelést kell foganatosítanom.

1.3. Távlati cél megfogalmazása

Jelen fejlesztés távlati célja a biztosítói adatbázisokhoz való laza kapcsolódás megteremtése a szolgáltatást igénybe vevő ügynökök beazonosítása érdekében. Erre legkorábban viszont csak egy sikeres tesztidőszakot követően kerülhet sor. Ezért a mostani fejlesztés során még egy teljesen független szoftver kialakítására kell törekedni.

2. Követelmény megfogalmazása

- A készítendő szoftver szolgáltatásait egy webhelyen keresztül lehessen elérni, amely a kliensként használt informatikai eszközök webböngészőjében jelenjen meg.
- A szolgáltatások igénybevételének előfeltétele a sikeres autentikáció.
- Az ügyfelekkel kapcsolatos beviteli adatokat egy vagy több weboldalon keresztül lehessen megadni.
- A bevitt adatok egy adatbázis-kezelő rendszerben kerüljenek eltárolásra.
- A szolgáltatások igénybevételekor a válasz is a weboldalon jelenjen meg, amelynek tartalmaznia kell az adatbázis-kezelő által tárolt adatok egy később meghatározott részét.
- A megjelenő adatokat lehessen rendezni, valamint utólag módosítani, törölni.
- A beviteli adatok megadása lehetőleg egyszerűen és gyorsan történjen.
- A webhely legyen áttekinthető, mégis tartalmazza a legfontosabb adatokat, biztosítsa a fontosabb funkciókat.
- A szoftver teremtsen meg a közösségi hálózatokhoz, navigációs rendszerekhez és a biztosítói adatbázishoz való későbbi, további fejlesztést igénylő csatlakozás alapvető feltételeit.

3. Specifikáció

Az alkalmazással szemben támasztott alapvető elvárások megismerését követően sor került a program specifikációjára (elvárt szolgáltatások pontosítása, részletezése), amelynél

igyekeztem szöveges leírással kiegészített, formáknak alávetett módszert alkalmazni UML (Unified Modeling Language – egységes modellező nyelv) nyelven megfogalmazott diagramok segítségével. Hiszen ezáltal érhető el a leginkább a könnyű áttekinthetőség, és általa a félreértések elkerülése, valamint a tesztelési lehetőségek biztosítása. (Sándor S, László V, 2003) Továbbá Katona J et al. 2014 és 2016 munkákban is megjelenik az UML alkalmazása, mint hatékony szoftvertervezést támogató eszköz.

3.1. *Adattárolási igények áttekintése*

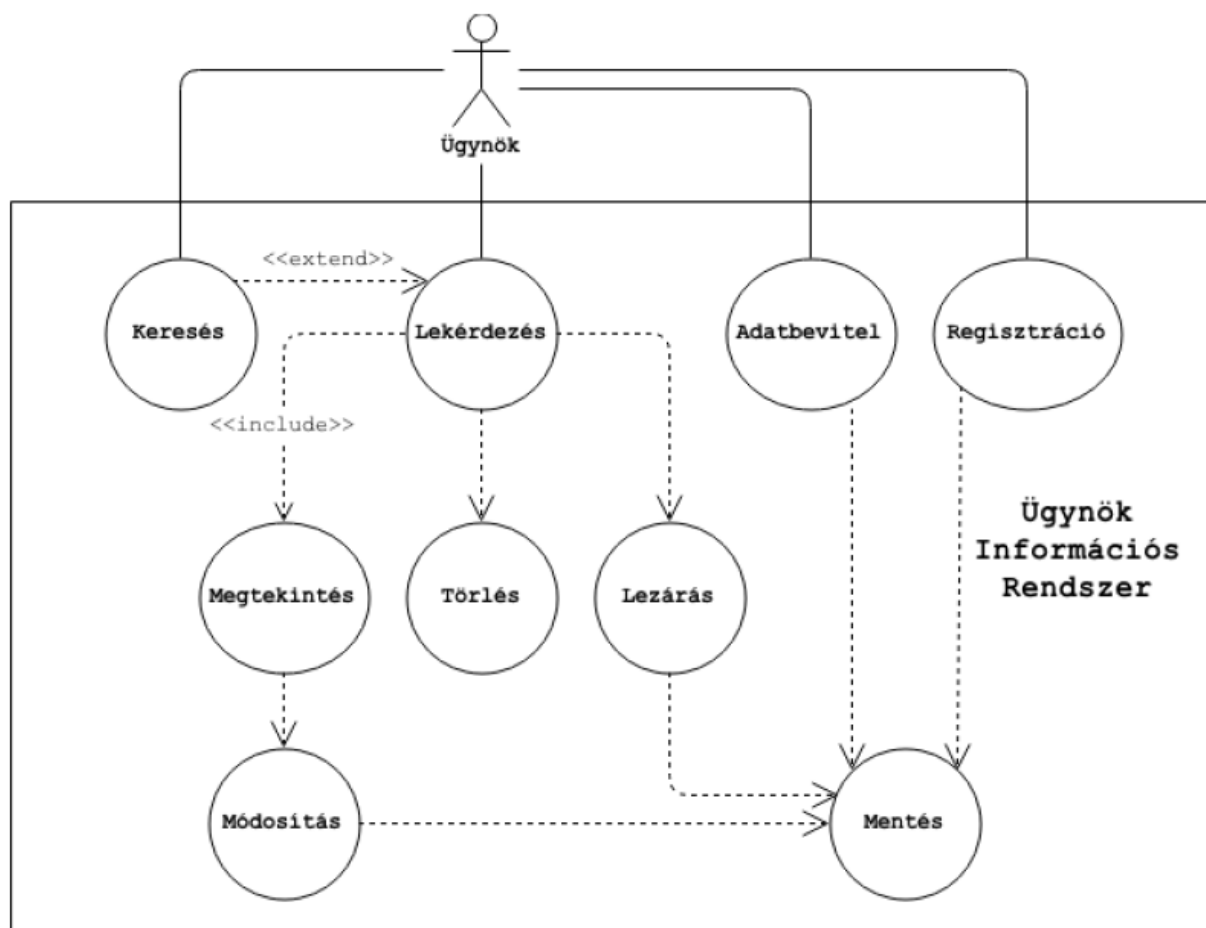
A specifikáció első lépéseként a tárolandó adatokat kellett áttekinteni, melyeket rövid idő alatt 5db egyedhalmazra választottunk szét. Majd ezt követően feltérképeztük a közöttük fennálló kapcsolatrendszer is. Egy-egy egyedhalmazon a modellezni kívánt világ közös tulajdonságokkal rendelkező egyedeinek halmazát értjük, melyek akkor vannak egymással kapcsolatban, ha van olyan, a feladat szempontjából fontos rendezőelv, amely az egyedhalmazok egyedeit egymáshoz rendeli. (Antal B, 2012)

Mindebből a további egyeztetési és tervezési munka érdekében egy egyszerűsített egyedkapcsolat modellt készítettem, mellyel szemléletes módon, könnyen érthető formában tudtam ábrázolni a készítendő adatbázist.

3.2. *Szolgáltatások megfogalmazása*

- Egy újabb konzultáció során sikerült megfogalmazni a tervezendő alkalmazás alapvető szolgáltatásait, amelyet egy használati eset diagram segítségével rögzítettünk (1. ábra).
- Lekérdezés: Az adatbázisban tárolt adatok lekérdezése és megjelenítése az adott weboldalon, listák formájában. Külön lekérdezés szükséges az ügyfeladatok, a hozzájuk kapcsolódó ügyszakok és az ügyekhez kapcsolódó részfeladatok részére.
- Keresés: Szelektív lekérdezés az adatbázisban szereplő ügyfelek közül. A kereséshez szükséges bemenő paraméter az ügyfél neve, illetve a nevének töredékszavai.
- Adatbevitel: Új adatok felvitel az adatbázisba. Külön beviteli lehetőség szükséges az ügyfeladatok, az ügyek és a részfeladatokhoz részére. Az adatbevitel során fontos szempont a gyorsaság és az egyszerűség, ezért csak a legfontosabb adatok bevitel legyen kötelező, és ahol megoldható, ott legyen lehetőség alapértelmezett értékek megadására is.

- Regisztrálás: A készülő webalkalmazás szolgáltatásait igénybe venni kívánó biztosítási ügynökök regisztrálhatják saját magukat egy regisztrációs űrlap kitöltése által.
- Megtekintés: A weboldalon lévő listák egyes elemeinek adatait jeleníti meg részletesebben. A lekérdezések során alapértelmezetten mindig az első rekord adatai kerüljenek kifejtésre.
- Törlés: Törli az adott lista egyes eleméhez tartozó rekordot az adatbázisból.
- Lezárás: A listák egyes elemeinek státuszát „Zárt”-ra állítja. A lekérdezések eredményeiben a Zárt rekordok alapértelmezetten ne jelenjenek meg.
- Módosítás: Elvégzi a lista éppen megtekintett rekordjához tartozó adatok módosítását, illetve új adatokkal való kiegészítését.
- Mentés: Új adatok felvitele vagy a régiek módosítása után menti azokat az adatbázisba.



1. ábra: Elvárt szolgáltatások

4. Felhasznált technológiák

A további tervezés érdekében ki kellett választani a felhasználandó technológiákat, valamint át kellett tekinteni az általuk kínált lehetőségek körét. Mivel a tervezendő szoftver végső produktuma egy vagy több weboldal megjelenítése, ezért az elemzést is itt volt érdemes elkezdeni.

A weblapok forráskódját XHTML (Extensible Hypertext Markup Language – kiterjesztett hiperszöveges jelölőnyelv) formátumban, HTML5 nyelven terveztem megírni, míg a formázást és elrendezést CSS (Cascading Style Sheets – egymásba ágyazott stíluslapok) alkalmazásával kívántam biztosítani.

A legfontosabb döntés azonban a használandó programozási nyelvekre vonatkozott, a megfelelő nyelv kiválasztása ugyanis egy hozzá illő fejlesztői környezettel együtt jelentősen lerövidítheti a forráskód elkészítésére fordított időt, ráadásul a végeredmény minőségére, hatékonyságára is hatással lehet.

A követelményrendszer ismételt áttanulmányozása során arra a következtetésre jutottam, hogy a készítendő alkalmazásnak a perzisztencia (adatok állandósága), a többszálúság (adatok párhuzamos elérése) és a távoli elérhetőség kritériumainak feltétlen eleget kell tennie, ugyanakkor ezek megvalósításának pontos részleteivel már nem kívánok foglalkozni. Ezért én a Java Enterprise Edition (Java vállalati verziója, röviden Java EE), mint programozási platform mellett döntöttem. Ez ugyanis moduláris szoftverkomponensek segítségével, széleskörűen támogatja a többretegű, elosztott alkalmazások készítését, miközben a fejlesztő számára indifferens részleteket jótékonyan elrejt. (Péter B. et al. 2007), (Jendrock et al, 2014).

A perzisztencia biztosítását egy relációs adatbázissal kívántam megoldani, amit egy Oracle adatbázis-kezelő szoftverrel terveztem menedzselni. Eközben a weblapokhoz tartozó funkcionalitás megteremtését JavaServer Faces (röviden JSF) technológia használatával gondoltam, amely egy a Java EE platform részét képező, komponens-orientált webprogramozási keretrendszer. (Péter B. et al. 2007), (Jendrock et al, 2014).

5. Programtervezés

A megrendelői elvárások és a technológiai lehetőségek megismerése után sor kerülhetett a szoftverfejlesztés következő fázisára, vagyis a megvalósítás (implementálás) terveinek egyre részletesebb kidolgozására.

5.1. Tervek felvázolása

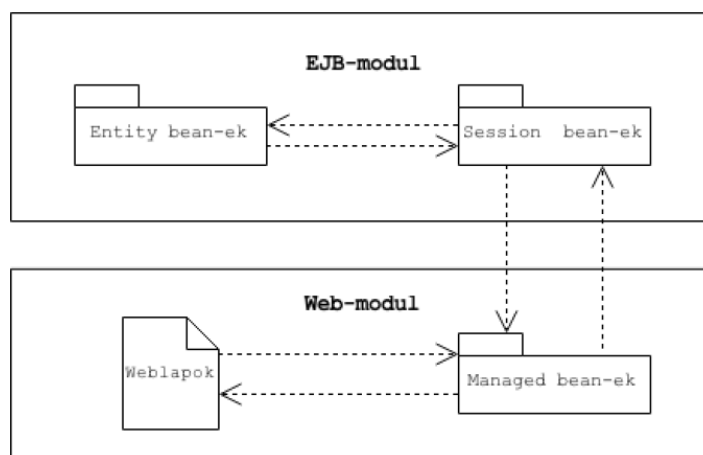
A konkrét tervezet kialakításának első lépéseként a szerkezeti elemeket igyekeztem meghatározni, vagyis a szükséges komponensek áttekintésével egy statikai modellt készíteni.

A szoftver vázát egy EJB- és egy webkonténer alkotja, amelyek strukturális szempontból modulnak is nevezhetők; az első egy jar, míg az utóbbi egy war kiterjesztésű állományként szerepel az alkalmazásszerverek fájlrendszerében.

Mindkét modul további részegységeket tartalmaz. Itt található az üzleti logikai réteg entity és session bean-jeinek, valamint a webréteg managed bean-jeinek csomagjai (package) is. Illetve a webmodulba kerül egy web nevű könyvtár is, amelyben a kliensréteg által megjelenített weblapok forráskódjai, azok stíluslapjai és az esetleges szkriptfájlok kerülnek elhelyezésre. Mindezen komponensek a szolgáltatások biztosítása érdekében természetesen egymás közötti kommunikációt folytatnak. (2. ábra)

Érdemes megfigyelni, hogy az entity bean-ek és a weblapok közvetlenül sosem kerülnek kapcsolatba. Így bármelyikük továbbfejlesztése, esetleges cseréje a másik működőképességére nincs hatással (a réteges architektúrák elvének megfelelően).

A tervből szándékosan maradt ki a message-driven bean-ek csomagja, ugyanis jelen specifikációban aszinkron üzenetkezelésre még nem lesz szükség.



2. ábra: Alkotóelemek diagramja

5.2. Tervek finomítása

A szerkezeti elemek felvázolása után a pontos részletek kidolgozásába fogtam a programterv későbbi implementációjának megkönnyítése, valamint a koncepcionális hibák és hiányosságok felderítése érdekében.

5.2.1. Adatbázis tervezése

A specifikáció során már sikerült meghatározni, hogy a kitűzött feladat megvalósításához milyen adattárolási igényeknek kell megfelelni. Azonban az összetett vagy többértékű attribútumok tárolása, kezelése még átgondolást igényelt.

A megrendelő eredeti elképzelése szerint az ügyfelek és az ügynökök állandó lakcímei, levelezési címei vagy cégek esetén a székhelyük adatai egy `String` típusú (szöveges) állományban kerültek volna eltárolásra. Azonban a későbbi módosíthatóság, valamint a keresési, lekérdezési lehetőségek támogatása érdekében célszerűbbnek tartottam ezen adatokat részekre bontva, egy-egy külön attribútumba helyezni. Ráadásul e címekből egy ügyfél vagy ügynök esetén egyszerre több is előfordulhat. Így ezen adatok tárolására egy újabb adattáblát (Címek) vezettem be, melynek elsődleges kulcsával kívántam biztosítani a kapcsolatot az Ügyfél, illetve az Ügynök táblával.

Azonban a név attribútum esetén ezt a megoldást eltúlzottnak találtam. Ezért a többértékű keresztnév használata helyett egyszerűen bevezettem az Ügyfél és Ügynök táblába egy újabb attribútumot, a középső nevet. Illetve hasonlóképp jártam el a többértékű telefonszám esetén is.

Döntésemet jelentős részben az is indokolta, hogy a követelmények között az egyszerűsített adatbevitel kiemelt fontossággal szerepelt. Ezért a másodlagos adatok megadását (cím, név, telefonszám) csupán lehetőségként kell biztosítanom. De a használatukra valószínűleg majd csak ritkán kerül sor.

Az adatok áttanulmányozása során arra is felfigyeltem, hogy az ügyfelek és a jogi személyiségű ügyfelek (cégek) kapcsolattartó személyei lényegében csak az ügynökhöz való viszonyukban különböznek egymástól. Természetes, hogy egy kapcsolattartó esetén elegendő volna a nevének és elérhetőségének tárolása is. Azonban a későbbiek során – más ügyből kifolyólag – még belőle is lehet egy újabb ügyfél. Így a majdani (redundáns) adatrögzítés lerövidíthető volna, ha a kapcsolattartók adatai is eleve az Ügyfél táblában kerülnének letárolásra. Ilyen értelemben tehát az ügynökök egy egységes felületen keresztül tudnák igazgatni a már meglévő és a még csak kilátásban lévő ügyfelek adatait. Ezáltal a szoftver az ügyfélkörük bővítéséhez is jelentős segítséget nyújthatna.

Ezen gondolatmenetet még tovább folytatva felmerült, hogy a cégek esetén nem csak a kapcsolattartók adatait lenne célszerű eltárolni. Hanem lehetőséget kéne biztosítani a megismert (egyéb) munkatársak felvételére is. Egyrészt a fent említett, praktikussági okok

miatt. Másrészt a hatékonyabb marketing kedvéért. A legjobb reklám ugyanis szájhagyomány útján terjed. Ezért amennyiben az alkalmazás a meglévő és leendő ügyfelek munkatársi viszonyait is tudná szemléltetni, akkor a biztosítási ügynökök sokkal célirányosabban tudnák a klientúrájukat bővíteni.

Az adatokat egészen más szempögből vizsgálva feltűnt, hogy az adatbevitel során sokszor egy előre definiált listából kéne az egyes elemeket kiválasztani. Ezért az alacsonyabb tárhelyigény érdekében célszerűnek tartottam a listák elemei helyett csupán az elemek kezdőbetűit tárolni (az egyértelmű azonosíthatóság elvének betartása mellett).

5.2.2. Szolgáltatások összehangolása

A követelmények részletes kidolgozásával világossá vált, hogy az alkalmazásnak milyen alapvető szolgáltatásokat kell nyújtania a felhasználói felé. Ezek döntő többségénél azonban nem egymástól elszeparált funkciók megvalósításáról lesz szó, hanem egymással összedolgozó, tevékenységükben összehangolt szolgáltatásokról. Éppen ezért érdemesnek tartottam megvizsgálni az együttműködésük mikéntjét is.

5.2.3. Adatok megjelenítési terve

Véleményem szerint úgy lehet a lehető legtöbb, hasznos információt elhelyezni a szoftver kimenetűl szolgáló weblapon (az egyszerűség és átláthatóság követelményeinek betartása mellett), ha az adatok megjelenítése egy hierarchikus rendszerben egymásra épül.

5.2.4. Navigációs terv

Az adatok megjelenítésével kapcsolatos egyeztetés során újabb nézőpontok kerültek napvilágra. Ugyanis az ügyfélközpontú információközlésen túl a napi feladatok könnyed áttekintése is praktikus volna. Illetve a későbbiekben, a már rendeltetésszerűen használt alkalmazás a biztosítási ügynökök teljesítményéről is szolgáltatathatna összesített adatokat. Ebből következően újabb igények kerültek megfogalmazásra, és így a követelmények leírása is az alábbiak szerint bővült:

- A lekérdezési szisztéma alapvetően háromféle séma szerint épüljön fel:
 1. Ügyfelek és a hozzá kapcsolódó adatok megjelenítése.
 2. Aktuális feladatok és az azokhoz kapcsolódó ügyfeladatok szemléltetése.
 3. Ügynökök és a hozzájuk tartozó tevékenységek adatainak lekérdezése.

- Be lehessen állítani, hogy a három séma közül melyik legyen alapértelmezetten az adott ügynök számára.

Az igény kielégítése érdekében felkerül a weboldalra egy navigációs panel, amin keresztül elérhető lesz a háromféle szisztéma, vagyis az Ügyfél, a Napi és a további fejlesztést igénylő Vezetői nézet.

6. Implementáció

6.1. Perzisztencia biztosítása

A technológiai háttér feltérképezése, majd a tervek felvázolása, pontosítása után elérkezett az idő mindezen ismeretanyag konkrét formába öntésére, melynél az eddig bevált módszer szerint igyekeztem az alapoktól, a tárolandó puszta adatoktól kezdve kiindulni.

Ennek értelmében előbb SQL (Structured Query Language – strukturált lekérdező) nyelven megfogalmazott konfigurációs fájlokat készítettem, amelyek futtatása által az adatbázisban létrejöttek a táblák, azok attribútumai, megszorításai. Illetve helyenként alapértelmezett (default) értékeket is megadtam egy-egy tulajdonság kapcsán, hogy az adatrögzítést végző ügynökök munkáját ezzel is elősegítsem.

6.1.1. Vállalati Entity bean-ek létrehozása

Mivel lényegében véve az entity bean-ek szavatolják a kapcsolatot a szigorúan vett adatréteg és a funkcionalitásért felelős üzleti logikai réteg között, ezért munkálkodásomat feltétlen ezek megteremtésével kellett folytatnom.

Minden egyedhalmaz számára egy-egy osztályt hoztam létre, amelyben deklaráltam a szükséges tagváltozókat, és az azok elérését biztosító getter – setter metódusokat.

Például a `@Table`-lel meghatároztam, hogy ez az osztály az adatbázis mely táblára képződjön le az alábbiak szerint:

```
@Entity
@Table(name = "CLIENT")
public class Client implements Serializable {
```

6.2. Funkcionalitás életre keltése

6.2.1. Managed bean-ek teremtése

Bár az adatok minden információ-rendszer esetén látszólag kiemelt jelentőséggel bírnak, valójában a rajtuk végzett műveletek és a táalásuk módja érdemel igazán figyelmet. A pusztán adat ugyanis különböző kontextusba helyezve más-más jelentéssel kaphat.

Az efféle műveleteket pedig (többek közt) a managed bean-ekben elhelyezett metódusok végzik. Ezért következő lépcsőként ennek megalkotásába fogtam. Vagyis létrehoztam egy menedzselte osztályt a webkonténeren belül, amely a vezérlő és modell feladatkörét egyaránt el fogja látni.

Annotációk segítségével a bean-t elneveztem, ezen a néven regisztráltam, majd a hatókörét nézet típusúra állítottam. Innentől fogva ez a konténer által menedzseltek tekinthető.

```
@ManagedBean(name = "managedClient")
@ViewScoped
public class ManagedClient implements Serializable{
```

6.2.2. Metódusok megalkotása

A vezérlő logika vázának megteremtése után még azt fel kellett tölteni tartalommal, vagyis az egyes szolgáltatásokhoz tartozó metódusokat meg kellett valósítani. Ennek folyamatát az Ügyfél-nézethez tartozó, az ügyféllista és űrlap menedzselését végző bean működésének bemutatásával prezentálom. Az aktuális ügyfelet lekérdező metódus az alábbi módon néz ki:

```
public List<Client> listClients() throws IOException {
    clients = clientFacade.findAll(filter);
    if (clients != null && !clients.isEmpty() && client == null)
    { client = clients.get(0);
    clientFacade.setClient(client);
    } return clients; }
```

6.2.3. Session Bean-ek felépítése

A managed bean-ek megalkotása során számtalan alkalommal hagyatkoztam az entitások adatbázisból lekért adataira. Csakhogy ezek eléréséhez, életciklusaik kezeléséhez a session bean-ek használata szükséges. Ennek értelmében entitásonként létrehoztam egy-egy ilyen „köztes” osztályt az alábbiak szerint:

```
@Stateless
public class AddressFacade extends AbstractFacade<Address> {
    @PersistenceContext(unitName = "AIS-PU")
    private EntityManager em;
```

6.2.4. Lekérdezések végrehajtása

A menedzselt bean-ekben már létrehoztam azon lekérdező metódusokat, amelyek a tagváltozókba mentve biztosítják a megjelenítési réteg számára szükséges adatok elérését. Ők maguk a session bean-ek közvetítésével férnek hozzá az entitások attribútumaihoz. Csakhogy a session bean-ek már a Java Persistence Query Language (a Java perzisztens lekérdezőnyelve, röviden JPQL) nyelven fogalmazzák meg a kéréseiket.

6.3. Felület megalkotása

Miután megbirkóztam a szoftver lelkét képező üzleti logikai réteg legfőbb problémáival, a felhasználó szempontjából legfontosabb rész, a megjelenítendő weboldalak felépítésébe kezdtem. Ám minden építkezés az alapozással kezdődik, ezért előbb a weblapok kezelésére hivatott szervletet kellett beállítanom.

6.3.1. Szervlet beállítása

A kéréseket kiszolgáló szervlet és az adott weboldal közötti kapcsolatot a `web.xml`-nek nevezett konfigurációs fájl biztosítja. Vagyis a webszerver az ebben foglaltak alapján találja meg „az adott URL-hez tartozó szervletet”. Ezért az alábbiak szerint regisztráltam a `Faces Servlet` nevű szervletet, majd beállítottam, hogy az `xhtml` kiterjesztésű állományaimat ő dolgozza fel.

6.3.2. Adatok konvertálása és validálása

A szoftver működésének elengedhetetlen feltétele, hogy a tárolandó ügyfelek, ügyek és feladatok adatait az ügynökök adhassák meg a weblapokon elhelyezett, beviteli komponensek segítségével. Csakhogy ezen attribútumok nem csupán szöveges információt tartalmazhatnak, mivel sok esetben számok, dátumok tárolására is szükség volna. Viszont az adatbevitelre szolgáló mezők mindenképp szöveges (String) típusú adatokat továbbítanak a webszerver felé. Ezért e probléma megoldása érdekében a JSF beépített konvertereit célszerű alkalmazni.

6.3.3. Belső navigáció

Mivel a többféle lekérdezési szisztéma érdekében a webes felületet kétféle nézetre osztottam, így e weblapok közötti navigáció is kardinális kérdéssé lépett elő. Nem csupán statikus navigálási lehetőséget kellett biztosítanom a felhasználók számára (másik oldalra mutató linkek), hanem a funkcionalitásba ágyazott, dinamikus hivatkozásokat is célszerű volt elhelyezni (gombok).

6.3.4. Utolsó simítások

Bármilyen jól is működtek már ekkor a szoftver által biztosított szolgáltatások, az i-re a pontot a külsőnek kellett feltennie. Ennek érdekében HTML címkékhez, CSS stíluslapokhoz és egy aprócska JavaScript program használatához folyamodtam. Lubbers, P., Albers, B., Salim, F., & Pye, T. (2011)., Frain, B. (2012)

Lényeges ugyanis, hogy a JSF címkék sokasága közt az egyszerű HTML címkék is használhatóak. Sőt, valójában a JSF nyelvi elemei vannak HTML címkékbe ágyazva, mivel a kliens csak ez utóbbiakat képes megjeleníteni. Így én fieldset címkét használtam az űrlapok keretbe foglalására. CSS alkalmazásával pedig (többek közt) sikerült más-más színnel jelölve megkülönböztetnem a listákban szereplő páros és páratlan sorokat. Miközben a JavaScript fájl segítségével a listák rendezési lehetőségeiről gondoskodtam.

7. A program tesztelése

A tesztelési fázis első, rám eső felét részben már sikerült lefolytatnom. Természetesen már a kód fejlesztése során is teszteltem a program működését, de szisztematikus vizsgálatot még csak most végeztem.

Ezáltal többek közt teszteltem az Ügyfél nézet alábbi funkcióit:

- Új ügyfél felvételét, már meglévő megtekintését, módosítását, illetve törlését.
- Új ügy felvételét, már meglévő megtekintését, módosítását, lezárását, illetve törlését.
- A feladatlistában szereplő egyedek lezárását és törlését.
- Ügyfél, illetve ügy választása esetén a „kapcsolt” adatok megtekintését.
- Egy-egy lista minden adatának törlése esetén a rendszer választát.

Majd megvizsgáltam a Napi nézet alábbi szolgáltatásait:

- Új ügy felvételét, már meglévő megtekintését, módosítását, illetve törlését.

- Új feladat felvételét, már meglévő megtekintését, módosítását, illetve törlését.
- Napi feladat kiválasztása esetén a „kapcsolt” adatok megtekintését.
- A megtekintett ügyfél adatainak módosítását.
- Egy-egy lista minden adatának törlése esetén a rendszer választát.

Végezetül ellenőriztem az új ügynök regisztrációs lehetőségeit, és a nézetek közötti navigációt.

A teszteredmények kiértékelése után (és közben) javítottam a felmerülő hibákat, amelyek nagyrészt figyelmetlenségből fakadtak. A megoldásuk struktúrális változással nem járt.

8. Összegzés

A nem formális tanulás során elvégzett alkalmazásfejlesztési feladat eredményes lett, sikerült a felmerült problémákat megoldani, a tesztelési eredmények is pozitívak lettek. A tanulás ezen nem formális módja hatékonyan kiegészíti az iskolai keretek között tanultakat, azok alkalmazása terén további ismeretek elsajátítását alapozza meg.

A feladat során egy ügyfélkapcsolat kezelő rendszer kialakítása volt a cél biztosítási cégek ügynökei számára. A feladat a valós életből jött probléma megoldására adott lehetőséget, mely további előnye volt az önálló feladatnak. Összességében a bemutatott fejlesztéshez hasonló feladatok ötleteket adhatnak más diákok, oktatók számára is az önálló feladatok alkalmazásának minél szélesebb körben alkalmazottá tétele érdekében.

Irodalomjegyzék

Sagitova, R. (2014). Students' self-education: learning to learn across the lifespan. *Procedia-Social and Behavioral Sciences*, 152, 272-277.

Barbosa, J., Silva, A., Ferreira, M. A., & Severo, M. (2017). The impact of students and curriculum on self-study during clinical training in medical school: a multilevel approach. *BMC medical education*, 17(1), 9.

Péter B., et al. (2007). *Szoftverfejlesztés Java EE platformon*. Szak Kiadó Kft.

Antal B. (2012). *Az adatbázis-kezelés alapjai*. Buza Antal, Dunaújváros, 2012.

Sándor, S., & László, V. (2003). *Szoftvertechnológia és UML*. ELTE Eötvös K.

Kővári A. (2019). A felnőttoktatás 4.0 és az az ipar 4.0 kihívásai az életen át tartó tanulásban. *PEDACTA*, 9(1), 9–16.

Katona J et al. (2017). Examine the effect of different web-based media on human brain waves. In *2017 8th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*. 407-412

Lubbers, P., Albers, B., Salim, F., & Pye, T. (2011). *Pro HTML5 programming* (pp. 107-133). New York, NY, USA:: Apress.

Katona J et al (2014). Control of incoming calls by a windows phone based brain computer interface. In *2014 IEEE 15th International Symposium on Computational Intelligence and Informatics (CINTI)*. 121-125.

Jendrock, E., Cervera-Navarro, R., Evans, I., Haase, K., & Markito, W. (2014). *The java ee 7 tutorial* (Vol. 1). Addison-Wesley Professional.

Frain, B. (2012). *Responsive web design with HTML5 and CSS3*. Packt Publishing Ltd.

Váczy Zsuzsa (2012). Fejlesztési lehetőségek nem formális tanulási környezetben. *Könyv és Nevelés*. 2012(1).