

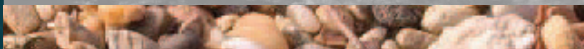
Dunaikavics

A Dunaújvárosi Egyetem online folyóirata 2021. IX. évfolyam VIII. szám

Műszaki-, Informatikai és Társadalomtudományok

HELENA MANOJLOVIC

A gesztusalapú technológia helye az oktatásban



ZAKOTA ZOLTÁN

Az Edison programozási nyelv és szerepe a konkurens programozásban



BARTHA MIKLÓS–ZAKOTA ZOLTÁN
A Forth programozási nyelv alkalmazási lehetőségeiről a számítástechnika hatékony oktatásában



GYÖRFFYNÉ HOLLÓ KRISZTINA

Az érintés nélküli adatgyűjtés kockázatai és a kockázatszámítás módszerei



DunaKavics

A Dunaújvárosi Egyetem online folyóirata 2021. IX. évfolyam VIII. szám

Műszaki-, Informatikai és Társadalomtudományok

MEGJELENIK ÉVENTE 12 ALKALOMMAL

SZERKESZTŐBIZOTTSÁG

András István, Bacsa-Bán Anetta, Balázs László,
Nagy Bálint, Németh István, Pázmán Judit, Rajcsányi-Molnár Mónika.

Felelős szerkesztő Németh István
Tördelés Duma Attila

Szerkesztőség és a kiadó címe 2400 Dunaújváros, Táncsics M. u. 1/a.

Kiadja DUE Press, a Dunaújvárosi Egyetem kiadója
Felelős kiadó Dr. habil András István, rektor



A lap megjelenését támogatta a Nemzeti Kulturális Alap

<http://dunakavics.due.hu>

ISSN 2064-5007

Tartalom

HELENA MANOJLOVIC

A gesztusalapú technológia helye az oktatásban 5

ZAKOTA ZOLTÁN

Az Edison programozási nyelv és szerepe a konkurens programozásban 27

BARTHA MIKLÓS-ZAKOTA ZOLTÁN

*A Forth programozási nyelv alkalmazási lehetőségeiről a számítás-
technika hatékony oktatásában* 45

GYÖRFFY NÉ HOLLÓ KRISZTINA

*Az érintés nélküli adatgyűjtés kockázatai és a kockázatszámítás
módszerei* 77

Galéria

(Duma Bálint fotói) 98



A gesztusalapú technológia helye az oktatásban

Összefoglalás: A gesztusvezérlés, mint feltörekvő technológia új lehetőségeket nyitott a tanulás számára. A gesztusalapú eszközök használata az újonnan megjelenő tanulási technológia (gesztusalapú tanulás) részeként olyan tudományterületekhez kapcsolódik, mint az oktatástechnológia, az oktatástervezés, az ember-számítógép-interakció, a gépi tanulás, a mesterséges intelligencia, a pszichológia, a tudományok tanulása, a pszichometria és a számítástechnika különféle ágai. Ma már létfontosságúak az innovatív interdiszciplináris kutatások és a gesztuson alapuló tanulási rendszereket dokumentáló publikációk. Általuk az oktatók jobban megérthetik, hogyan lehet a gesztusalapú technológiát megvalósítani és hatékonyan felhasználni a tanításhoz és a tanuláshoz különféle oktatási környezetben.

A kutatók és rendszerfejlesztők számára kihívást jelentő kérdés az, hogy a gesztusalapú technológia milyen előnyökkel járhat a tanulás szempontjából. Így e tanulmány egyik célja ennek a kérdésnek a feltárása, illetve a gesztusalapú tanulási módszerrel foglalkozó szakirodalom áttekintése a tanulási eredményekre gyakorolt hatás kontextusában.

Kulcsszavak: Gesztusalapú tanulási rendszer (GBLS); gesztusalapú eszköz; oktatás.

Abstract: Gesture-based devices as an emerging technology have opened up new opportunities for learning. The use of gesture-based enabled devices as part of emergent learning technology (gesture-based learning) is related to disciplines such as educational technology, instructional design, human computer interaction, machine learning, artificial intelligence, psychology, learning sciences, psychometrics, and various branches of computational engineering. Therefore, innovative cross-disciplinary research and related publications that document specific gesture-based learning systems and their

* Szabadkai Műszaki

Szakfőiskola

E-mail: helenam@vts.su.ac.rs

[1] Reinders, H. (2014): Touch and gesture-based language learning some possible avenues for research and classroom practice. 6.

[2] Információs és Kommunikációs Technológiák – az információ gyűjtésének, tárolásának, továbbításának és prezentálásának hardver-, szoftver- és média-feltételeit, legyen az információ formája hang, adat, szöveg, vagy kép. Magába foglalja a telefon, mobiltelefon, hardver, szoftver területét egészen az internetig (Detschew, 2008).

[3] Johnson, L.–Becker, S. A.–Witchey, H.–Cummins, M.–Estrada, V.–Free-man, A.–Ludgate, H. (2012): NMC Horizon Report: 2012 Museum Edition. *The New Media Consortium*. <https://www.learntechlib.org/p/182014/>

[4] Wojciechowski, A. (2012): Hand's poses recognition as a mean of communication with natural user interfaces. *Bull. Pol. Ac.: Tech.*, 60. Pp. 331–336. <https://doi.org/10.2478/v10175-012-0044-3>

[5] Ali Shakroum, M.–Wai-wong, K.–Chun Che Fung, L. (2016): The Effectiveness of the Gesture-Based Learning System (GBLS) and Its Impact on Learning Experience. *Journal of Information Technology Education: Research*, 15. Pp. 191–210. <https://doi.org/10.28945/3518>

associated designs are now vital. Once completed, educational researchers can better understand how gesture-based computing can be effectively utilized and implemented for teaching and learning across a wide array of educational settings.

However, how would a gesture-based technology benefit learning is a challenging question for novice researchers and system developers in education. Thus, the aim of this study is to explore this question.

Keywords: Gesture-Based Learning System; gesture-based device; education.

Bevezető

A digitális erőforrásokkal létrejövő interakció manapság az érintésre, a gesztusokra, és egyre inkább a szem mozgására épül. Számos hétköznapi szórakoztató elektronikai termék ma már tartalmaz érintés-alapú interfészeket: az e-könyvolvasóktól a táblagépeken át, a személyi számítógépektől kezdve az autó GPS-rendszerekig. [1] A kérdés az, hogy milyen előnyökkel járnak ezen új interakciós formák a tanítás szempontjából?

Az IKT-eszközöket [2] évek óta használják az oktatás területén. A modern oktatási korszakban a számítógépek minden osztályterem alapvető elemévé váltak. A gesztusalapú eszközök, mint feltörekvő technológia új lehetőségeket nyújtanak a tanulás terén. Sajátosságaik lehetővé teszik a felhasználó számára, hogy kontrollerként közvetlen kapcsolatba lépjen a számítógéppel mozdulatok és mozgások által, méghozzá oly természetes módon, mint ahogy azt a mindennapi életben teszik. [3] Az interakció a beszéd, a gesztus, ujj- és kézmozdulat, arckifejezés és a szemmozgatus útján valósulhat meg. [3]; [4]

Az elmúlt években számos kutatást végeztek a gesztusalapú tanulási rendszer (GBLS) használatának és előnyeinek kivizsgálására, viszont a tanulási eredményekre gyakorolt hatás vizsgálata még mindig kevés. [5] A pszichológiai tanulmányok rég kimutatták, hogy a gesztusok befolyásolják a tanulás módját, azt ahogyan gondolkodunk és

észleljük a világot. Nagyszámú tanulmány kimutatta, hogy a gesztusalapú tanulási rendszer módszere pozitívan befolyásolja a hallgatók teljesítményét, az élvezetes és interaktív tanulási környezet megteremtése által. [6]; [7]

Noha a kezdetben a gesztusalapú technológia nagy mértékben a játékokban és a mobileszközökben kapott helyet, a tanulási célok könnyebb és érdekesebb elérésének a lehetősége a közelmúltban óriási érdeklődést váltott ki az oktatók körében. [7] A technika alkalmazásai és fejlesztése az oktatás minden területén folyamatosan bővül. Ezekkel az eszközökkel végzett kutatások és az innovatív tanítási módszerek bevezetése számos területen növekszik, mint a gyógypedagógia, a fizika, a matematika, a fizikoterápia, a művészetek, a zene, a tudomány, az írástudás és a társadalmi fejlődés. [3] Az oktatáskutatók a gesztusalapú eszközök különböző hatásait vizsgálják: a természetes bemenet hatását, illetve azt, hogy milyen hatással lehetnek a tanulás más aspektusaira, például a memóriára. [8] Újszerűsége és széles körű alkalmazási lehetőségei miatt a gesztusalapú tanulás nagyon érdekes az oktatók, az ezzel foglalkozó szakemberek és a kutatók számára egyaránt. A gesztusalapú technológia az oktatás területén interdiszciplináris együttműködést és innovatív gondolkodást igényel.

A számítógéppel való együttműködés új módja

A közelmúltban az információval való interakció új módjai széles körben elérhetővé váltak számos technológiai fejlődés eredményeként. Az érintés ma már sok eszköz fontos jellemzője. A gomb vagy a képernyő megnyomását manapság már úgynevezett „haptikus visszajelzéssel” nyugtázzák (rövid rezgés jelzi a parancs beérkezését). Ezenkívül napjainkban gyakran alkalmazzák a többérintéses parancsokat (multi-touch) a telefonok, táblagépek és más eszközök manipulálására. Az ujjak csúsztatása az objektumok mozgatásához a képernyőn, a kicsinyítéshez vagy nagyításhoz, és a bonyolultabb többujjas érintéskombinációk példák az eszközökkel való kölcsönhatás meglehetősen újszerű módjai. Az újabb asztali számítógépek és laptopok egyre inkább rendelkeznek beépített érintőképernyővel a tartalom manipulálásának további módjaiként. [1]

[1] Reinders, H. (2014): Touch and gesture-based language learning some possible avenues for research and classroom practice. 6.

[3] Johnson, L.–Becker, S. A. –Witchey, H.–Cummins, M. –Estrada, V.–Freeman, A.–Ludgate, H. (2012): NMC Horizon Report: 2012 Museum Edition. *The New Media Consortium*. <https://www.learntechlib.org/p/182014/>

[6] Di Tore, S.–Elia, F.–Aiello, P.–Carlomagno, N.–Sibilio, M. (2012): Didactics, movement and technology: New frontiers of the human-machine interaction. *Journal of Human Sport and Exercise*, 7. <https://doi.org/10.4100/jhse.2012.7.Proc1.20>

[7] Sheu, F.-R.–Chen, N.-S. (2014): Taking a signal: A review of gesture-based computing research in education. *Computers–Education*, 78. Pp. 268–277. <https://doi.org/10.1016/j.compedu.2014.06.008>

[8] Chao, K.–Huang, H. –W.–Fang, W.-C.–Chen, N.-S. (2013): Embodied play to learn: Exploring Kinect-facilitated memory performance. *British Journal of Educational Technology*, 44. <https://doi.org/10.1111/bjet.12018>

[3] Johnson, L.–Becker, S. A. –Witchey, H.–Cummins, M. –Estrada, V.–Freeman, A.–Ludgate, H. (2012): NMC Horizon Report: 2012 Museum Edition. *The New Media Consortium*. <https://www.learntechlib.org/p/182014/>

[4] Wojciechowski, A. (2012): Hand's poses recognition as a mean of communication with natural user interfaces. *Bull. Pol. Ac.: Tech.*, 60. Pp. 331–336. <https://doi.org/10.2478/v10175-012-0044-3>

[9] Liu, Z. (2013): *Design a Natural User Interface for Gesture Recognition Application*. 22.

[10] Malizia, A.–Bellucci, A. (2012): The artificiality of natural user interfaces. *Communications of the ACM*, 55. (3.) Pp. 36–38. <https://doi.org/10.1145/2093548.2093563>

[11] Blake, J. (2011): *Natural User Interfaces in NET* [PDF]. <https://sciarium.com/file/55588/>

[12] Chen, N.-S.–Fang, W.-C. (2014): Gesture-Based Technologies for Enhancing Learning. In: *Lecture Notes–Educational Technology*. Pp. 95–112. https://doi.org/10.1007/978-3-642-38291-8_6

Az ember és a számítógép közötti interakció (human-computer interaction) két fő paradigmaváltáson ment keresztül az elmúlt három évtizedben: a parancssori interfészekről (Command Line Interfaces – CLIs) a grafikus felhasználói felületekig (Graphical User Interfaces – GUIs). A közelmúltban nyilvánvalóvá vált egy harmadik paradigmaváltás, amely a természetes felhasználói felületeken (Natural User Interfaces – NUI) alapul. [9] E felületeknek célja, hogy a felhasználók természetesebb módon kommunikálhassanak a számítógépes rendszerekkel, hasonlóan a való világban végzett tevékenységünkhöz [10]; [11] ezt olyan felhasználói felületként definiálta, melynek célja a meglévő készségek újrahasznosítása, a tartalommal való közvetlen interakcióhoz. Az IKT támogatja a természetes felhasználói interakciókat, abból a célból, hogy a hagyományos interfészek helyett, különféle interakciós formák épülhessenek be az eszközökbe, például multi-touch, szemkövetés, hang, gesztusok, kézírás stb. [4]

A gesztusalapú interakció (gesture-based interaction) egy kifejezetten gyorsan fejlődő terület. Más néven gesztusalapú számítástechnikaként (gesture-based computing) ismert, amely az emberi testet használja fel beviteli eszközként. [12] Lehetővé teszi a felhasználók számára, hogy sokféle eszközzel, például mobiltelefonokkal, táblagépekkel, játékkonzolokkal vagy számítógépes rendszerekkel lépjenek kapcsolatba. A kölcsönhatás például mozgásérzékelők, adatkesztyű vagy gravitációs érzékelőn keresztül valósul meg, természetes felhasználói felület által. A gesztusalapú technológia alkalmazása nem csak a test, hanem a kognitív dimenzióra gyakorolt lehetséges hatásra is összpontosít. [12]

Szóval, a gesztusalapú interakció egy testalapú interakció a számítógéppel, amely különböző érzékelők, például gravitációs, infravörös és strukturált fényű 3D-érzékelő használatával válik lehetővé, és a következő funkciókat biztosítja [3]: gesztus- és hangfelismerés; helyzet, gyorsulás és irány; vizuális jelzők olvasása; rázás, forgatás vagy mozgás reakció; lehetővé teszi, hogy a test és a kéz beviteli eszközökként működjenek. E funkciók által megváltozik a számítógéppel való interakció. A kézmozdulatok vezérlőparancsként történő használata növelheti az inger-válasz kompatibilitást, amely jobb érzéket adhat a felhasználónak a közvetlen manipulációhoz. A természetes felhasználói felület által a testi bevétel, beleértve a gesztusokat és a járást, felhasználható például egy avatár vezérlésére, ami azt a látszatot kelti, hogy a felhasználó személyesen

a virtuális környezetben jár. [12] A vezérlőbemenet és az ebből fakadó effektus leképezése intuitívabb, közvetlenebb és kompatibilisebb a természetes felhasználói felületen, mint a hagyományos interfészeknél.

Az *iPhone*, az *iPad*, a *Nintendo Wii*, a *Microsoft Xbox 360 Kinect technológia* és más gesztusalapú eszköz és szoftver fizikai mozgások formájában fogadják a jeleket, ideértve a kopogtatásokat, érintéseket vagy olyan mozdulatokat, amelyek lehetővé teszik a felhasználó számára a rendszer irányítását. 2019. januárjában a Szövetségi Kommunikációs Bizottság (the Federal Communications Commission) jóváhagyta a *Google Project Soli* érzékelő technológiáját, amely miniatűr radart használ az érintés nélküli gesztusok észlelésére. A jóváhagyást követően a Google gyorsabb ütemben fejlesztheti a gesztustechnológiát.

A gesztusvezérlő technika már megtalálható különféle alkalmazásokban: például a dél-afrikai *O. R. Tambo* nemzetközi repülőtérén (*O. R. Tambo International Airport*). Egy kávéipari vállalat olyan gépet telepített, amely elemzi az utazók arcmozdulatait és automatikusan kiad egy csésze kávé, amennyiben ásítást észlel valakinél. A *Samsung TV* támogatja a mozgásvezérlést a csatornaváltásnál, a játékoknál és az internet használatnál. A *Leap Motion* eszköze lehetővé teszi a felhasználó számára, hogy kézmozdulatokkal irányítsa a számítógépet vagy a laptopot. A *BMW* célja, a gesztusvezérlés által, a zavartalan vezetés elősegítése. A *BMW* technológiája innovatív, és biztosítja a vezető biztonságát és kényelmét. Egyszerű kézmozdulatokat használva különböző műveletek végrehajtása végezhető el: hívás fogadása vagy elutasítása, hangerő szabályozása, a visszapillantó kamera szög módosítása, illetve egyéni beállításokra is lehetőség van.

A mozgás egy olyan fejlemény, amely lehetővé teszi a felhasználók számára, hogy a telefonjukat irányítúként használják, vagy fejjel lefelé fordítsák az eszköz kikapcsolásához. A beépített gyorsulásmérők felismerik, azt amikor a felhasználók megdöntik telefonjukat, amivel a képernyőn lévő objektumokat mozgatva, növelhetik vagy csökkenthetik a hangerőt stb. [1]

A *Microsoft Kinect* technológiája például felismeri a játékosok mozgását, lehetővé téve számukra, hogy testüket felhasználva irányítsák a játék karaktereit és tárgyait. A gesztusvezérlés a mobileszközökbe is beépül, lehetővé téve a felhasználók számára, hogy a képernyőn keresztül integressenek, be vagy kikapcsolják a hangot, vagy webkamerákká váljanak. Az újabb fejlemények

[1] Reinders, H. (2014): Touch and gesture-based language learning some possible avenues for research and classroom practice. 6.

[12] Chen, N.-S.-Fang, W.-C. (2014): Gesture-Based Technologies for Enhancing Learning. In: *Lecture Notes-Educational Technology*. Pp. 95–112. https://doi.org/10.1007/978-3-642-38291-8_6

[1] Reinders, H. (2014): Touch and gesture-based language learning some possible avenues for research and classroom practice. 6.

[13] Alibali, M. W.–Nathan, M. J.–Wolfgram, M. S.–Church, R. B.–Jacobs, S. A.–Martinez, C. J.–Knuth, E. J. (2014). How Teachers Link Ideas in Mathematics Instruction Using Speech and Gesture: A Corpus Analysis. *Cognition and Instruction*, 32(1), 65–100. <https://doi.org/10.1080/07370008.2013.858161>

[14] Goldin-Meadow, S. (2017): Using our hands to change our minds. *Wiley Interdisciplinary Reviews: Cognitive Science*, 8. (1–2.) P. 368. <https://doi.org/10.1002/wcs.1368>

[15] Goldin-Meadow, S. (2005): *Hearing Gesture: How Our Hands Help Us Think* (New Ed edition). Belknap Press: An Imprint of Harvard University Press.

[16] Cook, S. W.–Duffy, R. G.–Fenn, K. M. (2013): Consolidation and Transfer of Learning After Observing Hand Gesture. *Child Development*, 84. (6.) Pp. 1863–1871.

[17] Keene, K. A.–Rasmussen, C.–Stephan, M. (2012): Gestures and a chain of signification: The case of equilibrium solutions. *Mathematics Education Research Journal*, 24. (3.) Pp. 347–369. <https://doi.org/10.1007/s13394-012-0054-3>

közé sorolandó, például a *Leap Motion* gesztusfelismerő rendszere, amely felismeri az egyes ujjmozdulatokat, és lehetővé teszi a gépelést vagy rajzolást a kéz levegőben tartásával a képernyő előtt. [1]

A tekintetkövetéses (*Eye-tracking*) kutatások azt mutatják, hogy lekövethetők a kognitív folyamatok hiszen követni lehet, hogy például milyen szempontokra figyelnek a felhasználók. Noha ez a technológia még mindig gyerekcipőben jár, a tekintetkövetés fejleményei jelentős ígéreteket mutatnak abban, hogy a felhasználók egyszerűen kezelhetik a képernyőn megjelenő tárgyakat úgy, hogy a szemüket egy adott pontra összpontosítják (<http://www.tobii.com>). [1]

A gesztusok bevitelén túl a gesztusalapú technológia gyakorlati tanulási lehetőségeket is kínál. Mivel a gesztusok és a testmozgás jelentik a fő szerepet, a technika lehetővé teheti vagy segítheti a fizikai tanulást, kezdve a rehabilitációtól, a testneveléstől és a fizikai erőnlétől. Ezenkívül a test bevonása a tanulási tevékenységbe megnövelheti a tapasztalati tanulást. A gesztusalapú technológia lehetővé teszi a hallgatók számára, hogy aktívan és fizikailag részt vegyenek a tanulásban.

A gesztus előnyei

Az emberek gesztusokat használnak, amikor nem tudnak szavakba önteni egy-egy kifejezést. A hallgatók is gyakran gesztusokkal kommunikálnak, melyekkel gondolataikat próbálják közvetíteni a tanárok felé. [13]; [14]. A gesztusok alternatív módjai a tudás és az ötletek kifejezésére [15], sőt a tudás befogadására is tanáraik gesztusaik által. [16] Alibali és munkatársai (2014) tanulmánya szerint a tanárok gesztusokkal erősíthetik oktatási feladataikat, ezzel segítve a hallgatóikat ötleteik összekapcsolásában és a bonyolult fogalmak egyszerűsítésében. Ezenkívül a gesztusok használata a tanulás során elősegíti a hallgatók jobb tanulását a tananyag megértése és a problémamegoldás szempontjából. [17] Folyt néhány kísérlet a gesztusok értékének vizsgálatára a tanulás során, ahol összehasonlították azokat a hallgatókat, akik a gesztusokat alkalmazták, miközben új koncepciót tanultak, olyan csoporttal ahol

kizárólag a kommunikációt alkalmazták. E kísérletek eredménye is rámutatott a gesztusok előnyére, hiszen bizonyítottan több tudást sajátított el a csoport, amelyben a gesztus is célzott alkalmazásra került. [18]; [19] Ebből kifolyólag a hallgatók ösztönzése a gesztusok használatára a tanulás során pozitívan befolyásolhatja tanulási eredményeiket.

A gesztusalapú tanulás

A Kinect-érzékelőt 2010 novemberében adta ki a Microsoft az Xbox 360 játékkonzol részeként, amely példa az első vezérlő nélküli videojáték-konzolra. A „Te vagy a vezérlő” szlogennel hirdették. [20] A Kinect alkalmazását a tanulásban különböző fogalmakkal definiálták, mint például Kinect az oktatásban, természetes felhasználói felület (*natural user interface*), gesztusos felületek a tanulásban és a Kinect által vezérelt tanulás. [21]; [22]; [23] A Microsoft mozgásérzékelő szenzorja lehetővé teszi a felhasználók számára, hogy fizikailag és kognitív intuitív módon vegyenek részt virtuális tevékenységekben. A felhasználók a testüket használják az egér és a billentyűzet helyett a feladat végrehajtására. [12] A Kinect számos fejlett természetes felhasználói felület funkciót is kínál, mint például a kimondott parancsok vagy bemutatott objektumok, arckövetés és beszédfelismerés. A Kinect-játékok oktatási előnyük mellett elősegítik a testi fejlődést, hiszen fizikai tevékenységet igényel a játszásuk. [24] Ezenkívül a Kinect-játékok olcsónak tekinthetők, ha összehasonlítjuk őket más tanulási technológiákkal [23], hiszen a Kinect-alapú tanulási alkalmazások többsége nyílt forráskódú és ingyenesen elérhető.

A gesztusalapú tanulási rendszer a Kinect felhasználói felületeit használja, hogy élvezhető és interaktív tanulási környezetet teremtsen bármely tanteremben [23]; [3] A rendszer olyan oktatási technológia, amelyet világszerte egyre több

[3] Johnson, L.–Becker, S. A.–Witchey, H.–Cummins, M.–Estrada, V.–Freeman, A.–Ludgate, H. (2012): NMC Horizon Report: 2012 Museum Edition. *The New Media Consortium*. <https://www.learntechlib.org/p/182014/>

[12] Chen, N.-S.–Fang, W.-C. (2014): Gesture-Based Technologies for Enhancing Learning. In: *Lecture Notes–Educational Technology*. Pp. 95–112.

[18] Broaders, S. C.–Cook, S. W.–Mitchell, Z.–Goldin-Meadow, S. (2007). Making children gesture brings out implicit knowledge and leads to learning. *Journal of Experimental Psychology. General*, 136. (4.) Pp. 539–550. <https://doi.org/10.1037/0096-3445.136.4.539>

[19] Cook, S.–Goldin-Meadow, S. (2006): The Role of Gesture in Learning: Do Children Use Their Hands to Change Their Minds? *Journal of Cognition and Development – J COGN DEV*, 7. Pp. 211–232. https://doi.org/10.1207/s15327647jcd0702_4

[20] Lang, S.–Block-Berlitz, M.–Rojas, R. (2012): *Sign Language Recognition Using Kinect*. P. 402. https://doi.org/10.1007/978-3-642-29347-4_46

[21] Chang, C.-Y.–Chien, Y.-T.–Chiang, C.-Y.–Lin, M.-C.–Lai, H.-C. (2013): Embodying gesture-based multimedia to improve learning. *British Journal of Educational Technology*, 44. (1.) E5–E9. <https://doi.org/10.1111/j.1467-8535.2012.01311.x>

[22] Evans, M. (2012): *Gestural Interfaces in Learning*. Pp. 3337–3340. <https://www.learntechlib.org/primary/p/40102/>

[23] Hsu, J. (2011): The Potential of Kinect as Interactive Educational Technology. *2nd International Conference on Education and Management Technology*, 13.

- [5] Ali Shakroum, M.–Wai-wong, K.–Chun Che Fung, L. (2016): The Effectiveness of the Gesture-Based Learning System (GBLS) and Its Impact on Learning Experience. *Journal of Information Technology Education: Research*, 15. Pp. 191–210.
- [7] Sheu, F.-R.–Chen, N.-S. (2014): Taking a signal: A review of gesture-based computing research in education. *Computers–Education*, 78. Pp. 268–277.
- [8] Chao, K.–Huang, H.-W.–Fang, W.-C.–Chen, N.-S. (2013): Embodied play to learn: Exploring Kinect-facilitated memory performance. *British Journal of Educational Technology*, 44. <https://doi.org/10.1111/bjet.12018>
- [22] Evans, M. (2012): *Gestural Interfaces in Learning*. Pp. 3337–3340.
- [23] Hsu, J. (2011): The Potential of Kinect as Interactive Educational Technology. *2nd International Conference on Education and Management Technology*, 13.
- [24] Ali Shakroum, M.–Wong, K. W.–Fung, C. C. (2018): The influence of Gesture-Based Learning System (GBLS) on Learning Outcomes. *Computers–Education*, 117. Pp. 75–101.
- [25] Chang, Y.-J.–Chen, S.-F.–Huang, J.-D. (2011): A Kinect-based system for physical rehabilitation: A pilot study for young adults with motor disabilities. *Research in Developmental Disabilities*, 32. (6.) Pp. 2566–2570.
- [26] Vernadakis, N.–Gioftsidou, A.–Antoniou, P.–Ioannidis, D.–Gian-nousi, M. (2012): The impact of Nintendo Wii to physical education students' balance compared to the traditional approaches. *Computers–Education*, 59. (2.) Pp. 196–205.
- [27] Chang, W., Fang, W.-C., Lin, Y.-L.,–Chen, N.-S. (2014): Gesture-Facilitated Learning of English Word Stress Patterns. *2014 International Conference of Educational Innovation through Technology*, Pp. 37–42.
- [28] Kuo, F.-R.–Hsu, C.-C.–Fang, W.-C.–Chen, N.-S. (2014): The effects of Embodiment-based TPR approach on student English vocabulary learning achievement, retention and acceptance. *Journal of King Saud University – Computer and Information Sciences*, 26. (1.) Pp. 63–70.

oktatási rendszer alkalmaz. A kifejezést a gesztusalapú felhasználói felület, az oktatási és tanulási eszköz használatának leírására alkalmazzák. [5] E rendszerek alkalmazása előnyös lehet mind a tanárok, mind a tanulók számára [23], hiszen az interakciós felület serkentheti a hallgatói motivációt úgy, hogy leköti a hallgatók figyelmét. Ezek a multimodális rendszerek megkönnyítik a kineziológiai kölcsönhatásokat és összehangolják azokat a hallási és a vizuális információkkal. Ennek a három különböző bemeneti módnak az összehangolása teszi a gesztusalapú tanulási rendszert kiváló eszközzé mind a vizuális, mind az auditív, de különösen a kinesztetikus tanulási stílusú hallgatók számára. [24]

A gesztusalapú technológiával, mint különböző oktatási célokat lehetővé tevő vagy segítő technológiával, számos tanulmány foglalkozott: tantermi oktatás [21], memórianövelő [8], fizikai rehabilitáció/terápia [25], testnevelés [26], és nyelvtanulás. [27]; [28]

A gesztusalapú tanulási rendszerek számos előnyt jelenthetnek az osztályteremben, mivel egyedülálló interakciós módszert kínálnak a tananyaggal [22]. Ezen felül az eszközök használatának egyik legfőbb oka a megfizethetőség és számítógépes programozási tudás nélkül is alkalmazhatók. [7] Az esetek többségében igen kis mértékű módosítást kell végrehajtania a pedagógusnak, hogy használni tudják az eszközt a tanórán.

Néhány tanulmány, amely komplex kognitív és procedurális típusú tanulást igénylő tárgyakat tartalmaz bemutatja, hogy a tanulási rendszerek testre szabhatók vagy megtervezhetők úgy, hogy kifejezetten a tanulás igényei és céljai kielégítését szolgálják. [7] Ide sorolandók az orvosi képzésben alkalmazott eszközök (például a mellvizsgálati eljárás), az építészet tervezési elemeinek feltárása vagy a gyártáshoz kapcsolódó összeszerelés/szétzerelés.

Egy másik érdekes csoport a tanúláshoz szükséges (digitális) kézzelfogható tárgyakból áll. Az ezekben a tanulmányokban használt és leírt tanulási rendszereket speciális igények és tanulási célok kielégítésére hozták létre. Például egy óriási fog és fogkefe a gyerekek fogmosási szokásra való tanítására [29], valamint különféle tárgyak, amelyek a gyermekek cselekedeteire reagálnak a dallamok és tempó tanulására a zenetanulás során [30], vagy a speciális oktatásban alkalmazott „Wii-terápia”.

Habár az oktatási technológiák eltérőek lehetnek, ugyanazt az elképzelést erősítik: a testmozgáson alapuló tanulást (*embodiment-based learning*), amely alatt az érthető, hogy a testmozgás formálja a megismerést, illetve hogy a megismerés a külvilággal kölcsönhatásban álló testek révén fejlődik ki. Több kutatás is bizonyította a megtestesült interakciók lehetséges előnyeit a tanulás és a testi fejlődés szempontjából. [31]; [32]; [33]; [34]

A gesztusalapú tanulási rendszerek tervezését és fejlesztését körbe ölelő tanulmányok területei eltérőek. Ide tartoznak az orvostudomány, a zene, az autizmus, mint speciális oktatási forma, a feldolgozóipar, az építészet stb. [7] A gesztusalapú eszközök közé tartozik az adatkesztyű, mozgáskövető, haptikus visszacsatolás és virtuális valóság (VR). Az építészetben az eszközök tartalmaznak érzékelőket, címkéket, kamerákat, projektorokat stb. Az orvosi képzésben az elsődleges eszközök közé tartoznak a fizikai mellszimulátorok és a VR.

[7] Sheu, F.-R.–Chen, N.-S. (2014): Taking a signal: A review of gesture-based computing research in education. *Computers–Education*, 78. Pp. 268–277.

[29] Sylla, C.–Branco, P.–Coutinho, C.–Coquet, E. (2011): TUIs vs. GUIs: Comparing the learning potential with preschoolers. *Personal and Ubiquitous Computing*, 16. Pp. 421–432. <https://doi.org/10.1007/s00779-011-0407-z>

[30] Bakker, S.–Antle, A. N.–Van Den Hoven, E. (2012): Embodied metaphors in tangible interaction design. *Personal and Ubiquitous Computing*, 16. (4.) Pp. 433–449. <https://doi.org/10.1007/s00779-011-0410-4>

[31] Marshall, P. (2007). Do tangible interfaces enhance learning? *Proceedings of the 1st International Conference on Tangible and Embedded Interaction*, Pp. 163–170. <https://doi.org/10.1145/1226969.1227004>

[32] Mazalek, A.–Van den Hoven, E. (2009). Framing tangible interaction frameworks. *AI EDAM*, 23, 225–235. <https://doi.org/10.1017/S0890060409000201>

[33] O’Malley, C.–Fraser, D. S. (2004): *Literature Review in Learning with Tangible Technologies*. <https://researchportal.bath.ac.uk/en/publications/literature-review-in-learning-with-tangible-technologies>

[34] Zuckerman, O.–Arida, S.–Resnick, M. (2005): Extending Tangible Interfaces for Education: Digital Montessori-Inspired Manipulatives. In *CHI 2005: Technology, Safety, Community: Conference Proceedings—Conference on Human Factors in Computing Systems*. <https://doi.org/10.1145/1054972.1055093>

[21] Chang, C.-Y.–Chien, Y.-T.–Chiang, C.-Y.–Lin, M.-C.–Lai, H.-C. (2013): Embodying gesture-based multimedia to improve learning. *British Journal of Educational Technology*, 44. (1.) E5–E9.

[23] Hsu, J. (2011): The Potential of Kinect as Interactive Educational Technology. *2nd International Conference on Education and Management Technology*, 13.

[35] Burns, C.–Myhill, D. (2004): Interactive or inactive? A consideration of the nature of interaction in whole class teaching. *Cambridge Journal of Education*, 34. (1.) Pp. 35–49.

[36] Kain, D. (2003): Teacher-Centered versus Student-Centered: Balancing Constraint and Theory in the Composition Classroom. *Pedagogy: Critical Approaches to Teaching Literature, Language, Composition, and Culture*, 3.

[37] Brush, T.–Saye, J. (2001): The Use of Embedded Scaffolds with Hypermedia-Supported Student-Centered Learning. *Journal of Educational Multimedia and Hypermedia*, 10. Pp. 333–356.

Tanulási eredmények

Az, hogy milyen hatással vannak az ilyen típusú interakciók az oktatásra bizonyos mértékig még ismeretlen terület.

A gesztusalapú tanulási rendszerek stimuláló eszközként működhetnek, ami hozzájárulhat a hallgatói motiváció fokozásához, mivel a rendszer egyedülálló és természetes interakciós módszert alkalmaz. A multimodálisnak köszönhetően, amely képes megkönnyíteni a kinesztetikus interakciókat és összehangolni azokat a hallási és a vizuális információkkal. Az e három különböző bemeneti-modalitás összehangolása miatt a rendszer kiváló tanulási módszer, amely támogatja a különböző tanulási stílussal rendelkező hallgatókat [23]. Még egy nagy előnye a fizikai fejlődéssel kapcsolatos. O'Hanlon (2007) tanulmánya kimutatta, hogy a gesztusalapú tanulási módszer segítheti a túlsúlyos hallgatókat azáltal, hogy fizikai tevékenységet, aktivitást igényel. Egyes kutatások szerint a hallgatók jobban tanulnak amennyiben a kognitív feladatokat fizikai mozgásokkal kombinálják. [21]

Interaktivitás

Az osztálytermi interaktivitás a tanárok által az osztálytermi tevékenységek felett gyakorolt kontroll szintjével mérhető. Más szavakkal, az osztálytermi interaktivitás a tanár által alkalmazott módszertantól függ; legyen szó tanár- vagy tanulóközpontú tanulásról. [35] A tanárközpontú pedagógia a hagyományos tanulási módszert képviseli, ezzel szemben a tanulóközpontú tanulás több lehetőséget kínál a hallgatói részvételre, az ismeretek tartalmának elemzésére és rendszerezésére. [36] A hallgatók közötti interaktivitás elmélyíti a tudást, elősegítheti a kreativitást, esélyt ad az osztálytermi tevékenységekben való aktív részvételre, és számos hallgatói puha készség fejlesztésére szolgál, mint például a problémamegoldás, a kritikus gondolkodás, és a kommunikáció. [37]

A tanulóközpontú tanulás a konstruktivizmus tanuláselméletén alapszik, amely szerint a hallgatók tapasztalataik és az új ötletek kölcsönhatásából tanulnak. [36] Manapság egyre gyakoribb a technológiák jelenléte az

osztálytermekben a tanulóközpontú tanulási módszerek elősegítésére és az interaktivitás támogatására. A gesztusalapú tanulási rendszerek alkalmazása elősegítheti az interaktivitást az osztályteremben. [38] A rendszer egyszerre több felhasználót képes fogadni, amivel ösztönzi a kollaborációt. [23]

Multimodalitás

A multimodalitás az oktatás területen olyan tanulási környezetre utal, amely megkönnyíti a tananyagelemek bemutatását több mint egy-szenzoros módszerrel (vizuális, auditív, hangzásbéli, írásbeli és kinesztetikus). [39] A tananyagok különféle módon történő bemutatása általában a tanulói figyelem felkeltését eredményezi. [39] A multimodális tanulási környezet elsődleges előnye, hogy lehetővé teszi a hallgatók számára a tanulást a számukra megfelelő módon. [40]

A gesztusalapú tanulási rendszere multimodális tanulási környezetet biztosít, amely szinte minden tanulási stílussal rendelkező egyén tanulását megkönnyítheti. Más, a kinesztetikus tanulási stílust figyelmen kívül hagyó tanulási technológiákkal ellentétben a gesztusalapú lehetővé teszi a felhasználók számára, hogy testmozgásuk segítségével kölcsönhatásba lépjenek a tananyaggal. [23]

A megtestesült kogníció

A megtestesült kogníció (*embodied cognition*) elméletben rejtőzik az alapja, a testünk elménkre gyakorolt hatásának magyarázata. [12] Az elmélet azt sugallja, hogy az eszközzel való manipulálás és az interakció megváltoztatja a gondolkodásmódunkat és megértésünket, és ez a változás hatással van a környezetünk érzékelésére. Más szavakkal, a megtestesültkogníció-elmélet azt állítja, hogy az emberek a testükkel megtanulják kiegészíteni az agyukat. [41]

[12] Chen, N.-S.–Fang, W.-C. (2014): Gesture-Based Technologies for Enhancing Learning. In: *Lecture Notes–Educational Technology*. Pp. 95–112.

[23] Hsu, J. (2011): The Potential of Kinect as Interactive Educational Technology. *2nd International Conference on Education and Management Technology*, 13.

[38] Homer, B. D.–Kinzer, C. K.–Plass, J. L.–Letourneau, S. M.–Hoffman, D.–Bromley, M.–Hayward, E. O.–Turkay, S.–Kornak, Y. (2014): Moved to learn: The effects of interactivity in a Kinect-based literacy game for beginning readers. *Computers–Education*, 74. Pp. 37–49. <https://doi.org/10.1016/j.compedu.2014.01.007>

[39] Sankey, M.–Gardiner, M. (2010): Engaging students through multimodal learning environments: The journey continues. *ASCILITE 2010 – The Australasian Society for Computers in Learning in Tertiary Education*.

[40] Picciano, A. G. (2009). Blending with Purpose: The Multimodal Model. *Journal of Asynchronous Learning Networks*, 13. (1.) Pp. 7–18.

[41] Kirsh, D. (2013): Embodied Cognition and the Magical Future of Interaction Design. *ACM Transactions on Computer-Human Interaction*, 20.

[42] Ratner, H. H.–Foley, M. A.–McCaskill, P. (2001): Understanding Children's Activity Memory: The Role of Outcomes. *Journal of Experimental Child Psychology*, 79. (2.) Pp. 162–191.

[43] Stevanoni, E.–Salmon, K. (2005): Giving Memory a Hand: Instructing Children to Gesture Enhances their Event Recall. *Journal of Nonverbal Behavior*, 29. Pp. 217–233.

[44] Barsalou, L. (2008): Grounded Cognition. *Annual Review of Psychology*, 59. Pp. 617–645.

[45] Kontra, C.–Goldin-Meadow, S.–Beilock, S. L. (2012): Embodied learning across the life span. *Topics in Cognitive Science*, 4. (4.) Pp. 731–739.

[46] Atkinson, D. (2010): Extended, Embodied Cognition and Second Language Acquisition. *Applied Linguistics*.

Néhány tanulmány hangsúlyozza a fizikai gesztusok és a kognitív feladatok kombinációjának pozitív hatását a tanulásra. [42]; [43] A megtestesült-kogníció-elmélet szerint az elménkre hatással lehet a cselekvési tapasztalat. Hosszú távon a cselekvési tapasztalatok hatással lehetnek a világról alkotott felfogásunkra. Rövid távon minden cselekvési tapasztalat továbbra is hatással van a tanulásra és a fejlődésre az egész életen át. Ez összhangban van a tapasztalati tanulás koncepciójával is, amelyet Dewey (1938) javasolt.

Mivel a gesztusalapú technológiák főleg gesztusokat vagy testmozgást foglalnak magukban, a megtestesült kogníciót széles körben alkalmazták a tanulásra gyakorolt fizikai hatás támogatására. A megtestesült kogníció az „észlelés, cselekvés, a test és a környezet közötti kölcsönhatásra” összpontosít [44], amely eltér a hagyományos perspektívától, ahol a testnek alig van szerepe a megismerésben. A megtestesült kogníciót feldolgozó tanulmányok a cselekvések különböző szerepét figyelték meg a kognitív folyamatokban, és azt sugallják, hogy az emberi elme szorosan kapcsolódik a szenzomotoros tapasztalatokhoz. [45]

A megtestesült (*embodied*) és kiterjedt (*extended*) kognitív elmélet [46] szerint a megismerés eleve kapcsolódik a testi érzésekhez, és a környezettel való kölcsönhatás kulcsfontosságú szerepet játszik a tanulásban, még az absztrakt kognitív feladatokban is.

A memória verbalizációja könnyebb, ha a tanulási testtartást vesszük fel a visszahívás során. A megtestesült interakció (*embodied interaction*) egyfajta szociokognitív cselekvés, amelyet az érintés és a gesztusos interfészek képesek támogatni.

Kuznetsov, Dey és Hudson (2009) hordható karkötőket alkalmaztak, amelyek különböző haptikus impulzus-aláírásokat térképeztek fel új koncepciókhoz. Amikor ezekkel a koncepciókkal később találkoztak, a megfelelő haptikus jelek visszajátszódtak, ami elősegítette a felidézést és a felismerést. A visszajelzések különösen a memóriazavarral küzdőknek nyújtottak segítséget.

A gesztusalapú eszközök alkalmazási köre az oktatásban

A kutatók és rendszerfejlesztők számára kihívást jelentő kérdés az, hogy a gesztusalapú technológia milyen előnyökkel járhat a tanulás szempontjából. A gesztusalapú eszközöket számos oktatási területen alkalmazzák, ilyenek a gyakorlati képzések, munkahelyi segítségnyújtás, segítő eszközök a mindennapi használatra, illetve rehabilitáció.

A gesztusalapú számítástechnika potenciálisan transzformatív technológia (*transformative technology*) lehet, mivel a tanulás médiumaként történő felhasználása megváltoztathatja attitűdjeinket és elősegíti az aktív tanulási módszereket.

A „Wii-terápia” eszközök és a hasonló termékek a fizikai és kognitív nehézségekkel küzdő tanulók számára óriási segítséget jelentenek a mindennapi motoros készségek, a konkrét munkakörülmények, a rehabilitáció és más készségek tekintetében. [47] Burke és munkatársai (2010) az *iPhone*-okat és más kiegészítő eszközöket olyan teljesítmény-támogató rendszeré alakították, amelyek elősegítik egy készségsorozat megtanulását az összetett feladatok egyszerűsítésével és jelzések biztosításával (*just-in-time* támogatás). Petersson és Brooks (2007) robotkészülékkel és kamerával olyan környezetet terveztek, amely lehetővé teszi a gyermekek (4–6 évesek) számára a játékkerápiát. Az *Exergame*-et a speciális oktatásban is alkalmazták a fizikai tevékenységek elősegítésére. [48] Ezek a tanulmányok azt mutatják, hogy a gesztusalapú eszközök ezen alkalmazásai pozitív hatással vannak a fizikai vagy kognitív nehézségekkel, illetve mindkettővel rendelkező hallgatók tanulási eredményeire. Az alkalmazott technológia elsődleges célja a teljesítménytámogatás azáltal, hogy különböző jelzésekre, azonnali visszajelzéssel (*just-in-time*) nyújtanak információt, amelyre a feladatok végrehajtása során szükség van. Egy másik előnye az adaptív és segítő technológia a tanuláshoz.

Érdemes észre venni, hogy a gesztusfelismerő technológia jóval több, mint amit a *Nintendo Wii* nyújt a testmozgás terén – a játékkörnyezetek olyan tevékenységeket népszerűsítene és fejlesztenek, amelyek javítják a szociális készségeket, magukban foglalják a csapatmunkát, és lehetővé teszik a felhasználók számára, hogy kollaboratív módon oldjanak meg problémákat. Mindez a tanulóközpontú tanítási módszer alkalmazását jelenti, ahol a hallgatón, és nem a tanáron van a fókusz. [49]

[47] Shih, C.-H.–Wang, S.-H.–Chang, M.-L. (2012): Enabling people with developmental disabilities to actively perform designated occupational activities according to simple instructions with a Nintendo Wii Remote Controller by controlling environmental stimulation. *Research in Developmental Disabilities*, 33. Pp. 1194–1199.

[48] Hourcade, J. P.–Bullcock-Rest, N. E.–Hansen, T. E. (2012): Multitouch tablet applications and activities to enhance the social skills of children with autism spectrum disorders. *Personal and Ubiquitous Computing*, 16. (2.) Pp. 157–168.

[49] Times, J. A. M. T. J. A. M. writes about technology for C. com S. also a contributor to the N. Y., Post, W., book, the G. H. most recent,–March, R. A. L. S. was published in. (2021): Google's Soli Uses Gesture Technology to Immerse K–12 Students. *Technology Solutions That Drive Education*. <https://edtechmagazine.com/k12/article/2019/08/googles-soli-uses-gesture-technology-immersed-k-12-students>

[7] Sheu, F.-R.–Chen, N.-S. (2014): Taking a signal: A review of gesture-based computing research in education. *Computers–Education*, 78. Pp. 268–277.

[50] Wheeler, M. (2010): Physics experiments with Nintendo Wii controllers. *Physics Education*, 46. P. 57.

[51] Barakova, E. I.–Lourens, T. (2010): Expressing and interpreting emotional movements in social games with robots. *Personal and Ubiquitous Computing*, 14. (5.) Pp. 457–467.

[52] Bekker, T.–Sturm, J.–Barakova, E. (2010): Design for social interaction through physical play in diverse contexts of use. *Personal and Ubiquitous Computing*, 14. (5.) Pp. 381–383.

A gesztusalapú technikai eszközöket a természettudományi és a matematikaoktatás terén is alkalmazzák, leginkább a fizikai, kémiai és matematikai tantárgyakhoz kapcsolódóan. [7] Ezek a szimulációt és a digitális manipulációt biztosítják a tanítás, tanulás folyamán. [50] Általánosságban a szimulációkból, a digitális manipulációkból és a kísérletekből tevődnek össze a gesztusalapú oktatási eszközök a természettudományos oktatás területén.

Az alkalmazások között megtalálható olyan is, amely az érzelmek tanulásában, a társadalmi viselkedésben és az együttműködés elősegítésében játszik nagy szerepet. [51]; [52] Tore és munkatársai (2012) a *Kinect*-et, a vizuális-motoros játékot, alkalmazták az ötödik osztályos tanulók integrációs képességek javítása érdekében. A *Kinect*-technológia önmagában is a tanulás egyik módja lehet. A gesztusalapú technológia olyan közegnek tekinthető, amelyben a hallgatók interaktív, aktív tanulási módon tanulhatnak, nem pedig egyszerűen a tananyagok lejátszásának vagy elérésének eszközeként jelenik meg.

Az idősebb tanulónál a „*Wii Fit*” alkalmazták Grieser és munkatársai (2012) az egyensúly gyakorlására. A technológia alkalmazásakor a testmozgás és a további tevékenységek automatikus naplózásra kerülnek, miután összehasonlíthatók. Az adatokat további oktatási stratégiákra is fel lehet használni, például teljesítmények bemutatására, játékos versenyek szervezésére a hallgatók között, és a gyakorlatok egyes részeinek a vizualizálására. [7] A gyógypedagógiai „*Wii-terápia*” és a testneveléshez kötődő „*Exergame*” között szoros párhuzam áll fenn. Ezek a gesztusalapú tanulási rendszerek vagy oktatási megközelítések hasonló okokból hatékonyak: megfizethetőségük és a használatukhoz és fenntartásukhoz szükséges alacsony technikai készségek igénye. A hallgatók ismerik az eszközöket, és kisebb módosításokkal vagy módosítások nélkül oktatási technológiává (vagy segítő technológiává) alakíthatók. Nincs szükség magas szintű programozási ismeretekre a rendszer futtatásához. Ezek az eszközök gyakran további motivációt és tanulói elköteleződést biztosítanak. Az iskolákban ezek a mozgásérzékelő eszközök hasznosak lehetnek a közös képernyőkön és együttműködési platformokon alapuló terekben.

Eddigi kutatások

Számos tanulmány szerint a gesztusalapú technológiák hozzájárulhatnak a hatékony tanulási környezet elősegítéséhez. [53]; [54]

Az elmúlt években nagyszámú empirikus tanulmány készült, amely a gesztusalapú tanulási rendszerek fejlesztésére és tesztelésére irányul. Különböző oktatási területeken használták, beleértve a fizikát, a matematikát, a zenét és a művészetet, a természettudományt, a társadalmi fejlődést és a fizioterápiát. [7] Ezenkívül néhány kutatás tanulmányozta a gesztusalapú tanulási rendszerek hatását az egyes tanulási szempontokra, például a memóriára és a fizikai rehabilitációra [8]; [47] A *KinectEducation* (<http://www.kinecteducation.com>) egy nem kereskedelmi platform a gesztusalapú tanulási rendszerek összes érintettje számára, beleértve a tanárokat, hallgatókat és szoftverfejlesztőket [55] [23] szerint a gesztusalapú tanulási rendszerekben nagy potenciál rejtőzik.

Úgy gondoljuk, hogy a közeljövőben ez a tanulási technológia az osztálytermek fókuszában fog állni, hiszen a *Kinect*-technológia számos olyan tulajdonsággal rendelkezik, amelyek alkalmassá teszik az oktatás területén történő alkalmazásra.

[5] Ali Shakroum, M.–Wai wong, K.–Chun Che Fung, L. (2016): The Effectiveness of the Gesture-Based Learning System (GBLS) and Its Impact on Learning Experience. *Journal of Information Technology Education: Research*, 15. Pp. 191–210.

[7] Sheu, F.-R.–Chen, N.-S. (2014): Taking a signal: A review of gesture-based computing research in education. *Computers–Education*, 78. Pp. 268–277.

[8] Chao, K.–Huang, H.-W.–Fang, W.-C.–Chen, N.-S. (2013): Embodied play to learn: Exploring Kinect-facilitated memory performance. *British Journal of Educational Technology*, 44.

[23] Hsu, J. (2011): The Potential of Kinect as Interactive Educational Technology. *2nd International Conference on Education and Management Technology*, 13.

[47] Shih, C.-H.–Wang, S.-H.–Chang, M.-L. (2012): Enabling people with developmental disabilities to actively perform designated occupational activities according to simple instructions with a Nintendo Wii Remote Controller by controlling environmental stimulation. *Research in Developmental Disabilities*, 33. 1194–1199.

[53] Beauchamp, G.,–Kennewell, S. (2010). Interactivity in the classroom and its impact on learning. *Computers–Education*, 54. (3.) Pp. 759–766.

[54] Hair, J., Hult, G. T. M., Ringle, C.,–Sarstedt, M. (2016). A Primer on Partial Least Squares Structural Equation Modeling.

[55] Kandroudi, M.,–Bratitsis, T. (2012). Exploring the educational perspectives of XBOX kinect based video games. *Proceedings of the European Conference on Games-Based Learning*.

[1] Reinders, H. (2014): Touch and gesture-based language learning some possible avenues for research and classroom practice. 6.

[5] Ali Shakroum, M.–Wai-wong, K.–Chun Che Fung, L. (2016): The Effectiveness of the Gesture-Based Learning System (GBLS) and Its Impact on Learning Experience. *Journal of Information Technology Education: Research*, 15. Pp. 191–210.

[21] Chang, C.-Y.–Chien, Y.-T.–Chiang, C.-Y.–Lin, M.-C.–Lai, H.-C. (2013): Embodying gesture-based multimedia to improve learning. *British Journal of Educational Technology*, 44. (1.) E5–E9.

[24] Ali Shakroum, M.–Wong, K. W.–Fung, C. C. (2018): The influence of Gesture-Based Learning System (GBLS) on Learning Outcomes. *Computers–Education*, 117. Pp. 75–101.

[56] Ayala, N. A. R.–Mendivil, E. G.–Salinas, P.–Rios, H. (2013). Kinesthetic Learning Applied to Mathematics Using Kinect. *Procedia Computer Science*, 25. Pp. 131–135.

[57] Hsiao, H.-S.–Chen, J.-C. (2016): Using a gesture interactive game-based learning approach to improve preschool children's learning performance and motor skills. *Computers–Education*, 95. Pp. 151–162.

[58] Ke, F.–Lee, S.–Xu, X. (2016): Teaching training in a mixed-reality integrated learning environment. *Computers in Human Behavior*, 62. 212–220.

[59] Meng, M.–Fallavollita, P.–Blum, T.–Eck, U.–Sandor, C., Weidert, S.–Waschke, J.–Navab, N. (2013): Kinect for Interactive AR Anatomy Learning. In 2013 IEEE International Symposium on Mixed and Augmented Reality. *ISMAR 2013*. P. 278.

[60] Sankaranarayanan, G.–Weghorst, S.–Sanner, M.–Gillet, A.–Olson, A. (2003): *Role of Haptics in Teaching Structural Molecular Biolog*. P. 366.

[61] Hamza Lup, F.–Adams, M. (2008): *Feel the Pressure: E-learning Systems with Haptic Feedback*. P. 450.

A gesztusalapú tanulási rendszerek hatékonyságáról több tanulmány számolt be. [5]; [56]; [21]; [57]; [58]; [59]

A rendszerek jellemzői (az interaktivitás és a multimodalitás) olyan oktatási tanulási környezetet hoznak létre, amely pozitívan befolyásolja a hallgatók belső motivációját, ami a tanulási eredmények javításához vezethet. [24]

A gesztusalapú interfészek használata tanulási módszerként megjelenik Chang és munkatársai (2013) tanulmányában, ahol a gesztusok és a testmozgás hatását tesztelték a tanulás során a *Kinect* segítségével. Az eredmények alapján a résztvevők jobb megértést és magasabb szintű információfeldolgozást mutattak a *Kinect*-alapú tanulási módszerrel. E mellett a hallgatók több információt tudtak felidézni a szokásos tanulási módszert alkalmazott hallgatókhoz képest.

A haptikus visszacsatolást is több helyen alkalmazták: például repülési szimulációnál [60], a fogászat és a középiskolai fizika órák területén. [61] Különösen ígéretes ez a technológia a visszajelzések tekintetében. Bebizonyosodott, hogy a gyakori visszajelzések jelentős hatással vannak a tanulásra, de a feladat teljesítése során adott visszajelzés zavaró lehet, akár egy tanár adja meg szóbeli észrevételekkel, akár egy számítógép a képernyőn megjelenő üzeneten keresztül. Egy erre a célra használt eszköz rövid rezgése az egyik módja annak, hogy például a válasz helytelen volt. Illetve haptikus visszajelzést lehet adni, valahányszor a hallgató megfelelő választ ad. [1]

Hwang és Cho (2012) ezt a technológiát használta az intonáció tanítására.

Wolski és Jagodziński (2019), a lengyel Adam Mickiewicz Egyetem kutatói, középiskolás tanulók körében vizsgálták a teljesítményt. A tanulók a *Microsoft Kinect* gesztustechnológiával felszerelt virtuális kémiai laboratóriumot használtak. A kutatók azt feltételezték, hogy a virtuális laboratóriumot használó tanulók jobban meg tudták oldani az összetett laboratóriumi feladatokat és, hogy a kézmozdulat-technika oktatási célú felhasználása megkönnyíti az együttműködést.

A gesztustechnika jövője biztosnak tűnik: a *Grand View Research* (<https://www.grandviewresearch.com/press-release/global-gesture-recognition-market>) becslései szerint a globális gesztusfelismerő piac 2025-re közel 31 milliárd dollárt fog elérni, szemben a 2017. évi 6,2 milliárd dollárral.

Metodológia

A tanulmány célja a gesztusalapú technológia szakirodalmának feltárása, különös tekintettel az oktatásra. E mellett a hallgatói vélemény feltérképezésére került sor a technológia ismeretéről, a tanulási eredményekre gyakorolt hatásáról és az esetleges fejlesztési szándékról. A felmérésben a Szabadkai Műszaki Szakfőiskola első és másodéves informatika szakos (Informatika, Műszaki informatika és Internet és elektronikus ügyvitel) hallgatói vettek részt.

ADATGYŰJTÉS ÉS ELEMZÉS

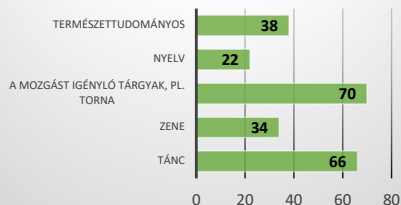
Az adatgyűjtés egy saját szerkesztésű kérdőív alkalmazásával valósult meg, amely online volt elérhető a hallgatók számára. A főiskola etikai szabályait betartva a részvétel teljesen önkéntes volt. A hallgatók tájékoztatást kaptak arról, hogy az online kutatásban az adatokat névtelenül gyűjtjük és, hogy személyes adatot nem rögzítünk. Továbbá, hogy a kutatás során nyert adatokat táblázatban név nélkül összegezzük, statisztikai elemzéseket végzünk rajta, amelyekből egyetlen résztvevő azonossága sem állapítható meg. A kérdőívet 91 első és másodéves hallgató töltötte ki: 15 lány és 76 fiú, ami reprezentatív mintának számít a Szabadkai Műszaki Szakfőiskola informatika szakos hallgatók vonatkozásában.

Eredmények

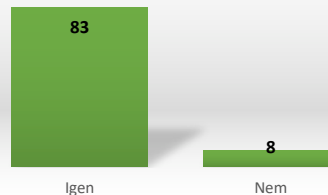
A kérdőívben szereplő kérdésekre adott válaszok eloszlását a diagramok szemléltetik.



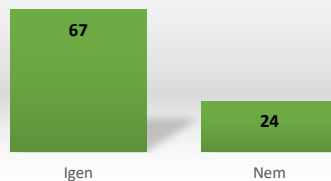
Van ötlete, milyen típusú tantárgyak integrálhatók a gesztusalapú tanulási rendszerekbe? (többet is kijelölhet)



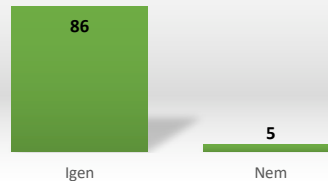
Felkeltette az érdeklődését a gesztusalapú technológia?



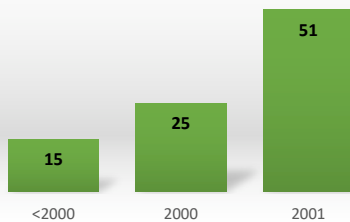
Amennyiben lehetőség adódna rá, részt venne a gesztusvezérelt technológia és eszközök fejlesztésében?



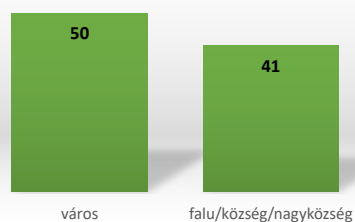
A kérdőív által sikerült-e valamelyest megismertetni Önnel e technológia fogalmát, lehetőségeit?



Születési éve



Lakóhelye?





Arra is kíváncsiak voltunk, hogy mik a hallgatók asszociációi a gesztusvezérlés fogalom hallatán. A válaszok között a szófelhőt felépítő fogalmak szerepeltek.

1. ábra. A gesztusvezérlés fogalomra adott hallgatói asszociációk



Forrás: Saját szerkesztés

Nem létezik szignifikáns eltérés a nemek vonatkoztatásában aszerint, hogy ismerik-e a gesztusalapú technológia fogalmát (Mann–Whitney, $U=580.5$; $p=0.795$). A faluban és a városban élők között sem mutatkozott meg különbség ebben a kérdésben (Mann–Whitney, $U=966.5$; $p=0.551$), illetve nincs szignifikáns eltérés a különböző korosztályoknál sem (Chi–Square=13.366; $p=0.1$).

A nemek vonatkozásában szignifikáns különbség mutatkozott meg a következő kérdésnél: Véleménye szerint a gesztusalapú tanulás hatással lehet a tanulási motivációra? (Mann–Whitney, $U=472.0$; $p=0.043$). Az összes megkérdezett lány pozitívan válaszolt a kérdésre.

A tanulási eredményességre gyakorolt hatásban szintén szignifikáns különbség mutatkozott a lányok és a fiúk válaszaiban (Mann–Whitney, $U=456.0$; $p=0.03$). A lányok nagyobb mértékben hiszik, hogy eredményesebb lehet a tanulás a technológia alkalmazása által.

A lakóhely vonatkozásában szignifikáns különbség van a városban és a faluban élő hallgatók között aszerint, hogy volt-e alkalmuk játszani gesztusvezérelt játékokkal (Mann–Whitney, $U=721.0$; $p=0.004$). A városban élők előnyét élveznek ebben a tekintetben.

A korosztály tekintetében egy kérdésben sem volt szignifikáns különbség a válaszok között.

Következtetés

A modern számítógépek felhasználói felületének fejlődése óhatatlanul eljutott arra a pontra, amelyen a felhasználó a főszereplő az ember–gép-interakcióban. Ez az interakció természetes és intuitív, a felhasználó mozgásán és hangján alapul, és előrelépést jelent a számítógéppel való kölcsönhatás megértésében. Kezdetben a mozgás- és hangfelismerési algoritmusok meglehetősen gyengék és pontatlanok voltak, de az idő előrehaladtával javultak, és ma már kielégítő a sikerarányuk.

Fontos felismerni, hogy e technológiák számos ága még gyerekcipőben jár; nincsenek követendő anyagok, meghatározott irányelvek vagy példák a bevált gyakorlatokra. Ebben az értelemben a mozgástechnológiák használata a tanítási kísérlet egyik formája. Természetesen nem minden iskola képes modern technológiát biztosítani a hallgatóinak. Azoknak azonban, akik képesek és hajlandók befektetni ilyen eszközökbe, a gesztusalapú technológia nagyon jó lehetőségeket kínál: fokozza az osztálytermi interakciót és lehetővé teszi a hallgatók számára, hogy lássák, megtanulják, megértsék és interakcióba lépjenek a környezettel, ezáltal interaktív digitális világot teremtsenek maguk körül.

A gesztusalapú eszköz oktatásban való használata kíváncsú, hiszen növelheti a hallgatók kreativitását és munkakedvét, illetve javíthatja a tanítási folyamatot és a tanulást. A gesztusok jellege nemcsak univerzális és természetesebb, mint az egér vagy a billentyűzet működtetése, de értékes eszköz lehet a hallgatók figyelmének fenntartásában és összpontosításában, valamint az interaktív tanterem népszerűsítésében. Az osztálytermi használat nagymértékben függ a jövőbeni szoftverektől és azok fejlesztési dinamikájától. Tartozunk annyival a következő generációnak, hogy megfelelően készítsük fel őket az egyre digitálisabb társadalomra.

Köszönetnyilvánítás

A közlemény megjelenését az EFOP-3.6.1-16-2016- 00003 „K+F+I folyamatok hosszú távú megerősítése a Dunaújvárosi Egyetemen” című projekt támogatta.

Az Edison programozási nyelv és szerepe a konkurens programozásban

Összefoglalás: A hetvenes évek egyik fő jellemzője a számítástechnikában az operációs rendszerek szerepének széleskörű elterjedése és fontosságának előre nem látott növekedése volt. Ennek oka egyrészt az alkalmazási területek gyors szaporodása, illetve azok komplexitásának növekedése, másrészt az integrált áramkörök technológiájának terén elért fejlődés volt. A mikroprocesszorok megjelenése megnyitotta az utat az elérhető áron hozzáférhető általános célú számítástechnikai rendszerek elterjedése előtt. Ezeknek a számítógéprendszereknek a hatékony kihasználásához pedig szükség volt új koncepciókon alapuló, megbízhatóan működő operációs rendszerekre, ami óriási hajtóerőt adott a konkurens programozás fejlődésének. A hetvenes-nyolcvanas évtizedekben számos új, e célból készült specializált programozási nyelv jelent meg. Jelen tanulmány célja bemutatni röviden az Edison programozási nyelvet, annak jelentőségét a konkurens programozásban, előnyeit és hátrányait, valamint hanyatlásának okait.

Kulcsszavak: Mikroprocesszorok megjelenése; Edison programozási nyelv.

Abstract: One of the main features of the 1970s in computing was the widespread use of operating systems and the unforeseen increase of their importance. This was partly due to the rapid proliferation of application fields and their increasing complexity, and partly due to the advances in integrated circuit technology. The advent of microprocessors has paved the way for the proliferation of affordable general-purpose computing systems, and the effective use of these computer systems required reliable operating systems based on new concepts, which gave a huge impetus to the development of concurrent programming. In the seventies and eighties, a number of new specialized programming languages developed for this purpose appeared. The aim of the present study is to briefly present the Edison programming language, its im-

* Partiumi Keresztény
Egyetem, Nagyvárad
E-mail: zzakota@gmail.com

[1] M. Kanellos:
„Moore says
nanoelectronics face
tough challenges,”
CNET, 10 03 2005.
[Online]. Available:
<https://www.cnet.com/news/moore-says-nanoelectronics-face-tough-challenges/>
[Hozzáférés dátuma:
09. 06. 2021.].

[2] B. Nichols,
„Programmer
Moneyball:
Challenging the
Myth of Individual
Programmer
Productivity,” Carne-
gie Mellon University
Software Engineering
Institute, 27. 01. 2020.
[Online]. Available:
<https://insights.sei.cmu.edu/blog/programmer-moneyball-challenging-the-myth-of-individual-programmer-productivity/>.
[Hozzáférés dátuma:
10. 06. 2021].

portance in concurrent programming, its advantages and disadvantages, as well as the reasons for its decline.

Keywords: The advent of microprocessors; Edison programming language.

Párhuzamosság, multiprogramozás, konkurencia

A hetvenes évek közepére szinte általánossá vált a határtalan optimizmus a számítástechnika fejlődését illetően, melyet mi sem fejezett ki jobban, mint a „Moore-törvény” terminus megalkotása és bevezetése Carver Mead számítógéptudós által, és annak gyors és kvázi-általános elfogadása. [1] Míg a törvény elég hosszú ideig jól alkalmazhatónak tűnt hardver téren, a szoftvert illetően már kevésbé vált be: a programozási költségek ára jóval kisebb ütemben csökkent, mint a hardver elemeké. Ennek egyrészt munkaerőpiaci, másrészt specifikus, a tevékenység jellegéhez kötött okai voltak.

A PROGRAMOZÁS HATÉKONYSÁGA

Nyilvánvaló, hogy a programozás hatékonyságát vagy termelékenységét nem lehet úgy növelni, mint egy számítógép komponensét, az integráció fokának növelésével. Programozók esetében a hatékonyság, bármely elfogadott meghatározását is vizsgáljuk, nem változik extrém mértékben. „Míg az átlagos teljesítmény különbözhet programozók között, a programfejlesztési erőfeszítések eltérésének csak fele tudható be az inherens programozói készségnek; a másik fele az egyes fejlesztők napi változásain belül van. Vagyis a programozók ugyanannyiban különböznek önmaguktól, mint a csoport többi tagjától.” [2]

Számos forrás egyetért abban, hogy az egy programozó által naponta megírt kódsorok száma nem változik jelentős mértékben a felhasznált nyelv típusának függvényében. A szóbanforgó időszakban, még nem álltak kellő mennyiségben rendelkezésre olyan szoftvereszközök, amelyekkel, legalábbis részben, automatizálni lehessen a programozási tevékenységet. Tehát, mindenképpen előnyös a magas szintű, erős utasításkészlettel rendelkező nyelvek használata. Mindez komoly motivációs tényezőként járult hozzá, hogy új magasszintű programozási nyelvek születhessenek egy relatíve rövid időszakon belül.

A hetvenes évekre, a számítógépek gyors technikai fejlődésének és a nagyméretű, komplex szoftverek iránti igény megjelenésének összjátékából előállt helyzetet nagyon jól illusztrálja E. W. Dijkstra egyik klasszikus szövege, amelyben felhívja a figyelmet egy új programozási filozófia kialakításának szükségességére. „Ha az embernek rendelkezésre áll egy nagyteljesítményű elektronikus számológép, azt ki is akarja használni. Ennélfogva a megoldandó problémák mérete alkalmazkodik a rendelkezésre álló berendezéshez [...] Fel kell ismernünk, hogy a programozás már ma is inkább a bonyolultság megszervezésének, a nagy méretekkel járó káosz lehető leghatékonyabb elkerülésének a művészete.” [3]

STRUKTURÁLT PROGRAMOZÁS

A megoldást a programozásnak erre a „krízisére” a strukturált programozási paradigma megjelenése jelentette. A kezdő momentumnak hagyományosan Dijkstra 1968-as, a *Communications of the ACM*-ben megjelent levelét tekintjük, melyben a goto-utasítás nyakló nélküli használatára hívja fel a figyelmet. [4] A strukturált programozás elmélete két nagyszerű könyvben lett bemutatva. Az első 1972-ben jelent meg, az O.-J. Dahl, E. W. Dijkstra és C. A. R. Hoare szerzőhármas nevével fémjelzett *Strukturált programozás* [5], magyarul [6]), melyből fennebb idéztem. Rá néhány évre, 1976-ban, következett egy másik alapmű, Niklaus Wirth *Algoritmusk + Adatstruktúrák = Programok* című műve [7], magyarul [8].

Utóbbi könyv jelentőségét az is növeli, hogy ebben kerül részletes bemutatásra alkalmazási szinten is a, szintén Wirth által kidolgozott, a strukturált programozás paradigmáját módszeresen implementáló Pascal programozási nyelv. [9] A Pascalt az Algol 60 alapján fejlesztették ki, de hozzá képest alkalmazhatósága jelentősen megnövekedett, a sokféle adatstruktúrálási lehetőség miatt. A nyelv megalkotásakor két fő szempontra voltak tekintettel: a programozás megtanulásának kényelmes alapjaként, illetve a nagy programok megírásának hatékony eszközeként történő felhasználásra. Ezért nagy hangsúlyt fektettek az alapfogalmak számának csekély voltára, az egyszerű és szisztematikus nyelvi struktúrára, valamint a hatékony megvalósíthatóságra. Ezen okok miatt, a Pascal valószínűleg minden idők legelegánsabb és egyik

[3] E. W. Dijkstra (1968): „A strukturált programozásról.” In: *Strukturált programozás*, Budapest: Műszaki.

[4] E. W. Dijkstra: „Letters to the editor: Go to statement considered harmful.” *Communications of the ACM*. 11. (3.) Pp. 147–148.

[5] O.-J. Dahl–E. W. Dijkstra–C. A. R. Hoare (1972): *Structured Programming*. London and New York: Academic Press.

[6] O.-J. Dahl–E. W. Dijkstra–C. A. R. (1978): *Hoare, Strukturált programozás*. Budapest: Műszaki.

[7] N. Wirth (1976): *Algorithms + Data Structures = Programs*. Englewood Cliffs, NJ: Prentice-Hall, Inc.

[8] N. Wirth (1982): *Algoritmusk + Adatstruktúrák = Programok*. Budapest: Műszaki.

[9] N. Wirth (1971): „The Programming Language Pascal,” *Acta Informatica*, 1. Pp. 35–63.

[10] N. Wirth (1977): „Toward a Discipline of Real-Time Programming,” *Communications of the ACM*, 8.

legnépszerűbb programozási nyelvéné vált. A CDC 6000 számítógépcsalád számára készítettek egy egylépcsős fordítót, amely teljes egészében Pascalban íródott. Szám-talan variánsát implementálták a legváltozatosabb számítógéprendszerekre. Számos programozási nyelvre volt meghatározó befolyással, amelyek között ott van az írás-ban később szereplő konkurens nyelvek nagy része is.

Megjelenésük után, körülbelül a hetvenes évekig, a számítógépprogramok szekvenciálisak voltak, vagyis a gép egy program utasításait a programozó által a programban meghatározott sorrendben hajtotta végre. A szekvenciális programok esetében, a determinisztikus jelleg prevalált, miszerint minden lépést pontosan, egyértelműen kell megfogalmazni hogy ugyanazon kiindulási adatokra tetszőleges számú végrehajtás esetén ugyanazt az eredményt szolgáltatassa. Hibamentes programok esetén nagy valószínűséggel lehetett számítani rá, hogy azonos bemenet mellett, többszöri futtatás esetén is ugyanúgy működnek. Egy szekvenciális program esetében, egy adott lépés végrehajtása csak aztán kezdődik el, hogy az őt megelőző lépés végrehajtása befejeződött.

PÁRHUZAMOSSÁG, KONKURENCIA, VALÓS IDEJŰ FELDOLGOZÁS

A megszakítással rendelkező számítógépek megjelenése után a programok egy részét már nem lehetett egyetlen szekvenciális utasítássornak tekinteni, mivel annak egyes részei egyidőben, vagy időben átlapolva, kerültek végrehajtásra. Az addigi szekvenciális programozás mellett kialakult a multiprogramozás is, amint azt Niklaus Wirth röviden összefoglalja egyik írásának elején: „A programok validálásához szükséges gondolatmenetek bonyolultabbá válásával, a programozás három fő kategóriára oszlik: szekvenciális programozás, multiprogramozás és valós idejű programozás. Egy szigorú programozási fegyelem betartásával és az e fegyelem alapján kialakított megfelelő magas szintű nyelv használatával drasztikusan csökkenthető a párhuzamosság és a végrehajtás időbeli korlátozásával kapcsolatos okfejtés bonyolultsága. Ez lehet az egyetlen gyakorlati módszer a valós idejű rendszerek analitikusan ellenőrizhetővé és végső soron megbízhatóvá tételéhez.” [10]

A multiprogramozás a párhuzamos feldolgozás egy kezdetleges formája, amelyben több program fut egyszerre egy uniprocesszor rendszerben. Mivel a rendszer csak egy processzort tartalmaz, a különböző programok, értelemszerűen, nem hajthatók egyidejűleg végre. Valójában, az operációs rendszer végrehajtja az egyik prog-

ram egy részét, majd egy másiknak egy részét és így tovább. Csak a felhasználó számára tűnik úgy, hogy az összes program egyszerre fut.

Ha egy multiprogram olyan részeket (folyamatokat, processzeket) is tartalmaz, amelyeknek futási idejére valamilyen korlátozó feltételnek is teljesülnie kell, akkor a feladat már a valós idejű (real-time) programozás körébe tartozik. A valós idejű, illetve multiprogramok közös tulajdonsága, hogy azonos bemenet esetén is működhetnek különböző módon és ezek egy része helyes, míg egy másik része hibás. Az ilyen programok működése általában nem determinisztikus, ami nagy mértékben megnehezíti a tervezésüket és ellenőrzésüket. Például tartalmazhatnak olyan hibákat, amelyek nem reprodukálhatók, és így szinte lehetetlen megtalálni azokat. Ezért, az ilyen programok esetében igyekezni kell olyan programozási nyelveket használni, amelyek már a szintaxis szintjén is tartalmaznak beépített biztonsági mechanizmusokat.

A multiprogramozás implementálásának egyik alapelve a konkurencia volt. Ezt a célt szolgálták a konkurens programozási nyelvek, amelyek a folyamatok vagy végrehajtási szálak egyidejű végrehajtásának koncepcióját használják egy program strukturalásának eszközeként. A másik alapelv a párhuzamosság. A párhuzamos nyelvek olyan programok megírására alkalmasak, amelyek több processzoron is futtathatóak. A konkurencia hasznos eszköz a párhuzamosság kifejezésében, de nem feltétlenül szükséges. A specifikus funkciók, mindkét esetben, a nyelv szintaxisának kell részét képezzék és nem holmi kiterjesztésnek, mint például egy könyvtárnak. Meg kell itt jegyezzük, hogy ez a megkülönböztetés konkurencia és párhuzamosság között inkább csak elméleti jellegű, mivel a szerzők nagy része is sokszor szinonimaként kezeli a két fogalmat.

Komoly lendületet adott a konkurens programozás fejlődésének az operációs rendszerek terén bekövetkezett előrelépés. A hetvenes évekre nyilvánvalóvá vált azok kitüntetett szerepe és fontossága a számítógéprendszerekben. Ennek oka egyrészt az alkalmazások területének kiterjesztése, illetve komplexitásának növekedése, másrészt az integrált áramkörök technológiájának terén elért látványos fejlődés volt. A mikroprocesszorok megjelenésével és széleskörű ipari felhasználásával megnyílt az út az elérhető áron hozzáférhető általános célú számítástechnikai rendszerek elterjedése előtt. Ezek hatékony kihasználásához pedig szükség volt új koncepciókon alapuló, megbízhatóan működő operációs rendszerekre. Mindez óriási hajtóerőt jelentett a konkurens programozás fejlődése számára, mind az elmélet, mind pedig a gyakorlat terén.

Minden operációs rendszer egyik alapvető problémája a párhuzamosan futó tevékenységek kontrollja. A gondot, egyrészt a rendszerkövetelmények, illetve célkitűzések, másrészt a hardver által megvont határok alkotják. Funkcionális szempontból tekintve, a rendszer a folyamat (processz) fogalmának segítségével írható le, ami egy a program végrehajtásához szükséges dinamikus entitás. Két vagy több folyamat konkurens végrehajtása feltételezi valamely kritikus mértékben rendelkezésre álló erőforrás megosztását. Ez megvalósítható exkluzív módon, szekvenciálisan használva azt egy-egy folyamat által, vagy több folyamat együttműködésével. Az utóbbi esetben szükség van a folyamatok közötti jel- és adatcserére, illetve a végrehajtásuk szinkronizálására, elkerülendő a nemkívánt kölcsönhatásokat.

A konkurencia alapelve nagyon egyszerűen megragadható: konkurencia = párhuzamosság + interakció. A párhuzamosság egymástól független programok/programrészek (folyamatok, processzek, taskok) időben (részben) egybeeső végrehajtását, míg az interakció a közöttük zajló (feltétlenül szükséges) kommunikációt jelenti. A konkurens kifejezés szerencsésen utal egyrészt arra, hogy a folyamatok versengenek egymással a különböző rendszererőforrások használatáért, másrészt pedig arra, hogy együttműködnek a közös feladat megoldásában.

A folyamat, mint a párhuzamos feldolgozás alapegysége, négy fontos tulajdonsággal rendelkezik:

- osztatlan: a folyamat, a konkurencia szempontjából elemi egységnek számít, nem tartalmaz olyan alegységeket/részeket, amelyek autonóm módon rendelkeznek különböző erőforrások felett;
- szekvenciális: egy adott folyamat végrehajtása tisztán szekvenciális, nem tartalmaz semminemű párhuzamosságot;
- aszinkron: a folyamatok relatíve független módon, egymással párhuzamosan futnak, kivéve a szinkronizálás során folyó kommunikációt;
- időszakos: egy folyamat csak a végrehajtásának megkezdése és annak befejezése közötti időtartam alatt létezik.

A folyamatok közötti interakciónak három alapvető formája van:

- kölcsönös kizárás: amikor egy folyamat egy adott erőforrás felett rendelkezik, más folyamatok nem interferálhatnak vele;
- szinkronizálás: a folyamatok közötti együttműködés időbeli feltételeinek szabályozása;
- egymás közti kommunikálás: folyamatok közötti üzenetcsere, amely tekinthető a szinkronizálás egy specifikus módjának is.

A konkurens programozási technikák óriási előrelépést jelentettek nem csak a modern operációs rendszerek megvalósításában, hanem a valós idejű rendszerek és impliciten a folyamatvezérlés kifejlesztésében is.

Konkurens programozási nyelvek

Az operációs rendszerek terén zajló kutató- és fejlesztőmunka számos konkurens programozási nyelv kidolgozását eredményezte a hetvenes évek folyamán. Ahogy az komplex fejlődési folyamatokban történni szokott, ezek némelyike sikeres volt, némelyike nem, egyesek utódnyelveket produkáltak, mások nem, egyesek széles körben ismertté váltak, míg mások csak szűk szakmai berkekben voltak használatosak. A legismertebb konkurens programozási nyelvek közé tartoznak a Concurrent Pascal, a Modula-2, vagy az Ada.

A szakirodalom megkülönböztet ugyan konkurens és valós idejű (real time) programozási nyelveket, de ez az osztályozás inkább csak elméleti jellegű lévén, hogy a gyakorlatban ezek nagymértékben fedik egymást. A valós idejű programozási nyelvek jórésze implementálja a konkurencia elveit és amelyik ezt

nem teszi meg, az is lehetővé kell tegye azok megvalósítását a gyakorlati kódolás során. A következő felsorolásban jómagam sem alkalmazom a szigorú elméleti kategorizálást, nem tartván azt célszerűnek a jelen kontextusban.

CONCURRENT PASCAL

A Concurrent Pascal (Konkurens Pascal) nyelvet Per Brinch Hansen alkotta meg 1972–75 között. Célja az volt vele, „[...] hogy az operációs rendszerek számára azt tegye, amit a szekvenciális Pascal a fordítók számára: a programozási erőfeszítés nagyságrenddel való csökkentését. A Concurrent Pascal kiterjeszti a szekvenciális Pascalt konkurens folyamatokkal és monitorokkal.” [11]

Ez az első célorientált megvalósítása egy konkurens programozási nyelvnek, amely messze meghaladta a konkurencia megvalósítására tett megelőző kísérleteket, mint amilyen a PL/I vagy az Algol 68 voltak. A szekvenciális Pascal egy megfelelően strukturált alapot képezett az új elvek biztonságos befogadására. A szintaxisát kibővítették szükséges új kategóriákkal úgy, mint az előzőekben említett folyamatokkal, illetve a Dijkstra által 1971-ben javasolt és Hoare [12], valamint Brinch Hansen által kifejlesztett monitorokkal [13] és a Dahl által 1967-ben, a Simula-ban bevezetett osztályokkal [14].

Bár a konkurens Pascal számos jó tulajdonsággal rendelkezett, amiknek egy részét a szekvenciális Pascaltól örökölte, másik részét pedig néhány akkoriban modernnek számító alapelv beépítéséből nyerte. Sajnos, a programozás terén lezajló gyors fejlődés miatt megjelenő újabb nyelvek hamar kiszorították és így nem válhatta be a hozzá fűzött reményeket. Maga Per Brinch Hansen, a megalkotója, is új konkurens nyelvvel rukkolt elő alig fél évtized múltán. Mégis, számítástechnikatörténeti jelentősége megkérdőjelezhetetlen marad.

MESA

A Mesa egy Algol-alapú nyelv, melyet eredetileg a Számítógépes Rendszerek Laboratóriumában (CSL), a Xerox Palo Alto-i Kutatóközpontjának egyik fiókjában terveztek meg az Alto, egy kísérleti mikrokódolt munkaállomás, számára. Használata, kezdetben, a PARC-ra és néhány olyan egyetemre korlátozódott, amelyeknek a Xe-

[11] P. Brinch Hansen (1977): *The architecture of concurrent programming*. New Jersey: Prentice-Hall Inc.

[12] C. A. R. Hoare (1974): „Monitors: An Operating System Structuring Concept,” *Communications of the ACM*, (10.) Pp. 549–557.

[13] P. Brinch Hansen (1973): *Operating System Principles*. NJ: Prentice-Hall Inc.

[14] O.-J. Dahl–B. Myhrhaug–K. Nygaard (1970): „*Common Base Language*,” Oslo.

[15] J. G. Mitchell –W. Maybury–R. Sweet (1979): „*Mesa Language Manual – Version 5.0*,” Palo Alto, CA.

[16] N. Wirth (2007): „Modula-2 and Oberon PowerPrint presentation,” In: HOPL III: Proceedings of the third ACM SIGPLAN conference on History of programming languages.

[17] N. Wirth (1977): „Modula: a Language for Modular Multi-programming,” *Software – Practice and Experience*, 7. Pp. 3–35.

rox adományozott néhány Altot. A Mesát később a Xerox kereskedelmi munkaállomásainak rendszerprogramozási nyelveként fogadták el.

A Mesa nyelv egy programozási rendszer eleme, amelynek célja a rendszerek és az alkalmazások széles skálájának fejlesztése és karbantartása. Különálló modulokból álló rendszerek fejlesztését támogatja, ellenőrzött információmegosztással. A nyelv olyan lehetőségeket foglal magában, mint a felhasználó által definiált adattípusok, típusok és interfészek szigorú ellenőrzését a kompiláció alatt, procedúra- és korutinvezérlő mechanizmusokat, valamint a konkurencia és a kivételek kezelésére szolgáló vezérlőstruktúrákat. [15]

Az Egyesült Államok Védelmi Minisztériuma megkereste a Xeroxot, hogy használhassa a Mesát az „IronMan” programozási nyelvéhez, de a Xerox elutasította. A Xerox PARC munkatársai azzal érveltek, hogy a Mesa olyan előnyt jelent a cég számára, amely produktívabbá teszi a Xerox szoftvermérnökeiket, mint más vállalatok mérnökei. A Védelmi Minisztérium ehelyett végül a jelöltek közül az Ada programozási nyelvet választotta és fejlesztette ki. 1976-ban, a Xerox PARC-ban, Niklaus Wirth megismerkedett a Mesával, amely nagy hatással volt Modula-2 nyelvének kialakítására, melyre úgy hivatkozik egyik alkalommal, mint a Pascal és a Mesa egy keverékére. [16]

MODULA

A Modula (MODULAr Programming Language – Moduláris Programozási Nyelv) elsősorban dedikált számítógépes rendszerek programozására szolgált, beleértve kisebb gépeken futó folyamatirányító rendszereket is. Nagyrészt a Pascalra épült, de bevezetve egy addicionális modulstruktúrát is. A modul olyan eljárások, adattípusok és változók halmaza, amelyben a programozó pontosan ellenőrzi a környezetből importált és az abba exportált neveket. A Modula tartalmaz általános multiproceszszálást megvalósító eszközöket, nevezetesen folyamatokat, interfész modulokat és jeleket. Lehetővé teszi ugyanakkor adott perifériás eszközök specifikálását is. [17] A nyelv fő célkitűzése az assembler-szintű programozás kiiktatása volt kisméretű rendszerek számára készült valós idejű alkalmazások esetében. Ugyanakkor kompakt, konkurens és fizikai szinten kezeli a ki- és bemeneteket.

MODULA-2

A Modula-2 nyelvet Niklaus Wirth alkotta meg 1977–80 között, a szintén általa készített Pascal és Modula nyelvekre alapozva. „Míg a Pascalt általános célú nyelvként tervezték meg és az 1970-es bevezetése után széles körben elterjedt, a Modula a multiprogramozással kapcsolatos kísérletekből született meg, ezért az adott alkalmazási terület releváns szempontjaira összpontosított. Kísérleti jelleggel 1975-ben definiálták és implementálták.” [18, P. 3] Első implementálása a PDP 11 minirendszerre történt, 1979-ben.

Az utódnyelv, a Modula-2, megpróbálja összeegyeztetni a rendszertervezés és a fizikai szintű programozás követelményeit, egybehangolva a Pascal és a Modula jellemzőit. A Pascal-hoz képest a modul és folyamat elvekkel bővül, valamint a procedúra típussal és gépi szintű programozási lehetőségekkel. A modularitásnak köszönhetően, bár a rendszerprogramozásban használatos, nagy mértékben megőrizte gépfüggetlenségét és a Pascalból örökölt átláthatóságát és egyszerűségét.

ADA

Az Ada egy majdnem 5 évig, 1974-től 1979-ig tartó többszakaszos szelekciós folyamat eredményeként született meg. A CII-Honeywell Bull-nál alkotta meg egy Jean D. Ichbiah által vezetett munkacsoport, amely 1980-ban publikálja a referencia kézikönyvet [19], mely szerint „[a]z Ada nyelv egy közös tervezési erőfeszítés eredménye, nagyméretű és valós idejű rendszerek programozására.” [19] „Az Ada-t három nyomós indok alapján tervezték: a programok megbízhatósága és karbantarthatósága fontosságának elismerése, a programozás, mint emberi tevékenység iránti odafigyelés és a hatékonyság.” [19] Az Ada-ra befolyással lévő elődnyelvek közül megemlíthendők a Pascal és rokonai (Euclid, Mesa, Modula), illetve az Algol 68 és a Simula.

A nyelv kifejlesztésével párhuzamosan zajlott a programozási közeg kialakítása is, APSE (Ada Programming Support Environment) néven. 1985-re létre is jött egy, de fordítóprogramból ekkora már 12 van validálva, többek között személyes számítógépekre is. Ugyanekkor már két nagyméretű Ada-ban megírt repülésirányító rendszer is működött, amelyek elokvensen bizonyították a nyelv erejét valós idejű környezetben.

[18] N. Wirth (1988): *Programming in Modula-2*, 4th (Szerk.): Springer-Verlag.

[19] US DoD (1980): „Ada Reference Manual,” In: *Ada, an introduction*, Springer Science+Business Media, LLC.

[20] H. Lienhard (1978): „The Real Time Programming Language PORTAL, Introduction and Survey,” *Landis-Gyr Review*, 25. Pp. 2–8.

[21] J. G. P. Barnes (1980): „The Standardization of RTL/2,” *Software-Practice and Experience*, 10. Pp. 707–719.

- Amint azt fennebb is idéztük, az Ada megalkotását három fő célkitűzés vezérelte:
- robusztusság és karbantarthatóság: átláthatóság, jóldefiniált típusok, külön kompilálás, kivételek kezelése;
 - a programozás hatékonysága: az alkalmazási területek változatosságához képest viszonylag kisméretű nyelv, moduláris koncepció, privát típusok, generikus egységek;
 - a programok hatékonysága: a kódos vagy drága megoldások elkerülése.

Megalkotói az Ada-t amolyan univerzális nyelvnek szánták, de az időközben bekövetkezett gyors technológiai változások (is) más irányba terelték a szoftverfejlesztést. Mindezek dacára tagadhatatlan, hogy a nyelv határkő a valós idejű programozás terén, főleg nagyméretű rendszerek tervezésében.

PORTAL

Megemlíthetjük még itt az 1978-ban publikált Portal nyelvet is [20], amelyet a svájci Landys-Gyr vállalatnál implementáltak az Eidgenössische Technische Hochschule Zürich-hel együttműködve, mind PDP 11, illetve NOVA 3 miniszámítógépes rendszerekre, mind pedig Intel 8080 mikroprocesszorra. A Portal képes összefogni egyedi programozók által tervezett részprogramokat oly módon, hogy azokból egy hierarchikus egységekből álló program keletkezzen, jól definiált interfészekkel. A nyelv magas szintű biztonságot képes nyújtani, magában foglalva párhuzamos számítási tevékenységek programozásának a lehetőségét, illetve több párhuzamosan működő program vezérlését is. Megvalósításakor különösképpen a valós idejű programozás követelményeit vették figyelembe, ezért a nyelv minden olyan alkalmazáshoz használható, amelyben az idő fontos szerepet játszik.

RTL/2

Egy szintén valós idejű programozási célokra készült nyelv az RTL/2 is, amint azt a neve is mutatja (Real Time Language 2). Ezt John G. P. Barnes tervezte az Imperial Chemical Industries (ICI) Ltd. számára 1971-ben, amit 1972-ben követett az implementálás ICI System 4 és PDP 11 rendszereken, majd 1974-ben külsőleg is hozzáférhetővé vált az SPL International által, végül 1980-ban vált brit szabvánnyá. [21]

Létrehozásának fő indokai egyrészt a valós idejű alkalmazások programozási és karbantartási költségeinek a csökkentése, másrészt a programozás alapelveinek életbe ültetése volt.

Az RTL/2, nevének dacára, nem tartalmaz a valós idejű programozásra jellemző specifikus utasításokat. A konkurenciát vagy a fizikai szintű ki- és bemenetet hagyományos módon, az operációs rendszerrel való interfészen keresztül kell megvalósítani. Ennek előnye a nagyfokú rugalmasság és az alacsony erőforrásigény. Ám ez ugyanakkor rontja a hordozhatóságot és a robusztusságot. Mindezek folyománaként, az RTL/2 egy kompakt és hatékony nyelv, mely megfelelő lehetőségeket nyújt a strukturált kódolás megvalósítására, ám kevésbé alkalmas az adatok strukturálására. Sajnos szekvenciális jellege nem teszi alkalmassá komplex valós idejű programok megírására.

Az RTL/2 előzménye egy RTL/1 nevű kísérleti prototípus volt. Elsődlegesen az Algol-68-ból származik, lecsökkentve annak hatalmas méretét és kiegészítve azt néhány hasznos tulajdonsággal, mint: moduláris programstruktúra, külön fordítás és hatékony lebegővesszős műveletek. Amint azt említettem, számos hiányossága van, de mindezek ellenére a kora hetvenes évek egyik legsikeresebb valós idejű és egyik legrepräsentatívabb Algol-alapú programozási nyelve.

FORTH

A FORTH programozási nyelv sem egy konkurens nyelv, a konkurencia megvalósítása adott esetben a programozóra lévén bízva. Charles Henry Moore, az arizonai Kitt Peak obszervatórium csillagásza alakította ki cirka egy évtized alatt, az 1960-as évek elejétől 1971-ig. Célja egy olyan könnyed, rugalmas és gyors nyelv megalkotása volt, amely ugyanakkor egy korlátlanul bővíthető utasításkészlettel is rendelkezik [22].

A nyelv bemutatása után következett Moore találkozása Elizabeth Ratherrel a US National Radio Astronomy Observatory-tól, akivel közösen megalakították a FORTH Inc. vállalatot [23], amit számos fejlesztés, verzió, illetve szabvány követett. 1976-tól kezdenek megjelenni a microFORTH fejlesztések Intel 8080, Motorola 6800 és Zilog Z80 mikroprocesszorokra, majd 1978-ban az első rezidens FORTH szoftver az Intel 8086 chipre. Több szabvány is születik, így a FORTH-79, a FORTH-83, majd az 1994-es ANS FORTH (ANSI) szabvány. 1984-ben megjelenik a MacFORTH, az első rezidens fejlesztés Apple Macintoshra.

[22] A. Seres-L. Fenyő-K. Gyalogh (1986): *A FORTH programozási nyelv*. Budapest: Műszaki.

[23] „*The Evolution of Forth*,” [Online]. Available: <http://www.forth.com/resources/evolution/index.html>. [Hozzáférés dátuma: 15. 06. 2021.].

[24] G. Lipovszí–
L. Subai–T. Beszeda
(1985): *FORTH
programozási rendszer
és nyelv*, Budapest: LSI
Alkalmazástechnikai
Tanácsadó Szolgálat.

[25] P. Brinch Han-
sen (1981): „Guest
Editorial: Introducing
the Edison Papers.”
*Software - Practice
and EXperience*, 1.
kötet 11. P. 323.

A Forthnak számos olyan jellemzője van, amely, ha nem is emeli a legnépszerűbb programozási nyelvek közé, de felhasználói táborán belül igen magas elégedettség forrása. Ezek közé a jellemzők közé tartozik a láncolt kódú interpreteren alapuló interaktivitás, illetve a kompilálhatóság. Fontos jellemzője, hogy túl az imperatív és procedurális jellegen, a nyelv szótárszerűen kiterjeszthető. Specifikuma, hogy ver-
mekre alapozott és a fordított lengyel jelölést használja, ami eléggé egzotikus jelle-
get kölcsönöz neki a programozási nyelvek között. A Forth strukturált, reflexív és
konkatenatív, de explicit nyelvtan és típusellenőrzés nélküli nyelv.

A nyelvnek számos előnye van, ami miatt igencsak népszerű a gyakorlott progra-
mozók körében [24]. Ilyen a teljes programozási környezet biztosítása, ami lehetővé
teszi a gyors szoftverfejlesztést, a bővíthetőség, a portabilitás, a kis erőforrásigény
és a gyors működés. Mivel könnyű interaktív kapcsolatot biztosítani a fejlesztővel
és a felhasználóval, ez nagy mértékben növeli a felhasználóbarátságát. Nagy elő-
nye még, hogy az előbb felsorolt tulajdonságokból kifolyólag, jól használható a
mikrorendszerek megvalósításában.

Mivel lehetővé teszi a multitaskingot, alkalmazási területei igen változatosak és ki-
terjedtek, jóval meghaladva a valós idejű alkalmazások korlátait. Ezek között olyano-
kat említhetünk meg, természetesen nem kizáró jelleggel, mint a szövegfeldolgozás,
adatbáziskezelés, robotvezérlés, folyamatirányítás, számítógépes grafika, oktatás. [24]

Az Edison konkurens programozási nyelv

A fennebb bemutatott jeles társaságba tartozik a, szintén Per Brinch Hansen által
megalkotott, Edison konkurens programozási nyelv is. Az Edisont 1978–80 között
definiálták, majd 1980-ban implementálták. A megrendelő a Mostek volt, amely sze-
retett volna egy multiprocesszor rendszerekre adaptált valós idejű nyelvet.

CÉLKITŰZÉSEK ÉS ALKALMAZÁS

A megrendelő által kitűzött cél bővült és ennek eredményeként egy elég erős nyelv
született ahhoz, hogy saját fordítóprogramjait, illetve egyszerű operációs rendszere-
ket is meg lehessen írni benne. A nyelv nem vezet be új gondolatokat, inkább a már
kipróbált eszközöket alkalmazza egy a Pascalnál egyszerűbb és a Concurrent Pascal-
nál erősebb nyelv létrehozására. [25]

Saját bevallása szerint, az Edison programozási nyelv megtervezésével, Brinch Hansen két fő célt követett:

- (1) egyszerűséget elérni, minek érdekében megkérdőjelezte számos jól ismert nyelvi fogalom szükségességét, ami fokozatosan oda vezetett, hogy kihagyott számos, a Pascal és a Concurrent Pascal programozási nyelvek értékes tulajdonságának tekintett nyelvi elemet (típusok, monitorok, folyamatok);
- (2) új betekintést nyerni a nyelvtervezésbe egy teljesen új megközelítéssel, amely szándékosan eltér attól a szellemtől, amelyben a Concurrent Pascalt megtervezték. [26]

PROGRAMSZERKEZET

Végül is az alkotónak sikerült célkitűzéseit megvalósítani és azok eredményeként az Edison nyelv egyszerű és kompakt nyelv lett, amelyben a konkurens programozás alapelvei úgy, mint modularitás, párhuzamosság és szinkronizálás elkülönülnek egymástól. A nyelv mechanizmusai általánosak és kevés megszorítást, ám ugyanakkor kevés könnyítést is tartalmaznak.

Egy Edison program állandók és típusok (opcionális) deklarációjából áll, melyet egy teljes procedúra deklaráció követ. Ez utóbbi fejrészből és törzsből áll, mely törzs szintén deklarációkból, valamint utasításokból áll. A teljes programok, a modulok és procedúrák mind blokkokként viselkednek. A blokkokat egymásba illesztve újabb blokkokat lehet alkotni.

A modul névvel ellátott entitások (konstansok, változók, procedúrák) egy csoportja, amelyek két kategóriába sorolhatók: helyi entitások, amelyek csak a modulon belül használhatók, illetve exportált entitások, amelyek mind a modulon belül, mind az azt közvetlenül körülvevő blokkon belül használhatók. A modul entitásai mindaddig léteznek, amíg a közvetlenül körülvevő blokk entitásai is. A modul arra szolgál, hogy helyi entitásait csak jól meghatározott procedúrák működtethessék, amelyeket a modulból a közvetlen környező blokkba exportálnak. Annak biztosítására, hogy a helyi változókat inicializálják használatuk előtt, egy modul tartalmaz egy utasításrészt, amelyet a modul entitások létrehozásakor hajtanak végre. [27]

Az Edison programok használhatnak könyvtárakat is, vagyis külön kompilált procedúrákból álló specifikus gyűjteményeket. Ezeket a procedúrákat a programból lehet meghívni a *lib* kulcsszó és egy könyvtárkulcs, egy implementálásfüggő szerkezet segítségével.

[26] P. Brinch Hansen (1981): „The Design of Edison,” *Software – Practice and Experience*, 4. Pp. 363–396.

[27] P. Brinch Hansen (1981): „Edison – a Multiprocessor Language,” *Software – Practice and Experience*, 4. Pp. 325–362.

[28] C. A. R. Hoare (1971): „Towards a Theory of Parallel Programming,” In: C. A. R. Hoare–R. H. Perrott (Szerk.): *Operating Systems Techniques*. Pp. 132–244.

[29] P. Brinch Hansen (1973): „Concurrent Programming Concept,” *Computing Surveys*, 5. (4.) Pp. 223–245.

TÍPUSOK ÉS ADATSTRUKTÚRÁK

Az Edisonbeli adattípusok hasonlóan a Pascaléra, de csak három predefiniált adattípus van a nyelvben: *int*, *char* és *bool*. A típusok diszjunktak, nincsenek a nyelvben altípusok, sem derivált (származtatott), sem pedig anonim típusok. Ez az egyszerűség kiiktatja a típusok ekvivalálásából származó ambiguitásokat és növeli a nyelv robusztusságát.

Lehetőség van összetett típusok definiálására is, így például zárójelek segítségével definiálhatók felsorolástípusok, halmazok, tömbök vagy rekordok. A tömbök csak egydimenziósak lehetnek, a rekordok pedig nem tartalmazhatnak variánsokat. Egy rekord különböző mezőit a Pascalból is jól ismert pont (.) operátorral lehet elérni.

Ami az operátorokat illeti, azoknak működése a Pascalbeli megfelelőikhez hasonlatos. Bevezeti viszont az Edison a *constructor* műveletet, amelynek segítségével két diszkrét típus értékei közötti megfeleltetést, gyakorlatilag egy szigorúan ellenőrzött egyszerű típuskonverziót, lehet megvalósítani. [28]

UTASÍTÁSOK ÉS VEZÉRLÉSI STRUKTÚRÁK

Az Edison hét utasítást tartalmaz, ebből kettő, az értékadás (asszignálás) és a procedúrahívás az általánosan elfogadottak. Van még egy üres utasítás is, a *skip*, végrehajtásának nincs semmi hatása és mindig végez. Az *if* és a *while* utasítások Pascalbeli megfelelőik általánosításai. A nyelv két specifikus, a valós idejű programozást lehetővé tevő, utasítást tartalmaz. Egyik a *when* szinkronizálási, másik pedig a *cobegin* konkurens utasítás.

A *when* szintaktikailag közel áll az *if* illetve *while* utasításokhoz, amennyiben mindhárom valamilyen utasításszekvenciát hajt végre adott logikai kifejezés igazságértékének függvényében. A *when* tulajdonképpen a Hoare által korábban javasolt feltételes kritikus szekció (conditional critical region) fogalom [28] általánosított megvalósítása [29]. Az utasításnak két fázisa van: a szinkronizálás és a tulajdonképeni végrehajtás. A szinkronizálás alatt a folyamat blokkolva van mindaddig, míg a feltétel nem teljesül, ami azt jelenti, hogy egyetlen más folyamat sem áll a kritikus szekcióban végrehajtás alatt. Mivel egy adott pillanatban csak egyetlen folyamat hajthatja végre a *when*-ben szereplő szinkronizált utasítást, a szinkronizálással egyszerre a kölcsönös kizárás is megvalósul.

A konkurens utasítás szerepe, hogy létrehozzon egy tetszőleges számú párhuzamos folyamatot és kezdeményezze azok végrehajtását. Az utasítás akkor fejeződik be, amikor az általa létrehozott összes folyamat befejeződik. A konkurens Pascaltól eltérően, az Edison megengedi a folyamatok dinamikus létrehozását és megsemmisítését, de azzal a kitételrel, hogy az egyidejűleg történik mindnek a számára.

[27] P. Brinch Hansen (1981): „Edison – a Multiprocessor Language,” *Software – Practice and Experience*, 4. Pp. 325–362.

ALKALMAZÁSI TERÜLETEK

Egyik tanulmányában, Per Brinch Hansen ezt írja az Edison lehetséges felhasználási területeiről: „[a] nyelv alkalmas mind a konkurens programozás elveinek tanítására, mind pedig megbízható valós idejű programok tervezésére egy többprocesszoros rendszer számára.” Ugyanott még ezt is olvashatjuk: „[a]z Edison kialakításának legfőbb gondja az egyszerűség elérése volt a közepes méretű, 1000–2000 soros szöveges programok tervezésében. Ez a cél néhány, a moduláris programozáshoz szükséges, elvont fogalom kiválasztásához vezetett. Különösen a modularitás, az egyidejűség és a szinkronizálás kívánalmait osztottuk szét különálló nyelvi jellemzőkre.” [27]

Ezek a célkitűzések, figyelembe véve azt is, hogy elsősorban olcsó, mikroprocesszor-alapú párhuzamos feldolgozást megvalósító rendszerek számára készült, bizonyos mértékben behatárolták a felhasználási területeit. Ez a nyelv relatíve alacsony robusztusságában is tükröződik. Mégis, az Edisont hatékonyan alkalmazták nem túl nagyméretű alkalmazások készítésére kisméretű mono- és multiprocesszor rendszerek esetében, operációs rendszerek vagy a saját programozási környezetek megírására.

ELŐNYÖK ÉS HÁTRÁNYOK

Az Edisont általában a konkurens Pascallal összevetve szokás minősíteni. Maga Per Brinch Hansen is így jár el, mikor róla ír. „Az Edison hozzáadott rugalmasságát (és bizonytalanságát) a Concurrent Pascalhoz képest egyesek visszalépésnek, mások kihívásnak fogják tekinteni. Számomra ez egyszerűen egy kísérlet, amely vagy megerősíti, vagy ellentmond annak a jelenlegi érzésemnek, hogy a programozási nyelvektől nem várható el a komplex absztrakciók támogatása, hanem ehelyett meglehetősen kényelmessé kell tenniük olyan programozási stílusok adoptálását, amelyek

[26] P. Brinch Hansen (1981): „The Design of Edison,” *Software – Practice and Experience*, 4. Pp. 363–396.

[30] P. Brinch Hansen (1982): *Programming a personal computer*. NJ: Prentice-Hall.

egyszerűbb fogalmakat használnak a bonyolultabbak felépítéséhez. Mondanom sem kell, hogy ez a programozási nyelv kialakítását egy kényes egyensúly megtalálásává teszi az asszemblerv nyelv anarchiája és a magasan specializált nyelvek kényszerzubonya között.” [26] Az Edison esetében, Brinch Hansen egyértelműen az asszemblerv rugalmassága és óhatatlanul, alacsony biztonsága irányába mozdította el az említett egyensúlyt.

A nyelv előnyei főként egyszerűségéből eredeztethetőek. Ezek között meg kell említeni a következőket:

- a fordítóprogram és a forráskód kis mérete;
- az adattípusok definiálásának és konvertálásának egyszerűsége;
- a komplex adatstruktúrák hiánya;
- az utasítások kis száma;
- a konkurencia implementálásának egyszerűsége;
- a könyvtárak használatának biztosítása;
- a nyelv adaptálása multiprocesszor rendszerekre.

A hátrányai nagyrészt ugyancsak az egyszerűségéből, illetve a fennebb említett programozási filozófiából fakadnak. Így például:

- a valós idejű programok megfelelő szintű megvalósításához szükséges robusztuság csak kevéssé fejlemezett programozási stílus alkalmazásával érhető el;
- a dinamikus adatstruktúrák hiánya;
- számos, a strukturált programozásból, megszokott utasítás hiánya;
- a meglévő utasítások időnként szokatlan formája;
- a dinamikus folyamatok egyidejű létrehozása és megszüntetése időnként zavaró;
- hiányzik a kivételek kezelésének mechanizmusa;
- a gépi szintű hozzáférések szegényes mivolta.

IMPLEMENTÁLÁS

Az Edison rendszert először 1980-ban implementálták egy PDP 11/23 típusú minirendszeren, majd rövid időn belül átvitték IBM személyi számítógépre is. [30] Az Edison-11 név alatt futó nyelvet olcsó mikroprocesszorok támogatására tervezték, ezért úgy döntöttek, hogy előre látva ezt a felhasználást figyelmen kívül hagyják a megszakítások kezelését a PDP 11-n, még gépi szinten is. [26]

Az Edisonnak számos implementálásáról tudunk, ám ezek főként kísérleti és/vagy oktatási jelleggel készültek. Maga Brinch Hansen is fontosnak tartotta az Edisonnak a párhuzamos programozás oktatásában való felhasználását. 1983 januárjában, a University of Southern California létrehozott egy 10 IBM személyi számítógépből álló labort, ahol a hallgatók az operációs rendszerek készítését tanulhatták és gyakorolhatták Edisonban. [31]

A '80-as években a varsói egyetemen is folyt egy érdekes kísérlet, melynek az volt a célja, hogy segítse a konkurens programozás elveinek tanítását. Az ott kialakított Edison-N rendszer egy Z80 mikroprocesszoron alapuló, Sinclair ZX Spectrum mikrokomputerekből álló hálózaton valósult meg (48 KB RAM-mal és 16 KB ROM-mal). [32] Az erlangeni ModulaWare GmbH-nál Guenter Dotzel implementálta 1982-ben, egy PDP-11/RT-11 rendszeren [33].

1981-ben, Magyarországon is megjelent egy komplex tanulmány a Magyar Tudományos Akadémia Számítástechnikai és Automatizálási Kutató Intézete égisze alatt, Böszörményi László tollából, „Multi-task rendszerek fejlesztése magasszintű nyelven” címmel, a *Tanulmányok* 128.-ik köteteként. [34] Ebben bemutatásra kerülnek, a párhuzamosság, multitasking, konkurencia és kapcsolódó alapfogalmak mellett a, fennebb már általam is említett, legfontosabb ilyen jellegű programozási nyelvek is. Ezek között megtalálható az Edisonnak is egy rövid leírása. [34]

Érdekes módon, a '80-as években, a romániai számítástechnika két jelentős intézményében, a temesvári, illetve a bukaresti műszaki egyetemeken, is jelentős erőfeszítéseket tettek az Edison implementálásának érdekében. A temesvári munkacsoportnak jelen tanulmány szerzője is tagja volt egy ideig. Az érdeklődés valószínűleg azzal magyarázható, hogy a függetlenedni akaró román számítógépipar már országszerte terítette a PDP-11 mintájára készült Coral mini-, illetve Intel 8080-alapú M18 mikrorendszereket. Ez egyértelműen hozta magával, szoftver téren is, a modernizálás követelményét, illetve a világ élvonalbeli kutatási eredményeinek az ismeretét és alkalmazását.

Bár az Edison jónéhány pozitív vonással rendelkezett úgy, mint egyszerűség, kis méret, adaptálhatóság, portabilitás, mégsem vált soha olyan népszerűvé, mint a létrejöttét meghatározóan befolyásoló Concurrent Pascal vagy Modula. Ennek okai, véleményem szerint, egyrészt a fennebb bemutatott hátrányainak, másrészt a párhuzamos programozás terén bekövetkezett gyors fejlődésnek és az újabb, vitathatatlanul magasabbrendűbb és komplexebb nyelvek megjelenésének tudható be.

Köszönetnyilvánítás

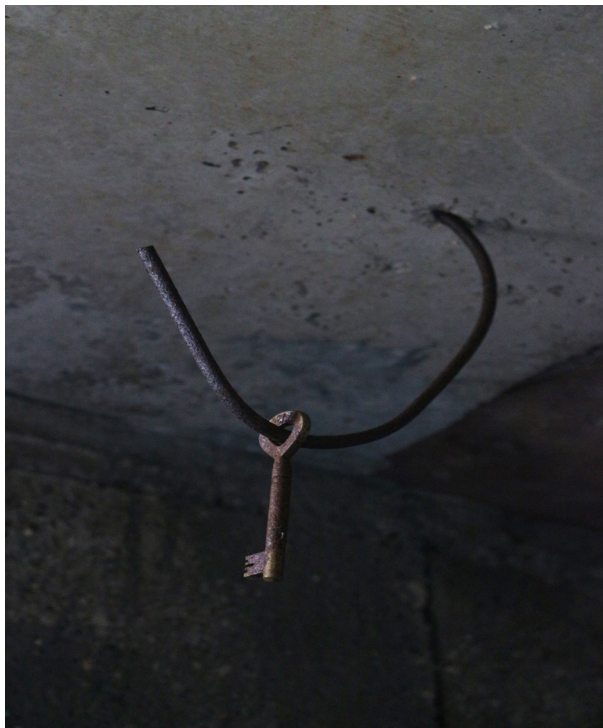
[31] P. Brinch Hansen (1983): „Using personal computers in operating system courses.” *ACM SIGOPS Operating Systems Review*, 17. (3.) Pp. 41–44.

[32] C. Dubnicki–J. Madey–W. Wygladala (1988): „Edison-N - an Edison implementation for a network of microcomputers-” *Software – Practice–Experience*, 18. (4.).

[33] G. Dotzel (1991): „The EDISON Multi-processor Language.” *The ModulaTor*, 1. (5).

[34] L. Böszörményi (1981): „Multi-task rendszerek fejlesztése magasszintű nyelven.” *Tanulmányok*, 128.

A közlemény megjelenését az EFOP-3.6.1-16-2016- 00003 „K+F+I folyamatok hosszú távú megerősítése a Dunaújvárosi Egyetemen” című projekt támogatta.



A Forth programozási nyelv alkalmazási lehetőségeiről a számítástechnika hatékony oktatásában

Absztrakt: A Forth az első integrált fejlesztési rendszer (Integrated Development Environment – IDE), mely 15 évvel előzte meg a Turbo Pascalt. Ma is fontos szerepet játszik a fejlesztésben, mivel igen jól használható a mikrorendszerek megvalósításában. A Forth fő előnyei között meg kell említenünk a számítógéptől való függetlenséget (portabilitás), a kis erőforrásigényt, a gyors működést és a bővíthetőséget. Ha mindehhez még hozzávesszük az interaktív kapcsolat biztosítására való alkalmasságát, könnyen megérthetjük, miért felül meg oly kiválóan a gyorsabb és hatékonyabb szoftverfejlesztés céljainak. Ugyanezen okokból kifolyólag, a Forth alkalmas lehet a programozás, illetve az informatika és számítástechnika oktatásában is. Tény, hogy minden integrált technológia, mint az Internet, adatbázisok, 3D stb. jelentős erőfeszítést követelnek meg mind a Forth implementáció fejlesztőitől, mind pedig, utólag, annak felhasználóitól. Kombinálni a Forth-ot meglévő Rapid Application Development (RAD) rendszerekkel lehetőséget nyújt rá, hogy a szükséges függvényeket és a Forth rendszer kiterjesztését minimális erőfeszítéssel illeszthessük. Jelen tanulmányunkban egy olyan új megoldást szándékozunk bemutatni, mely lehetőséget ad egy Forth-rendszer használatára az oktatásban, meglévő Pascal, illetve C++ fejlesztő rendszerrel kombinálva. Ennek során példákon keresztül mutatjuk be, hogy milyen egyszerű és kézenfekvő az általunk javasolt eljárás alkalmazása. A példák klasszikus oktatási témákra koncentrálnak, mint például a teknőc-grafika vagy a rekurzivitás. Jelen tanulmány előzményét két konferenciaszereplés ([1], [2]) képezi, amelyek után sajnos, egyéb elfoglaltságok miatt, a projekt háttérbe szorult. Ám, szándékunk szerint, a közeljövőben ezt sikerül tovább vinnünk.

Kulcsszavak: Forth; integrált fejlesztési rendszerek; programozás; számítástechnika-oktatás.

* Partiumi Keresztény
Egyetem, Nagyvárad
E-mail: mjbbartha@gmail.com

** Partiumi Keresztény
Egyetem, Nagyvárad
E-mail: zzakota@gmail.com

[1] M. Bartha–Z. Zakota (2015): „Using Forth Integrated with Pascal / C++ Development Systems in Education,” In: *Report of Meeting Researches in Didactics of Mathematics and Computer Sciences*, Novi Sad.

[2] M. Bartha–Z. Zakota (2015): „Using an Integrated Adaptive Forth System in Education,” In: *Proceedings of the 10th International Conference on Virtual Learning ICVL 2015*, Timișoara.

[1] M. Bartha–Z. Zakota (2015): „Using Forth Integrated with Pascal / C++ Development Systems in Education,” In: *Report of Meeting Researches in Didactics of Mathematics and Computer Sciences*, Novi Sad.

[2] M. Bartha–Z. Zakota (2015): „Using an Integrated Adaptive Forth System in Education,” In: *Proceedings of the 10th International Conference on Virtual Learning ICVL 2015*, Timișoara.

Abstract: Forth is the first Integrated Development Environment (IDE) to precede Turbo Pascal 15 years ago. Nevertheless, it still plays an important role in development today, as it can be used very well in the implementation of microsystems. The main advantages of Forth include computer independence (portability), low resource requirements, fast operation and scalability. Add to this the ability to provide an interactive connection, and we can easily understand why it meets the goals of faster and more efficient software development so well. For the same reasons, Forth may also be suitable for teaching programming as well as IT and computer science. In fact, all integrated technologies such as the Internet, databases, 3D, etc. they require a significant effort from both the developers of the Forth implementation and its users. Combining Forth with existing Rapid Application Development (RAD) systems allows you to fit the required functions and Forth system extensions with minimal effort. In the present study, we intend to present a new solution that allows the use of a Forth system in education, combined with an existing Pascal or C++ development system. In doing so, we show through examples how simple and obvious it is to use the procedure we propose. The examples focus on classic educational topics such as turtle graphics and recursion. The present study is preceded by two conference presentations([1], [2], after which, unfortunately, due to other occupations the project was pushed into the background. But we intend to take this further in the near future.

Keywords: Forth; Integrated Development Systems, Programming; ICT Education.

Az IKT által generált társadalmi változások

Az információs és kommunikációs technológiák (IKT) aktuális kérdései a társadalmi, politikai és gazdasági élet igen széles spektrumát érintik, ugyanakkor jelentős jogi és etikai vonzataik is vannak. Ilyen, például, az immár klasszikusnak számító, igen kiterjedt területet alkotó szellemi tulajdon védelme, szabadalmak, szerzői jogok, valamint, ehhez kapcsolódóan a tisztességes használat (Fair Use) kérdésköre. Ezzel rokon területet képez a kalózkodás és ahhoz kötődően a jogtalan P2P fájlmegosztás. Ehhez pedig az ingyenes szoftver, a nyílt forráskód, valamint a nyílt szabványok területe kötődik, ami viszont a trösztellenes szabályozást is befolyásolja. Már ebből is látszik, hogy mindez szervesen összefonódik és egy monstre kiterjedt elméleti és gyakorlati tudásteret alkot. Mindez jelentős befolyással van a közpolitikákra és a szabályozás gyakorlatára is, olyan területeken, mint pl.:

- az információhoz / tudáshoz való hozzáférés,
- a világháló semlegessége,
- nemzetközi egyezmények,
- frekvenciasávok allokációja.

Az IKT-k nem csak a közsférában generálnak átalakulásokat, de mély változásokat indukálnak a magánéletben is, az utóbbi évtizedekben főként a közösségi hálózatok által. Ugyanakkor nem hagyhatjuk figyelmen kívül azt sem, hogy az IKT által életre hívott problémák egy nem elhanyagolható része etikai jellegű, így például a kiberbűnözést, cenzúrát, biztonságot, vagy magánszférát érintő kérdések.

A folyamatok, természetesen, kétirányúak és a társadalmi környezet is változásokat indukál az IKT környezetben, a legváltozatosabb területeken, úgy mint:

- a világháló és alkalmazásai,
- a hálózati (főként TCP/IP-alapú) és mobil technológiák rohamos fejlődése és elterjedése,
- grafikus és multimédiás alkalmazások,
- beágyazott rendszerek,
- relacionális adatbázisok és adatbányászat,
- interoperabilitás,
- objektum-orientált programozás,
- szofisztikált alkalmazás-programozói felületek használata (API),
- ember-számítógép kölcsönhatás,
- kriptográfia és biztonság,
- folyamatosan megújuló és bővülő felhasználói területek.

A számítógépek világméretű elterjedésének köszönhetően az IKT befolyása drámaian megnövekedett és egy sor radikális változást gerjesztett a kultúrában. Ebből a számítástechnika akadémiai tantárgyként való teljeskörű elfogadása, majd folyamatos bővülése is következett. Általában pedig a „behálózottság” széleskörű elterjedése jött, úgy mint:

- e-üzlet
- e-kormányzás
- e-oktatás
- e-befogadás
- e-egészségügy
- e-egyre több minden...

[3] European Parliament, The, „European Parliament legislative resolution on the Commission communication on e-Europe – An Information Society For All: a Commission Initiative for the Special European Council of Lisbon, 23/24 March 2000 (COM(1999) 687 “ C5-0063/2000 “ 2000/2034(COS)),” 2000.

[4] Commission of The European Communities, „Communication from The Commission to The Council, The European Parliament, The European Economic and Social Committee and The Committee of The Regions - “i2010 – A European Information Society for growth and employment” {SEC(2005) 717},” Brussels, 2005.

[5] EP-C, „REGULATION (EU) 2021/694 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 29 April 2021 establishing the Digital Europe Programme and repealing Decision (EU) 2015/2240,” 29 04 2021. [Online]. Available: <https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32021R0694&from=EN>. [Hozzáférés dátuma: 15. 04. 2021]

IKT és EU politika

Mindez, természetesen, megjelent a köz- és különböző szakpolitikákban is. Már az eEurope 2000, az ún. Lisszaboni Stratégia is az információs társadalom megteremtését célozta meg az Európai Unión belül.

Alapvető céljai között olyanok szerepeltek, mint [3]:

- minden egyén és háztartás, valamint iskola, vállalat és intézmény bevezetése a digitális korbba, ill. csatlakoztatása a Világháléhoz;
- egy, a vállalkozási kultúrára alapozott, új gondolatok finanszírozására és kifejlesztésére kész, digitálisan alfabetizált Európa létrehozása;
- a teljes folyamat társadalmi befogadásának biztosítása, a fogyasztók bizalmának növelése és a társadalmi kohézió megerősítése.

A stratégia megvalósításának útját több dokumentum is fémjelzi. Ezek közül megemlíthjük itt az i2010 nevű dokumentumból az EB stratégiai keretprogramjának irányelveit [4]

- az Egységes Európai Információs Tér létrehozása:
 - hozzáférhető és biztonságos nagy sávszélességű kommunikáció,
 - gazdag és változatos tartalom és digitális szolgáltatások,
- innováció és beruházásnövelés az IKT-kutatásban:
 - világszínvonalú teljesítmények támogatása,
 - Európa közelítése a világszinten vezető pozíciót betöltő konkurenszekhez,
- a társadalmi, gazdasági és területi kohézió növelése:
 - egy befogadó európai információs társadalom létrehozása,
 - a növekedés és foglalkoztatottság támogatása a fenntartható fejlődéssel összhangban,
 - jobb közszolgáltatások és életminőség biztosítása,
- javaslatok kidolgozása és a keretprogram aktualizálása az elektronikus kommunikáció, az információs társadalom és a médiaszolgáltatások terén,
- közösségi pénzügyi eszközök stratégiai kutatásokba való befektetése,
- az IKT-innováció bátorítása.

Végül meg kell feltétlenül említeni az Európai Parlament és a Tanács 2021/694 rendeletét a Digitális Európa program (DIGITAL) létrehozásáról. [5]

A DIGITAL egy új uniós finanszírozási program, amelynek középpontjában a digitális technológia eljuttatása áll a vállalkozások, az állampolgárok és a közigazgatás számára. A Digitális Európa program stratégiai támogatást nyújt az olyan aktuális kihívások megválaszolásához, mint a környezeti problémák kezelése, a járványok elleni védekezés vagy a külső rendszerektől és megoldásoktól való függőség.

Ami az IKT szűkebb területét illeti, öt kulcsfontosságú irányban támogat projekteket:

- szuperszámítógépek,
- mesterséges intelligencia,
- kiberbiztonság,
- fejlett digitális készségek elsajátítása,
- digitális technológiák széles körű használatának biztosítása a gazdaságban és a társadalomban, ideértve a digitális technológiát is.

A tervezett összköltségvetés 7,5 milliárd euró (folyó árakon), amelynek célja a gazdasági fellendülés felgyorsítása és az európai társadalom és gazdaság digitális átalakulásának alakítása, mindenki számára, de különösen a kis- és középvállalkozások számára biztosítva előnyöket.

IKT és oktatás kölcsönhatása

Az IKT olyan változásokat generált az oktatás, képzés és nevelés világában, mind az elmélet, mind pedig a gyakorlat síkján, melyekre néhány évtizeddel ezelőtt senki sem gondolt volna. Ez a megállapítás nem csak a szakirányú képzésre vonatkozik, hanem általában az egész szakmára. A legszembetűnőbbek között olyanokat említhetünk meg, mint:

- a számítástechnika expanziójához vezető műszaki változások közvetlenül hatnak az oktatási/képzési/nevelési kultúrára;
- a számítógép-hálózatok könnyebben megvalósíthatóvá és rugalmasabbá teszik a távoktatást, jelentős növekedést generálva az üzletágban;
- az élethosszig tartó tanulás megjelenése hozzájárult a távolsági oktatási formák használatának elterjedéséhez;
- a számítógép-hálózatoknak köszönhetően sokkal gyorsabbá vált a tantervi erőforrások széleskörű cseréje;
- a technológia kihatással van a pedagógiára a bemutató szoftverek, egyéni munkaállomások stb. megjelenése által;
- a számítástechnikai tantervek valós időben követhetik a technológiai változásokat.

Ebben az esetben valós kölcsönhatásról, vagyis két irányban zajló folyamatok egyidejű meglétéről beszélünk. Egyrészt az IKT hat az oktatásra, olyan eszközökkel, mint pl.:

- az oktatási intézményrendszer átalakulása, új intézményi formák megjelenése;
- új oktatási/képzési formák megjelenése és elterjedése (Life Long Learning, távoktatás, online képzés stb.);
- új oktatási/képzési technikák kialakítása;
- új típusú (multimediális, interaktív stb.) tananyagok kidolgozása;
- az oktatási/képzési infrastruktúra átalakulása (IKT);

másrészt az oktatás terén végbement változások is kihatnak az IKT-szektor alakulására, olyan csatornákon, mint pl.:

- a szakemberek képzése;
- az általános digitális alfabetizációs szint emelkedése;
- visszahatás a kutatási, fejlesztési és innovációs tevékenységekre.

A kérdéskör komplexitása, és ebből fakadóan: vizsgálatának nehézsége, nemcsak az oda-vissza kapcsolatok számosságából fakad, hanem a két terület – IKT és oktatás – más szakterületekkel való szoros összefonódásából is. Az összetettséget növelő tényezők között meg kell említenünk olyanokat, mint:

- a tárgyak kiterjedtsége:
 - matematika, logika, természettudományok;
 - műszaki tudományok, technológiák;
 - társadalomtudományok;
 - filozófia, etika;
- inter- és multidiszciplinaritás;
- a tárgyak gyorsütemű változása/fejlődése;
- globalizáció.

Az IKT szerepének ugrásszerű növekedése az oktatásban

A koronavírus-járvány kitörése óta eltelt időszak jól illusztrálja, hogy milyen nagy jelentőséggel bír az IKT az oktatásban és mennyire felértékelődhet ez a szerep egy krízishelyzetben. Az előtt, a világ országaiban az oktatás nagyrészt a hagyományos módon folyt, ám a járvány indukálta változások hatására, a társadalom egyik legnagyobb tehetetlenségű alrendszere, a stabilitás és az értékek hosszú távú megőrzésének egyik legfőbb eszköze, elképzelhetetlen gyorsasággal alkalmazkodott a megváltozott körülményekhez. Iskolák és egyetemek, kihasználva az IKT-ban rejlő lehetőségeket, gyakorlatilag egyik napról a másikra tértek át az online formák használatára. Ez egyrészt – minimális átállási költséggel – lehetővé tette a képzés folytatását, másrészt viszont egy sor új problémát generált és új megvilágításba helyezett számos régít. Számos

lappangó hiba került felszínre, rávilágítva a hagyományos oktatási és képzési formák anakronizmusaira, belső ellentmondásaira. Láthatóvá váltak az intézményi infrastruktúra- és működési keret hiányosságai, rendellenességei, a szabályozás elavultsága nyilvánvalóvá vált. Az események az oktatás egészét és annak csatolt területeit is érintik, úgy mint [6]:

- a gazdaságossági szempontok prevalenciája az átalakulásban;
- az infrastruktúra, mint teher az oktatási intézmények számára;
- a technológiához való hozzáférés nehézségei;
- egy új digitális szakadék megjelenésének a lehetősége;
- a mobil technológia rohamos térnyerése;
- platformok és virtuális terek megjelenése és alkalmazása;
- a tanárok és oktatók digitális kompetenciájának kérdése;
- a diákok és hallgatók differenciálásának nehézségei;
- a tanulási eredmények mérésének és értékelésének nehézségei;
- a bizalom és ellenőrzés kérdése a tanár–diák–szülő–menedzsment viszonyrendszerben;
- a szülők tanulásban betöltött szerepének újraértékelése.

A koronavírus-krízis csak kiváltója volt azoknak a változásoknak, amelyek jó ideje váratnak magukra. Ezeknek a változásoknak középpontjában az online oktatás áll, mint egy eddig messze nem eléggé kiaknázott, felbecsülhetetlen lehetőség. Jövőjét az határozza meg, miként kísérletezünk vele, a kutatások által kapott eredményeket hogyan teszteljük, rögzítjük és értelmezzük. Ennek a kísérletnek a kimenete nemcsak az oktatás egészére lesz kihatással, de a politikai diskurzust és a gazdaságpolitikát is gazdagítja majd. Ennek az átalakulásnak egyik legfontosabb eleme a folyamatban résztvevő aktorok – diákok, tanárok, az oktatási folyamatok szervezői és vezetői és nem utolsósorban, a szülők – digitális műveltségének a folyamatos növelése és kiterjesztése. És itt nem csak az IKT eszközök felhasználói szintű használatáról van szó, hanem egy mentalitásváltásról, egy új szemléletmód kialakításáról is.

Digitális Alapműveltség

Az IKT és oktatás kölcsönhatásának vizsgálatakor roppant fontos annak eldöntése, hogyan határozzuk meg a digitális alapműveltség (Digital Literacy) fogalmát. Nevezik még digitális alfabetizációnak meg digitális írástudásnak is és ezek a nevek, egymás mellé téve ki is fejezik grosso modo a fogalom lényegét. Ám ne tévesszük szem elől, hogy megjelenése óta

[6] Zakota Zoltán (2020): „Ok-tatás járvány idején – az alkalmazkodás nehézségei,” Civil Szemle. *Oktatás, digitalizáció, civil társadalom külön-szám*, Pp. 59–71.

[7] ALA, „Digital Literacy,” American Library Association, 2021. [Online]. Available: <https://literacy.ala.org/digital-literacy/>. [Hozzáférés dátuma: 14. 07. 2021.]

a fogalom folyamatosan változik, lépést tartva az IKT-eszközök fejlődésével. Ezért szükség van egy kellően általános, átfogó eszközfüggetlen meghatározásra. Az Amerikai Könyvtárak Szövetsége (American Library Association – ALA) által adott definíció meg is felel elvárásainknak. E szerint, a digitális írástudás „az információs és kommunikációs technológiák felhasználásának képessége az információk megtalálásához, értékeléséhez, létrehozásához és kommunikációjához, beleértve mind a kognitív, mind a technikai képességeket.” [7]

A terület szerteágazó volta miatt, a legfontosabb kérdés, amit fel kell tennünk magunknak, hogy mit is oktassunk, mikor a digitális alpműveltség eléréséről beszélünk. Meglátásunk szerint, alapvetően a következő kérdéseket kell megválaszolnunk:

- mi egy számítógép és mire való?
- mi egy számítógép felépítése és mik működtetésének alapelvei?
- melyek a biztonságos használat feltételei?
- milyen társadalmi implikációi vannak az IKT-nak?
- hogyan állítsunk össze dokumentumokat (szerkesztés, szkennelés, nyomtatás)?
- hogyan használjunk alapszinten táblázatokat és adatbázisokat?
- melyek a számítógépes kommunikáció legelterjedtebb eszközei?
- hogyan kezeljük / használjunk különböző típusú állományokat?
- mik a multimediális alpműveltség nélkülözhetetlen elemei?
- hogyan alkalmazzuk a mobiltechnológiát?
- hogyan kezeljük / használjuk az internetes forrásokat?
- hogyan használjuk az IKT-t etikus módon?

Az informatikai és számítástechnikai oktatás jelentősége

Manapság, gyakorlatilag nincs olyan szakirány a közép, illetve felsőoktatásban, amelynek tanrendjében ne szereplnének, valamilyen formában, az IKT: számítástechnika, számítógéphasználat, informatikai alapfogalmak stb. Sajnos, számos esetben ez nem jelent egyebet, mint a Microsoft Office leggyakrabban használt programjainak – a Word szövegszerkesztőnek, illetve az Excel táblázatkezelőnek és ha még belefér a rendelkezésre álló időbe, az Access adatbáziskezelőnek – az alapfokú ismerete. Persze, irodai programcsomagot tanítani egy roppant kényelmes megoldás: tanárnak, hallgatónak alig kell készülnie és a sikeres vizsga garantálva, minimális erőfeszítéssel. Ráadásul a hallgatók nagyobb hányada már eleve úgy került be az egyetemre, hogy alapszinten tudja kezelni a szövegszerkesztőt és táblázatkezelőt. Felmerül tehát a kérdés: miért is tanítsunk IKT-t az egyetemen? Meglá-

tásunk szerint, igenis van létjogosultsága az IKT oktatásának egyetemi szinten is, amennyiben azt egy tágabb keretbe helyezzük, nem korlátozva a tárgyat néhány irodai szoftver felhasználói szintű bemutatására. Indokaink a következők:

- egy, a mindennapok igényeit kielégítő, digitális alapműveltségi szint biztosítása minden hallgató számára;
- az önálló gondolkodás fejlesztése: a technológiai tekintély elvének lerombolásával arra készíteni a hallgatókat, hogy ne fogadjanak el egy eredményt csak azért, mert azt valamilyen gépi eszköz segítségével számították ki, hanem mindig vizsgálják meg, hogy az adott kontextusban helytálló lehet-e;
- a technika varázstanítása: „a kellően fejlett technika megkülönböztethetetlen a mágia-tól” állítja Clarke Harmadik Törvénye [8] és célunk elérni, hogy hallgatóink felesleges csodálattól mentesen, racionálisan viszonyuljanak a technikához;
- ugyanakkor, szükségesnek látjuk növelni a technológiába vetett bizalmat is: a racionalitás terének kibővítésével szeretnénk elérni, hogy hallgatóink nagyobb bizalommal használják a digitális eszközöket és értsék, mikor és miért fogadhatják el az általuk produkált eredményeket;
- áttérni a kvantitatív gondolkodásmódról a kvalitatívra: nem csak számolásra és/vagy számítások elvégzésére használni a számítástechnikát, hanem értelmezni a számítások eredményeit, sőt az értelmezésben is használni azt;
- elsajátítani az algoritmikus gondolkodásmódot, vagyis egy racionális, a feladatok elemzésén és a megoldás lépésenkénti megtervezésén alapuló problémamegoldó technika elsajátítását;
- végül a legkézenfekvőbb, triviális válasz: azért, mert benne van a tantervben és előbb-utóbb sikeres vizsgát kell belőle letenni.

A Forth programozási nyelv

A Forth programozási nyelvet Charles Henry Moore, az egyesült államokbeli Kitt Peak obszervatórium csillagásza alakította ki az 1960-as évek elején. Célja egy olyan könnyed, rugalmas és gyors nyelv megalkotása volt, amely ugyanakkor egy korlátlanul bővíthető utasításkészlettel is rendelkezik [9].

Gyakorlott programozóként, Moore tisztában volt kora számítástechnikájának egyik alapvető problémájával, a hardver és a szoftver fejlődése közötti diszkrepanciával. Ahogy ő maga fogalmazza azt meg: „[a] minikomputerek közelmúltbeli fejlesztése javította képességeiket

[8] A. C. Clarke (1977): Profiles of the Future – An Inquiry into the Limits of the Possible Revised edition, 2nd. Popular Library, CBS Publications.

[9] A. Seres–L. Fenyő–K. Gyalogh (1986): A FORTH programozási nyelv. Budapest: Műszaki.

[10] C. H. Moore (1974): „FORTH: A New Way to Program a Mini-Computer,” *Astronomy and Astrophysics Supplement Series*, 1. (15.) Pp. 497–511.

[11] C. H. Moore–G. C. Leach (1970): „FORTH - A Language for Interactive Computing,” Amsterdam–NY.

[12] „The Evolution of Forth,” FORTH, Inc., [Online]. Available: <https://www.forth.com/resources/forth-programming-language/>. [Hozzáférés dátuma: 15. 07. 2015.]

[13] L. Brodie (1981): *Starting FORTH*. NJ: Prentice-Hall, Inc.

és csökkentette költségeiket addig a pontig, ahol nagyobb számítógépeket és egyedi hardvereket is kiszorítanak. Egy kicsi számítógép azonban nem tud hatékonyan használni egy hagyományos magas szintű nyelvet, és általában assembler nyelven van programozva. Ez az alacsony szintű megközelítés hozzáértő programozókat és jelentős erőfeszítéseket igényel. Szinte egyetemes boldogtalanság tapasztalható a programozás költségeit, késedelmét és minőségét illetően. Ez a helyzet a standard technikák hatékony megvalósításának határozott próbálkozásai ellenére is fennáll.” [10]

A nagyméretű számítógépekre gyártott szoftverek nyelvek hierarchiájára épülnek: az assembler meghatározza a fordító és az operációs rendszer nyelvét, emez pedig a munkakörnyezet nyelvét; a fordító az alkalmazói programok nyelvét, az utóbbi pedig meghatározza a bemenet nyelvét. A felhasználó valószínűleg nem is ismeri ezeket a nyelveket, sőt szinte biztos, de ettől függetlenül azok még léteznek, közte és a számítógépe között állva és korlátokat állítva elé. A nyelvek e hatalmas hierarchiája hatalmas beruházásokat igényel emberi és gépi munkában, ami persze jelentős költséggel is jár, mind a termelésben, mind a fenntartásban. És a sok erőfeszítés és költség után még mindig semmi garancia sincs rá, hogy a programok hibamentesek, felhasználóbarátak és képesek kielégíteni minden velük szemben megfogalmazott elvárást. A Forth célja, megalkotójának szavaival élve, véget vetni ennek a „többszintes rémálomnak”. [11]

A kora 1970-es években következett a nyelv bemutatása, illetve Moore találkozása Elizabeth Ratherrel a US National Radio Astronomy Observatory-tól, akivel közösen megalakították a FORTH Inc. vállalatot [12]. Ezután számos fejlesztés, verzió, illetve szabvány következett:

- 1976-tól: microFORTH fejlesztések Intel 8080, Motorola 6800 és Zilog Z80 mikroprocesszorokra;
- 1978: Forth az első rezidens szoftver az Intel 8086 chipre;
- 1979: FORTH-79 szabvány;
- 1983: FORTH-83 szabvány;
- 1984: MacFORTH az első rezidens fejlesztés Apple Macintoshra;
- 1994: ANS Forth (ANSI) szabvány.

Azt, hogy mi a Forth és hogy mire való talán Leo Brodie, a nyelv egyik legjobb ismerője és alkalmazója, valamint több nagysikerű Forth-tankönyv írója fogalmazta meg a legtömörebben [13]: „A Forth sok minden:

- magas szintű nyelv.
- assembler.

- operációs rendszer.
- fejlesztési eszköztár.
- egy szoftvertervezési filozófia.”

A Forthnak sok olyan jellemzője van, amely, ha nem is emeli a legnépszerűbb programozási nyelvek közé, de felhasználói táborán belül igen magas megelégedettségnek forrása. Ezek a jellemzők, röviden felsorolva, a következők:

- láncolt kódú interpreterre alapozó interaktivitás;
- kompilálható;
- imperatív és procedurális, szótárszerűen kiterjeszthető;
- veremre alapozott, fordított lengyel jelölést használó;
- strukturált;
- reflexív;
- konkatenatív;
- multitaskingot lehetővé tevő;
- explicit nyelvtan és típusellenőrzés nélküli.

A Forthnak számos előnye van, amelyek miatt a gyakorlott programozók körében megérdemelten népszerűségnek örvend [14]:

- teljes programozási környezet biztosítása;
- jól használható a mikrorendszerek megvalósításában,
- számítógéptől való függetlenség (portabilitás),
- kis erőforrásigény,
- gyors működés,
- gyors szoftverfejlesztés,
- interaktív kapcsolat biztosítása fejlesztővel és felhasználóval,
- bővíthetőség.

Forth-alkalmazások a nagyvilágban

A nyelv alkalmazási területei igen változatosak és kiterjedtek. Ezek között megemlíthetjük, természetesen a teljesség igénye nélkül, a következőket [14]:

- szövegfeldolgozás;
- adatbáziskezelés;
- robotvezérlés;
- beágyazott rendszerek programozása;

[14] G. Lipovszí-L. Subai-T. Beszedá (1985): *FORTH programozási rendszer és nyelv*. Budapest: LSI Alkalmazástechnikai Tanácsadó Szolgálat.

[15] „*Introductions*,”
FORTH, Inc., 2021.
[Online]. Available:
<https://www.forth.com/starting-forth/0-starting-forth/>.
[Hozzáférés dátuma:
15. 07. 2021.]

- folyamatirányítás;
- számítógépes grafika;
- oktatás.

A továbbiakban, illusztrálandó a Forth-alkalmazások diverzitását és dimenzióit, megemlítünk néhány rendszert, amelyeket a világ különböző részein implementált sikeresen a Forth Inc. [15]:

- AVCO / Textron Systems: épületautomatizálás és kiegészítő szolgáltatások a King Khaled Nemzetközi Repülőtéren, Szaúd-Arábiában; a rendszer kilenc PDP 11/44-es, 378 8086 alapú számítógépet és 320 8085-alapú biztonsági processzort tartalmaz, amelyek együttesen több mint 36000 pontot figyelnek és irányítanak.
- Eastman Kodak Company: filmsűrűség minőségellenőrzési rendszer, amely tartalmazza a film mozgásának vezérlését, a filmsűrűség-lépések automatikus felismerését és az egyedi IEEE-488 busz interfészt.
- Federal Express: kézi SuperTracker, amelyet minden FedEx szállító ügynök használ, vonalkódolvasót, billentyűzetet és 2×20 soros kijelzőt tartalmaz. Kiterjedt csomagbevezetési és nyomkövetési funkciókat végez, beleértve az amerikai repülőtéri kódok és több, mint 10000 irányítószám keresztingexelését. Az akkumulátor élettartamának meghosszabbítása érdekében intelligens kikapcsolási szekvenciával rendelkezik.
- NASA Goddard Űrrepülési Központ:
 - 50 láb hosszú, hatsuklós kar irányítása az űrsikló szimulátorhoz. A kiterjedt matematikai rutinok két háromtengelyes joystick-parancsot alakítanak át a szükséges csuklósebességekre hat különböző koordináta-rendszerben.
 - Multitasking operációs rendszer, Forth fordító és könyvtárak az UT69R000 sugárrzással edzett mikroprocesszorhoz, amelyet az Űrsikló műszereiben használnak.
 - Forth-alapú kis teherbírású szállási interfész modul (SPAIM) kifejlesztése, amely összekapcsolja a Shuttle Solar Backscatter Ultraviolet (SSBUV) műszert az Űrsikló repülési rendszereivel. Az SSBUV-t az ózonmérő műszerek kalibrálására használják a NOAA műholdak fedélzetén.
- Owens-Corning Fiberglas: évek óta a Forth-ot használja alapfirmware-ként az elosztott ipari vezérlőkben, amelyek számos különféle funkciót látnak el az üvegszál-termekek gyártásában.
- Saturn Corp: elosztott HVAC-rendszer a teljes autószerelő üzem számára, amely

több mint kétszáz 40 lóerős fűtési-hűtési-párásító egységet vezérel, Z-80-asokkal, egy kétszintű hálózaton keresztül, személyi számítógépeket használva szöveges és grafikus rendszerfigyelőként.

- Minnesotai Egyetem: PC-alapú rendszer távcsővezérléshez és adatgyűjtéshez (IEEE-488 buszon keresztül), adatelemzéshez és grafikus megjelenítéshez.
- VertexRSI (Tripoint Global): szoftver egyedi műholdas nyomkövetőkhöz. Frekvenciaszintetizátor vezérlést, távoli RS-232 parancsportot, vákuum fluoreszcens grafikus kijelzőt tartalmaz.
- Construction Computer Software: MARS és a CANDY alkalmazások, amelyek világszabványok. A CCS szoftver példa nagyméretű ProForth for Windows-ban írt alkalmazásra és a VFX Forth verzió már több mint 850000 kódsorból áll. CCS szoftvert használtak az új Hongkongi Chai Tak repülőtér tervezésénál is.
- Microsoft: Barefoot Auditor, melynek segítségével információkat gyűjt saját PC-iről és amelyet az MPE Forth rendszereinek egyikével írtak.
- Micross Automation Systems: az MPE ProForth for Windows szoftvert használja a kereskedelmi mosodai vezérlő rendszereiben, az MPE Forth 6 fordítót pedig a valós idejű vezérlést végző PLC-i számára.
- Europay International Open Terminal Architecture (OTA): az OTA egy Forth virtuális gép (VM) architektúrával szállítja az alkalmazásokat közvetlenül a fizetési terminálokhoz, függetlenül azok hardverétől vagy processzorától. Az OTA virtuális gépet számos CPU-ra telepítették. Az OTA projektben kevesebb, mint 30 programozó vett részt, két kontinensen több helyén.

[16] D. Rushkoff (2010): *Program or be Programmed*. New York: OR Books.

A Forth alkalmazása az oktatásban

Douglas Rushkoff, amerikai médiatudós és író, szerint „megtanítjuk rá a gyerekeket hogyan használjanak szoftvert az íráshoz, de arra nem hogyan írjanak szoftvert. Ez azt jelenti, hogy hozzáférésük lesz a mások által számukra biztosított lehetőségekhez, de nem tudják majd felmérni eme technológiák értékteremtő lehetőségeit saját maguk számára” [16].

[17] „4E4th – Forth For Education,” [Online]. Available: <http://www.4e4th-ide.org/>. [Hozzáférés dátuma: 15. 07. 2021.]

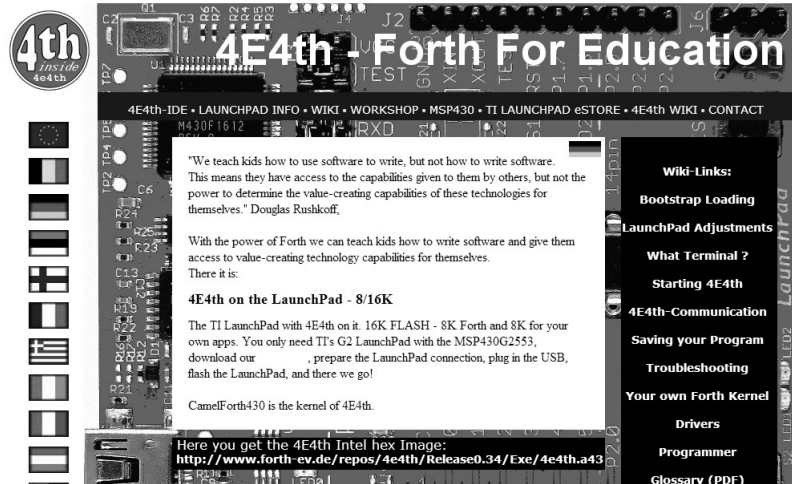
[18] D. Bruehl: „Forth for Education – 4E4th and 4E4th IDE,” [Online]. Available: <http://www.complang.tuwien.ac.at/anton/euroforth/ef12/papers/bruehl.pdf>. [Hozzáférés dátuma: 15. 07. 2021.]

[19] Texas Instruments, „MSP-EXP430FR5739 Experimenter Board,” [Online]. Available: <https://www.ti.com/tool/MSP-EXP430FR5739?DCMP=ep-mcu-msp-dsbg&en&HQS=ep-mcu-msp-dsbg-pr-evm2-en>. [Hozzáférés dátuma: 15. 07. 2021.]

[20] „4E4th - the New Version 2018,” Forth-Gesellschaft e.V., [Online]. Available: <https://wiki.forth-ev.de/doku.php/en:projects:4e4th:start>. [Hozzáférés dátuma: 15. 07. 2021.]

[21] „CamelForth,” [Online]. Available: <http://www.camelforth.com/news.php>. [Hozzáférés dátuma: 15. 07. 2021.]

1. ábra. A 4E4th oktatási platform nyitólapja.



Forrás: [17]

Ez az idézet olvasható a 4E4th – Forth for Education nyitólapján is (1. ábra), mely a Forth programozási nyelv eddigi legnépszerűbb oktatási alkalmazásának ad otthont [17]. A Forth for Education – 4E4th and 4E4th IDE projektet Dirk Bruehl kezdeményezte 2012-ben, mikor Forth felhasználóként, találkozott az igényeit kielégítő megfelelő hardverrel [18]. Ez a Texas Instruments MSP-EXP430FR5739 Experimenter Board volt, egy az MSP430FR57xx eszközök számára készült fejlesztési platform [19]. Ez kiválóan bevált, mint az új generációs MSP430-as, FRAM (Ferroelectric Random Access Memory) memóriát integráló, mikrokontrolleres eszközök alapja. Ugyanakkor kompatibilis számos kis fogyasztású TI wireless modullal is.

2018-ban, a 4E4th-nak megjelent egy új verziója [20], a B. J. Rodriguez által készített MSP430 CamelForth-ra alapozva. A CamelForth beágyazott mikroprocesszorok (8051, 8086, Z80 és 6809) számára készült, ANS Forth-tal kompatibilis megvalósítás. Eredetileg a The Computer Journal oktatási projektjének indult, de azóta népszerűvé vált a beágyazott rendszerek programozása terén. [21]

Van a Forth-nak egy fontos vonása, ami kihasználható az oktatásban is. A Forth a programozót teszi felelőssé a számítógép hatékonyságáért. Ugyan nehezebben elsajátítható, mint a hagyományos magas szintű nyelvek, amelyek lényegében hasonlítanak egymásra (vagyis egyiket megtanulva nem nehéz megtanulni másikat), de miután a programozó elsajátította, a Forth lehetőséget nyújt a processzoridő és a memóriaterület minimalizálására, valamint egy olyan szervezési filozófia birtokába kerül, amellyel drámai módon csökkentheti a projektfejlesztési időt.

A kérdésre, hogy miért pont a Forth nyelvet választaná valaki oktatási eszközül, a valószínűleg legfrappánsabb választ maga a nyelv megalkotója, Charles H. Moore adta: „öröm pár egyszerű szót leírni és megoldani egy problémát. Mint agytorna, messze jobb, mint a kártya, keresztrejtvény vagy sudoku és hasznos.” [22]

Egy újabb Forth-rendszer

Természetesen, felvetődik a kérdés, teljesen jogosan, hogy mi szükség lenne még egy Forth-rendszerre? A választ a szoftver és hardver eszközök gyors ütemű fejlődése rejti magában. Amíg a számítógépek egy alfanumerikus kijelző és billentyűzet segítségével kommunikáltak a felhasználóval, szinte minden gépre/processzorra megjelent a Forth és a megírt programok szinte módosítás nélkül használhatóak voltak bármely konfiguráción. Ám a grafikus operációs rendszerek és az új technológiák (USB, internet...) megjelenése szükségessé tette ezek használatának beépítését a programfejlesztő rendszerekbe. A Forth fejlesztésére több próbálkozás is történt, de mindegyik „intenzív” elven, azaz a szókészlet bővítésén alapult. Például a win32forth 2014 decemberi release több, mint 5000 Forth szót tartalmaz (2. ábra).

[22] N. Hamilton: „*The A-Z of Programming Languages: Forth*,” 08. 07. 2008. [Online]. Available: <http://www.computerworld.com/article/2534609/app-development/the-a-z-of-programming-languages--forth.html>. [Hozzáférés dátuma: 21. 08. 2015.]

2. ábra. A Forth-79 standard szókészletének egy részlete

ok			
words			
----- FORTH -----			
EMPTY	FORTHBOOT	LOADCONFIGFILE	USERCONFIGS
CONFIG\$	SET-SHELL	ABS\$	"OF-INCLUDE
("OF-INCLUDE)	"OF-COMPARE	("OF-COMPARE)	BACKGROUND
FOREGROUND	SIZE-POINTER	UPARROW-POINTER	WAIT-POINTER
NOWAY-POINTER	ICON-POINTER	IBEAM-POINTER	CROSS-POINTER
ARROW-POINTER	NEW-POINTER	SP>COL	SET-POINTER
NOTE	F^X	FPOP	FPUSH
(F\$)	FR\$	FDUP>R	F>R
R>F	FSP	FSTACK	?UPPERCASE
ASCIIZ->ASC-LEN	OF-RANGE	_OF-RANGE	SYS
^	ZEXEC	.BROWSE	.DOS
.EDITOR	WIN32S?	WIN98?	WIN95?
NT?	WITH-IMG	<CLASS	<OBJECT
HEAP>	<A	--	DOC
(*)	/*	CONSOLEWINDOW	CMD
MULTISTATUSBAR	STATUSBAR	-STATUSBAR.F	COMMANDWINDOW
LOGKEYSTROKES	HANDLEKEYDOWN	HANDLEKEYS	SET-CONSOLE-POPUP
SET-CONSOLE-MENU	MENU-OFF	CONSOLE-POPUP	WIN32FORTH-POPUP-BAR
CONSOLE-MENU	WIN32FORTH-MENU-BAR		MARK-ALL
CUT-CONSOLE	COPY-CONSOLE	LOADPROJECT	COUNTMAPPEDFILE
PROJECTHNDL	LATESTPROJECTFILES	"SETTOFOREGROUND	WINDOWTOFOREGROUND
SEARCH-WINDOW	*SEARCH	*BUFFER	#NUMBER-LINE>
+TRAILING	W-SEARCH	WILDCARD	NEXT-SEARCH
TARGET-SEARCH	SCAN-FOR-WILDCARD	CONTINUE-W-SEARCH?	WILDCARD-CHAR
CONTAINING?	STARTING-WITH?	CASESENSITIVE?	LAST-CNT
SEARCHIA	COMPAREIA	S-EXCHANGE	-DUP
NEVER	W_SEARCH.F	-LOADPROJECT.F	-LDE
.WINDOWS	.DFC	DEXH	XREF
HELP-SYSTEM	CLASS-BROWSER	.FORTHFORM	.IDE
-TOOLS.F	NEWEDITDIALOG	PAGE-UP-SETUP	PAGE-UP-DIALOG
ABOUT-WIN32FORTH	IDOK3	IDD_PROMPT2_TEXT	IDOK2
IDD_EDIT2_TEXT	IDD_PROMPT2_TEXT	IDD_EDIT2_DIALOG	IDD_EDIT_TEXT2
IDD_PROMPT_TEXT2	IDD_REPLACE_DIALOG	IDB_OPTION	IDD_EDIT_DIALOG
IDD_PROMPT_TEXT	IDD_EDIT_TEXT	IDD_TITLE_TEXT	IDD_4UP
IDD_2UP	IDD_PAGEUP	IDD_AVAIL_MEMORY	IDD_AVAIL
IDD_SAVE_MEMORY	IDD_ABOUT_TEXT4	IDD_ABOUT_TEXT3	IDD_ABOUT_TEXT2
IDD_ABOUT_TEXT	IDD_ABOUT_HEAD	IDD_ABOUT_FORTH	IDD_TEXT
IDD_CHECK	IDD_FONT	IDD_FONTDLG	FORTHDLG
BROWSEFORFOLDER	-BROWSEFLD.F	MIN-MENU-BAR	POPUPBAR
MENUBAR	NEXTID	IDCOUNTER	IDSTART
MODELESSDIALOG	DIALOG	LOAD-DIALOG	READ-DIALOG
FIND-DIALOG-ID	?DLGERR	DIALOGID?	COMBOFIELD
EDITFIELD	PICTUREBUTTON	TOOLBAR	DEFTOOLSPACING
DEFTOOLWIDTH	DEFTOOLHEIGHT	LBUTTONBAR	HBUTTONBAR
BHUTTONBAR	VBUTTONBAR	GENERIC-BUTTONBAR	BUILDBAR
DEFBHEIGHT	DEFBWIDTH	DIALOGWINDOW	BUTTONCONTROL
RADIOCONTROL	CHECKCONTROL	STATICCONTROL	GROUPOFCONTROL
LISTCONTROL	COMBOLISTCONTROL	COMBOCONTROL	EDITCONTROL
CONTROL	IN-BUTTON?	MOUSE-IS-DOWN?	INFO-FLAG
CALLWINDOWPROC	SUBCLASS-ENTRY	SUBCLASS-RETURN	SUBCLASS-WNDPROC
ZMESSAGE	MESSAGE-ORIGIN	MESSAGE-ON	MSG-WINDOW
MSGWINDOW	CHILD-WINDOW	LOADICONFILE	SEND-WINDOW
FIND-WINDOW	WINDOW	WNDPROCERROR	ERRSTRING
ERRCFA	CLASSNAMEID	DEFAULTMENUBAR	DIALOG&CONTROL
ZERO-WINDOWS	GENERIC-WINDOW	GET-MOUSE-XY	WRECT
DIALOG-LINK	WINDOWS-LINK	PRINT-CONSOLE	PRINT-SCREEN
#PRINT-SCREEN	ROWSTRING	4PRINT	2PRINT
FPRINT	\$FPRINT	PRINT	PRINTER
FOUR-PAGE	TWO-PAGE	SINGLE-PAGE	PRINT-SCALED
PAGE-SCALED	PRINT-MULTI-PAGE	START-RAW2	START-RAW
START-SCALED2	PRINT-SCALED	PAGE-SETUP	START-PRINTER2
THEPRINTER	WINPRINTER	SPLITLINE	-BLSCAN1

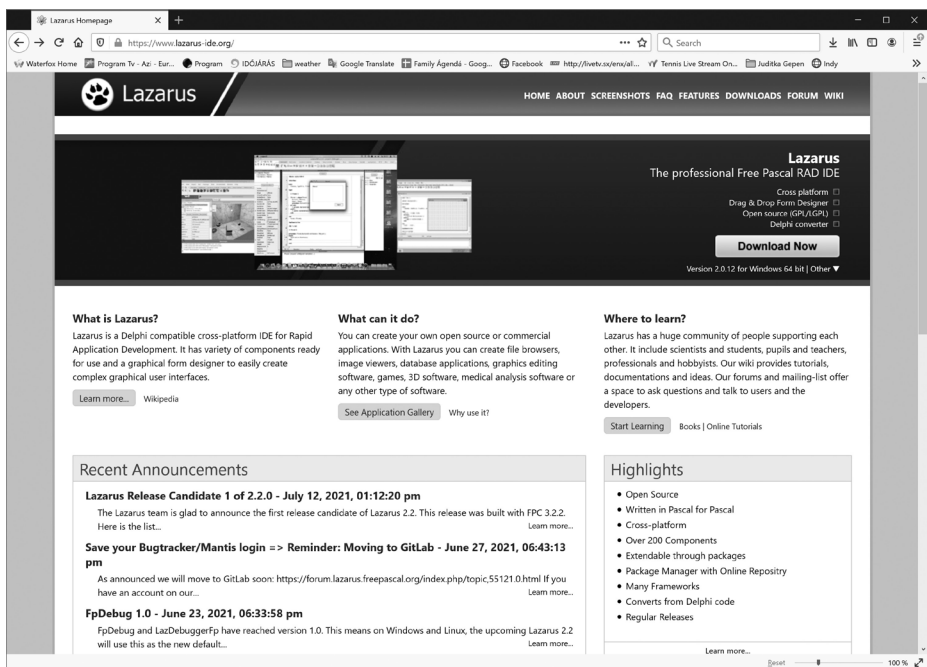
Date: Thursday, August 20 2015 10:49AM

Page: 1 of 10

Jelen projekt lényege az új lehetőségek kihasználása Forth-ból, gyakorlatilag az alapszókészlet bővítése nélkül. Egy speciális szövegmező használatával, amely fogadja a parancsokat és kijelzőként szerepel, a Forth-rendszer beépíthető bármely Delphi/Lazarus programba, mivel a kód és adatszegmens Forth-kóddal vezérelhető bármely Lazarus/Delphi-komponens. A szövegben szereplő ábrákon néhány, a rendszerünkkel készült programról mutatunk be képeket. Talán a mellékelt minimális teknőc-grafika illusztrálja a legjobban ennek a kombinációnak a lehetőségeit, ez ugyanis 5–6 sornyi programmal lett felépítve.

Mivel az általunk használt Lazarus Objective Pascal IDE [23] (3. ábra), illetve NASM Netwide Assembler [24] (4. ábra) open source és cross platform eszközök, az így épített Forth természetesen futtatható Windows, MAC és Linux rendszereken is. Ugyanakkor biztosítva van a platform ingyenessége és jogtisztasága is.

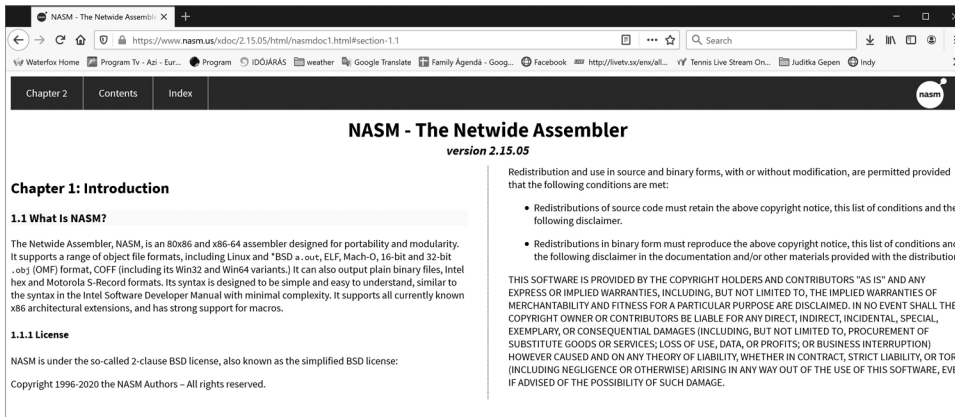
3. ábra. A Lazarus Objective Pascal IDE környezet



[23] „Lazarus – The professional Free Pascal RAD IDE,” [Online]. Available: <https://www.lazarus-ide.org/>. [Hozzáférés dátuma: 15. 07. 2021.]

[24] „NASM,” [Online]. Available: <https://www.nasm.us/>. [Hozzáférés dátuma: 15. 07. 2021.]

4. ábra. A NASM Netwide Assembler környezet



A Forth-rendszer használata a modern operációs rendszerekben két fő problémába ütközik. Az első, hogy az 1994-es szabvány kb. 200 Forth-utasítást tartalmaz. Ezek egytizedének az ismeretében el lehet kezdeni a Forth használatát bevezető alapszintű problémák megoldására, ami egy újonc számára is elfogadható sőt, megfelelő útmutatással vonzó is lehet. Természetesen később a programozó használni szeretné az operációs rendszer által hozzáférhetővé tett technológiákat is úgy, mint internet, grafika, USB bővítések stb. Amint azt fennebb említettük, létező Forth-rendszerek ezt az alapszótár bővítésével oldják meg, mint például a Winforth32, ami évekkel ezelőtt is 5000+ utasítást tartalmazott. Ez egyértelműen elbátortalanítja még a lelkes érdeklődőket is.

5. ábra. Példák szókészlet-bővítésekre

```

Nucleus layer

! * */ */MOD + +! - / /MOD 0< 0= 0> 1+ 1- 2+
2- 2/ < = > >R ?DUP @ ABS AND C! C@ CMOVE
CMOVE> COUNT D+ D< DEPTH DNEGATE DROP DUP EXECUTE
EXIT FILL I J MAX MIN MOD NEGATE NOT OR OVER PICK
R> R@ ROLL ROT SWAP U< UM* UM/MOD XOR

Device layer

BLOCK BUFFER CR EMIT EXPECT FLUSH KEY SAVE-BUFFERS
SPACE SPACES TYPE UPDATE

Interpreter layer

# #> #S #TIB ' ( -TRAILING . .( <# >BODY >IN
ABORT BASE BLK CONVERT DECIMAL DEFINITIONS FIND
FORGET FORTH FORTH-83 HERE HOLD LOAD PAD QUIT SIGN
SPAN TIB U. WORD

Compiler layer

+LOOP , , ." : ; ABORT" ALLOT BEGIN COMPILE CONSTANT
CREATE DO DOES> ELSE IF IMMEDIATE LEAVE LITERAL LOOP
REPEAT STATE THEN UNTIL VARIABLE VOCABULARY WHILE
['] [COMPILE] ]

```

6. ábra. Példa szókészlet-bővítésre

The screenshot shows the Win32Forth 6.14.02 window. The menu bar includes File, Edit, Display, Tools, Macros, and Help. The main area displays a list of words and constants in a grid-like format. The words are listed in columns, and the constants are listed in columns. The words include: 0<, WORD-SPLIT, D2*, 2/, UNDER+, BIT-POP, CDECR, RSHIFT, OR, FILL, CHAR+, CELL-, CELLS, W@, +!, 2R>, DUP>R, RF!, 4REVERSE, TUCK, ROT, DROP, _CASE, (LOOP), J, _WHILE, _THEN, BRANCH, NEXT-SEQ, MAXSTRING, NULL, -CELL, 0, EXECUTE, Douser, DO2VALUE, DOCON, 0<>, UM/MOD, 1-, 2*, TOGGLE, CINCRC, ISHIFT, AND, MOVE, CHARS, CELL+, W+!, C+!, 2>R, R@, RF@, S-REVERSE, NIP, OVER, ENDCASE, BOUNDS, ?LEAVE, I, UNTIL, _BEGIN, CALLF, CRIF, MAX-HANDLE, FALSE, CELL, -1, LIT, DO2VALUE!, DODEFER, DOCOL, 0=, UM*, 1+, ABS, OFF, DECR, INVERT, BLANK, CMOVE>, -CELLS, CELLS-, W!, C!, @, RDROP, R>, SF!, S-REVERSE, ?DUP, SWAP, ENDOF, UNLOOP, LEAVE, (DO), -?BRANCH, RETURNF, MAXCOUNTED, #VOCs, TRUE, 2, NOOP, UNNEST, DO2VALUE!, DOWAR, WORD-JOIN, D2/, U2/, NEGATE, BIT-MSB, ON, INCR, XOR, ERASE, CMOVE, +CELLS, CELLS+, SW@, C@, 2R@, R>DROP, >R, SF@, PICK, -ROT, DUP, OP, (+LOOP), K, (DO), AGAIN, ?BRANCH, EXEC-SEQ, MAX-PATH, PTRNULL, BL, 1, PERFORM, DOOFF, DO2VALUE, DODOES.

Displayed 2345 of the 6200 words in the system.
 ** Use: WORDS <substring> to limit the list **
 ** Use: CONSTANTS <substring> to display Windows Constants **

[18] D. Bruehl: „*Forth for Education – 4E4th and 4E4th IDE*,”

[Online]. Available:

<http://www.com-plang.tuwien.ac.at/anton/euroforth/ef12/papers/bruehl.pdf>.

[Hozzáférés dátuma:

15. 07. 2021.]

7. ábra. Példa szókészlet-bővítésre

```
TASK CLS FORTH \ WHILE ELSE IF REPEAT AGAIN END UNTIL
+LOOP LOOP DO THEN ENDIF BEGIN BACK BYE VLIST --> LOAD
BLOCK U. ? . D. .R D.R #S # SIGN #> <# SPACES RBUF
M/MOD */ *MOD MOD / /MOD * M/ M* MAX MIN DABS ABS
S->D COLD ABORT QUIT ( DEFINITIONS FORGET ' VOCABULARY
IMMEDIATE INTERPRET ?STACK U< DLITERAL LITERAL [COMPILE]
CREATE ID. ERROR (ABORT) UPPER -FIND NUMBER (NUMBER) WORD
PAD HOLD BLANKS ERASE FILL QUERY EXPECT ?ALIGN ."
(.) -TRAILING TYPE COUNT DOES> <BUILDS (;CODE) OCTAL
DECIMAL HEX SMUDGE ] [ COMPILE ?LOADING ?CSP ?PAIRS ?
EXEC ?COMP ?ERROR !CSP PFA NFA CFA LFA LATEST TRAVERSE
-DUP 2DROP 2DUP SPACE ROT > < = - , ALLOT HERE 2+ 1+
ATTR FILE PREV USE HLD R# CSP FLD DPL BASE STATE
CURRENT CONTEXT OFFSET SCR OUT IN BLK LIMIT FIRST VOC-
LINK DP FENCE WARNING WIDTH TIB R0 S0 +ORIGIN B/SCR
B/BUF C/L BL 3 2 1 0 USER VARIABLE CONSTANT ; : C! !
C@ @ TOGGLE +! DUP SWAP DROP OVER MINUS 0< 0= R R>
>R LEAVE ;S RP! SP! SP@ XOR OR AND U* U/ CMOVE KEY
TRKEY CR ?TERMINAL GOTOXY EXCALL EMIT ENCLOSE (FIND)
DIGIT I (DO) (+LOOP) (LOOP) 0BRANCH BRANCH D+ + EXECUTE
LIT OK
```

A másik probléma pedig az, hogy a mai grafikus felületű operációs rendszerek alapfüggvényei, mint például az ablakkezelés nem ritkán 10, vagy akár több paramétert is használnak. Ugyanakkor: „a Forth-programozás jelenlegi trendje az, hogy kerüljük a verem mélyebbé tételét 3 vagy 4 elemnél. Az ennél mélyebb verem általában annak a jele, hogy a programot nem megfelelően gondolták át.” [18] Még ha le is szűkítenénk a szótár bővítését a minimumra, a túl sok paraméter használata a verem átláthatóságát csökkenti és ezáltal a Forth programozási filozófia lényegének szem elől vesztesét okozza.

A továbbiakban leírt módszer mindkét akadályt áthidalja, egy adott problémakörhöz a RAD-rendszerben megírt általában néhány, szó szerint pár sornyi modul és a hozzá tartozó egységes Forth-interfész révén.

Példák

A továbbiakban bemutatunk néhány, a már említett Forth-fejlesztésben készített példaprogramot. Sajnos a kódok által megvalósított dinamikus struktúrákat itt csak néhány statikus ábrával tudjuk illusztrálni.

1. HANOI TORNyai

```
: move1 swap . . cr ;
```

```
: move1 213 excall swap . . cr ;
```

```
0 variable h_addr
```

```
: hanoi dup 1 = if drop move1 else
```

```
>R over over over + 6 swap - R 1 - h_addr @ execute
```

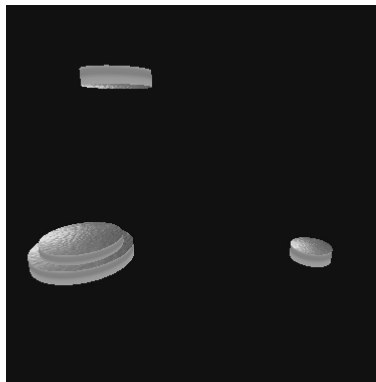
```
over over move1
```

```
swap over + 6 swap - swap r> 1 - h_addr @ execute endif ;
```

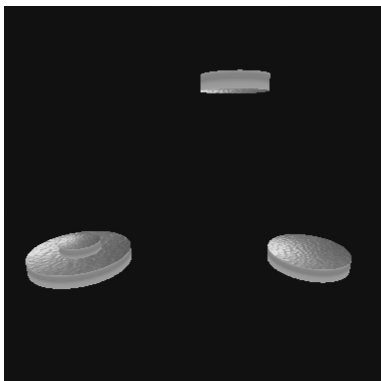
```
` hanoi cfa h_addr !
```

1 sor hozzáadásával (over over move1) szemléletes animációt hoztunk létre:

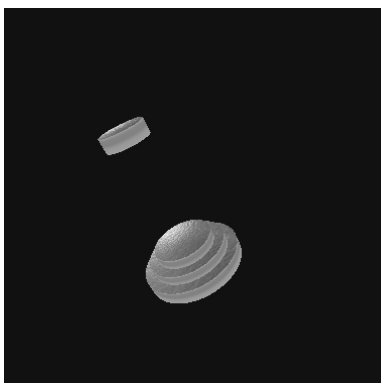
8. ábra. Hanoi tornyai 1.



9. ábra. Hanoi tornyai 2.



10. ábra. Hanoi tornyai 3.



2. PI KISZÁMÍTÁSA

Bár a Forth-rendszer egész számokkal dolgozik, ez a példa a Pi, a kör kerület/átmerő arányának (π) értékét számítja ki többszáz tizedesnyi pontossággal, Euler képlete alapján:

11. ábra. Pi kiszámítása

$$\frac{\pi}{4} = \arctan \frac{1}{2} + \arctan \frac{1}{3}$$

$$\arctan x = \sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} x^{2n+1} = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \dots$$

```

1000 constant v_l 0 variable ind
here v_l 4 * allot constant vtempa here v_l 4 * allot constant vtempb
here v_l 4 * allot constant at1/2 here v_l 4 * allot constant at1/3
: v@ 4 * r# @ + @ ; : v! 4 * r# @ + ! ;
: v_zero v_l 0 do i 4 * over + 0 swap ! loop drop ;
: v_print dup @ 1 d.r 46 emit v_l 1 do i 4 * over + @ 1 d.r loop drop ;
: c/ swap r# ! 0 v_l 0 do 10 * i v@ + over /mod i v! loop drop ;
: c* swap r# ! 0 v_l 0 do over v_l i - 1 - v@ * + 10 /mod swap v_l i - 1 - v! loop drop ;
: v+ swap r# ! 0 v_l 0 do v_l i - 1 - v@ swap + over v_l i - 1 - 4 * + @ + 10 /mod
swap v_l i - 1 - v! loop drop ;
: v:= swap r# ! v_l 0 do dup i 4 * + @ i v! loop drop ;
: v- swap r# ! 0 ind ! begin ind @ v@ 0= while 1 ind +! repeat
v_l 1 - ind @ do i v@ 1 - i v! 10 i 1+ v@ + i 1+ v! loop
0 v_l 0 do v_l i - 1 - v@ + over v_l i - 1 - 4 * + @
- 10 /mod swap v_l i - 1 - v! loop drop ;

vtempa v_zero 5 vtempa 4 + ! at1/2 vtempa v:=
: c_at1/2 100 1 do vtempa 4 c/ vtempb vtempa v:= vtempb i + 1+ c/ at1/2 vtempb
i 2 mod if v- else v+ endif loop ; c_at1/2

```

at1/3 v_zero 1 at1/3 ! at1/3 3 c/ vtempa at1/3 v:=

: c_at1/3 100 1 do vtempa 9 c/ vtempb vtempa v:= vtempb i i + 1+ c/ at1/3 vtempb
i 2 mod if v- else v+ endif loop ; c_at1/3

at1/2 at1/3 v+ at1/2 4 c* at1/2
cr cr ." PI = " v_print cr

Végrehajtás:

PI =

3.141592653589793238462643383279502884197169399375105820974944587
34437321538295376117614603945866419742307357130104932178994142901
91677721975482080111611784213707346556916072874749797963710806365
38111206230405528218070576329057925402081947421780511069476458023
99932921571046675475420374232180268792410134691659518585213598449
20767358804946747345712030063025145576704396915548362245833830913
73466398544594682380526838178493318223474814725796671080713297390
95166926844160375748446628322025101603352391239010539656295460932
05160399457031789098301143022696362323878213580456449260048604264
48031168494909216983899518708064610176494193661024380034854727866
33316928861193511654811079070811452536787953713999315319803361006
13022145585505207297124366836230175554369288524732676098812050165
77871362324685875034470906652738835934399526451762206844401493563
41202450789031429478911954474499001931340623868227425066397982009
86876517158640619357175225145973785962598604299127767657507236819
29472090952787340999460004

OK

3. 'TEKNŐC' GRAFIKA 3D VÁLTOZATA

Ez a bővítés további fejlesztési lehetőségeket kínál, mint például robotkar interaktív vezérlése, mértani hely problémák szemléltetése stb.

Lévén ez egy aránylag bonyolult bővítés, ezért a teljes implementációt bemutatjuk illusztrálандó, hogy ez esetben is relatíve kevés programsorral megoldható a Forth 3D utasításkészlet létrehozása.

Pascal OpenGL implementáció (a tényleges 3D utasítások végrehajtása):

```
-----
x3ang:=1; y3ang:=0; z3ang:=0;x3poso:=0;y3poso:=0;z3poso:=0;
if nrcmd>0 then
for i:=0 to nrcmd-1 do begin // dec
  if cmd[i]=5 then begin // ~moveto
x3poso:=x3poso+cparam[i]*x3ang; y3poso:=y3poso+cparam[i]*y3ang;
z3poso:=z3poso+cparam[i]*z3ang;
end;

  if cmd[i]=10 then begin // dirx
    x3ang:=1; y3ang:=0; z3ang:=0;
  end;
  if cmd[i]=8 then begin // ~moveto
    x3poso:=xparam[i]; y3poso:=yparam[i]; z3poso:=zparam[i];
  end;

  if cmd[i]=7 then begin // ~lineto
    glBegin(GL_LINES);
    glVertex3d(x3poso,y3poso,z3poso);
    dist:=sqrt((x3poso-xparam[i])*(x3poso-xparam[i])+(y3poso-yparam[i])*(y3poso-yparam[i])+(z3poso-zparam[i])*(z3poso-zparam[i]));
    x3ang:=-x3poso-xparam[i])/dist;y3ang:=-y3poso-yparam[i])/dist;z3ang:=-z3poso-zparam[i])/dist;
    x3poso:=xparam[i]; y3poso:=yparam[i]; z3poso:=zparam[i];
    glVertex3d(x3poso,y3poso,z3poso);
    glEnd();
  end;
```

```

if cmd[i]=4 then begin    // ~lin
  glBegin(GL_LINES);
  glVertex3d(x3poso,y3poso,z3poso);
  x3poso:=x3poso+cparam[i]*x3ang; y3poso:=y3poso+cparam[i]*y3ang;
  z3poso:=z3poso+cparam[i]*z3ang;
  glVertex3d(x3poso,y3poso,z3poso);
  glEnd();
end;

if cmd[i]=3 then begin // rz
  ang:=arctan(y3ang/x3ang)+pi*2.0*cparam[i]/360;
  if x3ang<0 then ang:=pi+arctan(y3ang/x3ang)+pi*2.0*cparam[i]/360;
  rrt:=sqrt(y3ang*y3ang+x3ang*x3ang);
  x3ang:=rrt*cos(ang);y3ang:=rrt*sin(ang);
end;

if cmd[i]=2 then begin // ry
  ang:=arctan(x3ang/z3ang)+pi*cparam[i]/180;
  if z3ang<0 then ang:=pi+arctan(x3ang/z3ang)+pi*cparam[i]/180;
  rrt:=sqrt(z3ang*z3ang+x3ang*x3ang);
  z3ang:=rrt*cos(ang);x3ang:=rrt*sin(ang);
end;

if cmd[i]=1 then begin // rx
  ang:=arctan(z3ang/y3ang)+pi*2.0*cparam[i]/360;
  if y3ang<0 then ang:=pi+arctan(z3ang/y3ang)+pi*cparam[i]/180;
  rrt:=sqrt(y3ang*y3ang+z3ang*z3ang);
  y3ang:=rrt*cos(ang);z3ang:=rrt*sin(ang);
end;

```

Pascal oldalról a Forth interfész:

```

case gparam of
1011 : begin Image1.left:=OpenGLControl1.left; // cl

```

```

    Image1.top:=OpenGLControl1.top;Image1.height:=384;Image1.width:=384;
    OpenGLControl1.visible:=false; image1.visible:=true;
    image1.canvas.fillrect(classes.rect(0,0,384,384));
    image1.canvas.pen.width:=3;
    image1.canvas.moveto(150,150); angle:=0;xpos:=150;ypos:=150;
//    image1.canvas.lineto(200,300);
end; // cl

1012 : begin angle:=angle+s0; end; //RT

1014 : begin //FD
    xpos:=round(s0*sin(pi*2.0*angle/360))+xpos;
    ypos:=round(s0*cos(pi*2.0*angle/360))+ypos;
    image1.canvas.lineto(xpos,ypos); delay(TrackBar1.position*100);
end; // FD

1018 : begin //plot
    image1.canvas.pixels[s0,s4]:=clred;
end; // PLOT

1050 : begin // rdw
    DrawGLScene();
    OpenGLControl1.SwapBuffers;
    sleep(200);
end;

1051 : begin // cl
    nrcmd:=0; x3ang:=1; y3ang:=0; z3ang:=0;x3poso:=0;y3poso:=0;z3poso:=0; end;

1052 : begin // lin
    cmd[nrcmd]:=4;cparam[nrcmd]:=s0 /100; nrcmd:=nrcmd+1; end; //

1053 : begin // mov
    cmd[nrcmd]:=5;cparam[nrcmd]:=s0 /100; nrcmd:=nrcmd+1; end; //
1054 : begin // rz

```

```

cmd[nrcmd]:=3;cparam[nrcmd]:=s0/100; nrcmd:=nrcmd+1; end; //

1055 : begin // ry
cmd[nrcmd]:=2;cparam[nrcmd]:=s0/100; nrcmd:=nrcmd+1; end; //

1056 : begin // rx
cmd[nrcmd]:=1;cparam[nrcmd]:=s0/100; nrcmd:=nrcmd+1; end; //

1057 : begin // lineto
cmd[nrcmd]:=7;xparam[nrcmd]:=s0/100;
yparam[nrcmd]:=s4/100; zparam[nrcmd]:=s8/100; nrcmd:=nrcmd+1; end; //

1058 : begin // moveto
cmd[nrcmd]:=8;xparam[nrcmd]:=s0/100;
yparam[nrcmd]:=s4/100; zparam[nrcmd]:=s8/100; nrcmd:=nrcmd+1; end; //

1060 : begin // dirx
cmd[nrcmd]:=10; nrcmd:=nrcmd+1; end; //

end; // case
//sleep(100);
application.processmessages;
-----
Forth 3D bővítés interfész:
: rdrw 1050 excall ;
: cl 1051 excall ;
: lin 1052 excall drop ;
: mov 1053 excall drop ;
: rx 1056 excall drop ;
: ry 1055 excall drop ;
: rz 1054 excall drop ;
: lineto 1057 excall drop drop drop ;
: moveto 1058 excall drop drop drop ;
: dirx 1060 excall ;

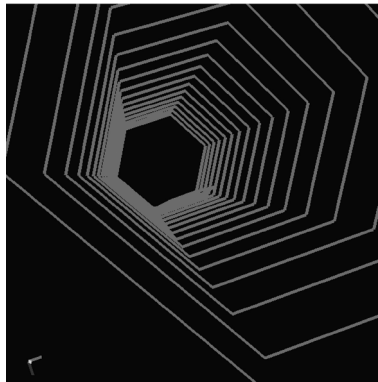
```


Mivel elég nehéz 3D modelleket szövegben leírni, ezért a következőkben képekkel illusztráljuk a lehetőségek egy részét. A létrehozott térbeli modellek egér segítségével szabadon forgathatók, ill. görgetéssel nagyítás/kicsinyítés érhető el. Kiemelnénk a képek bal alsó sarkában látható RGB koordinátarendszert, ami a mindenkori abszolút XYZ irányokat mutatja. Ez nagyban megkönnyíti a térbeli érzékelést (ugyanakkor sok kereskedelmi 3D programból is hiányzik).

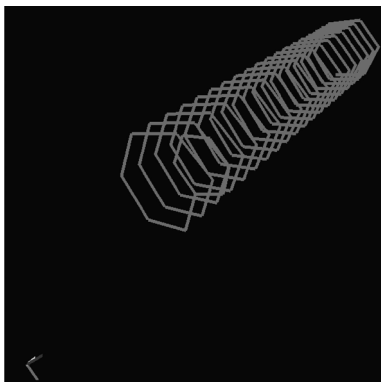
3.1 statikus 3D ábra

```
: aa 6 0 do 6000 rz 160 lin loop  
500 rz  
116 0 do 6000 ry 260 lin loop ;
```

12. ábra. 'Teknőc' grafika 3D változata – statikus 3D ábra



13. ábra. 'Teknőc' grafika 3D változata – statikus 3D ábra

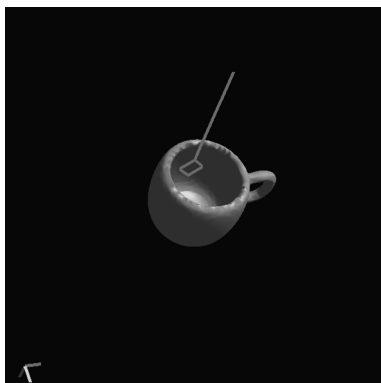


3.2 Animált 'vektor kiskanál'

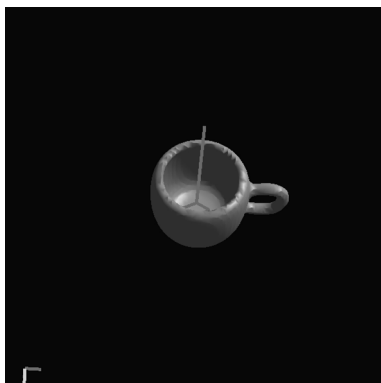
```
: spoon // 9000 ry 100 mov -9000 ry  
4500 ry 40 lin 9000 ry 40 lin  
-4500 ry 250 lin -250 mov 13500 ry  
40 lin 9000 ry 40 lin ;
```

```
: stir 4 0 do  
0 0 -50 moveto spoon dirx rdrw cl  
0 -50 0 moveto spoon dirx rdrw cl  
0 0 50 moveto spoon dirx rdrw cl  
0 50 0 moveto spoon dirx rdrw cl  
loop ;
```

14. ábra. Animált 'vektor kiskanál' 1.



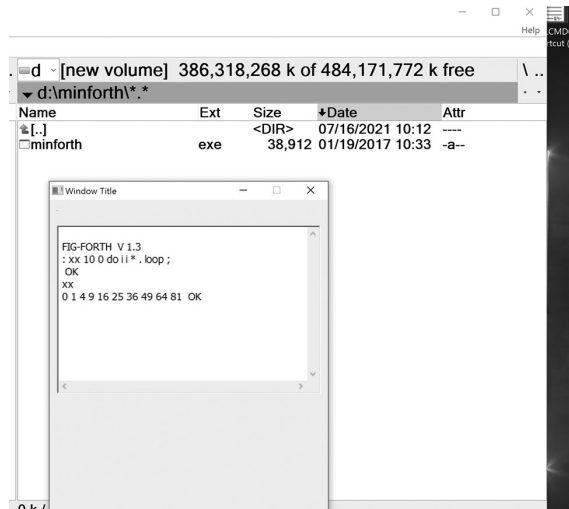
15. ábra. Animált 'vektor kiskanál' 2.



TOVÁBBI LEHETŐSÉGEK

Befejezésül megemlítenénk még egy fejlesztés alatt álló irányvonalat. Mivel egy Forth-rendszer szerves része a beépített fordító is, ezért kiválóan alkalmas evolúciós programok létrehozására. Az alapötlet viszonylag egyszerű: több ezer, vagy akár tízezer, azonos algoritmuson alapuló, de véletlenszerű módosításokat tartalmazó program egyidejű futtatása, majd ezek szűrése különböző kritériumok alapján. Elméletileg, igen sok generáció után bonyolult problémákat megoldó programok is létrejöhetnek. Ebben az irányban haladva hoztuk létre a minimális Forth-rendszert, amely egy 40k alatti EXE, ami lehetővé teszi egy otthoni gépen is a több ezer program-példány futtatását (16. ábra).

16. ábra. A minimális Forth rendszer



Köszönetnyilvánítás

A közlemény megjelenését az EFOP-3.6.1-16-2016- 00003 „K+F+I folyamatok hosszú távú megerősítése a Dunaújvárosi Egyetemen” című projekt támogatta.

Az érintés nélküli adatgyűjtés kockázatai és a kockázatszámítás módszerei

Összefoglalás: Bár egyre fejlettebb IT szabályok, technológia és kibervédelmi szakember csapat támogatja virtuális világunkat, napjaink részévé vált a világ különböző részeiről érkező támadási kísérletek észlelése és az incidentsek megakadályozása. Immár egyre több közszolgálati intézmény igénye az érintés nélküli megoldások alkalmazása a védelmi technológiában is. Egyes törekvések szerint a térfigyelő kamerák adatbázisának összekötése a hivatali, védelmi vagy akár a banki adatbázisokkal elengedhetetlen, hiszen az adatokat akár egy egységes bűnüldözői rendszerben is felhasználhatnák. Egyes országok kormányzatának álláspontja szerint a személyek köztéri valós idejű azonosítása visszaszoríthatja a bűnözést és a szigorúbb azonosítási rendszerrel csökkenthető az internetes incidensek száma. Az igény felvet néhány kérdést, vajon hogyan hat az esetlegesen túlzott ellenőrzés a többi, rendes életet élő polgárra? Valóban túlzott ez az ellenőrzés, vagy létezik olyan ország, ahol elengedhetetlenül szükséges és immár hatással van a többi országra is? Hogyan gyűjtik és használják fel a biometrikus adatokat és mi a célja? Milyen korlátai és kihívásai vannak a biometrikus adatgyűjtésnek? Kockázatkezelés szempontjából mekkora jelentőséggel bír az esetlegesen kibertámadásnak jobban kitett személyes adat? A fizikai határok kezdenek elmosódni, különböző típusú embercsoportok vándorolnak szerte a világban. Szinte pár óra alatt képesek vagyunk eljutni a világ egyik végéből a másikba, mivel a virtuális világban nincsenek határok, így a hasznos és a haszontalan vagy a kárt okozó üzenetek, programok vagy információk jórészt határok nélkül közlekedhetnek. IT eszközeinket egyre többféle védelemmel kell biztosítani, így védeni kell a szervereket, az alkalmazásokat, a routereket, a végpontokat és a végpontokon futó alkalmazásokat valamint a tárolt és éppen úton lévő adatokat egyaránt. A közterületen elhelyezett vagy az informatika által nyújtott, biometrikus adatokat rögzítő és továbbító, érintés nélküli eszközök és kommunikációjuk védelmét,

* Nemzeti Közzolgálati Egyetem
– Pannonia Egyetem Veszprém
E-mail: gyorffy.kriszta@gmail.
com

valamint az információbiztonsági garanciát sokszor multicégek vállalják. Egyes technológiákban az alkalmazások sikerességének titka akár az emberi arcok felismerése, az arcok mögött valós személyek tulajdonosságának beazonosíthatósága és egy, a valós személy élethű másolata, ahol a virtuális karakter lényegesen több információval rendelkezik, mint egy retusált kép vagy egy szöveges leírás. A technológiát integrálhatják akár egy adott ország digitális azonosítási rendszerébe, amelyet összeköthetnek más hatósággal is. Így például a bűnüldözői tevékenység kiterjeszthető más funkciót ellátó szakterületre is. Számos magánvállalkozás, autógyártással foglalkozó multivállalat vagy pénzügyintézet is segítheti a projekt megvalósulását. Az incidensek számossága és az elmúlt években okozott anyagi kár mértéke is mutatja, hogy a különböző projektek együttműködése elengedhetetlen abban az esetben, ha annak felhasználása az információbiztonsági alapelvek mentén történik. A személyes adatok utaztatása egyik rendszerből a másikba megfelelő minőségű és fejlettségű informatikai technológiát és felügyeletet igényel, kiemelt bizalmi státuszokkal. Az információbiztonsági alapelvek és előírások betartása ezért rendkívül összetett feladat, amely magas szintű technikai és munkaerő adottságokat igényel. Az adottságoknak és az információbiztonsági előírásoknak való megfelelést és ellenőrzését tanúsító cégek végzik. Az audit során feltárják a hiányosságokat, a biztonsági réseket és az adott rendszer működésével kapcsolatos kockázatokat is. Az érintés nélküli eszközök adatainak tárolására szolgáló adatbázisok létrehozásának és így módon való felhasználásának árnyoldalai is vannak. A különböző IT-eszközök interoperabilitási nehézségeit leküzdő rendszerek fejlesztése során adatvédelmi kérdésekkel, etikai alapelvekkel és információbiztonsági előírásokkal kerülhetünk szembe. A bűnüldözés és a megelőzés fontossága jelentős, de a módszerek kialakítása során a technikai és információbiztonsági előírások alkalmazása, a társadalom tudatosítása, a jogi, legfőképp adatvédelmi lehetőségek feltérképezése és az előírások betartása elengedhetetlen. Jelen tanulmány az érintés nélküli adatbevitel információbiztonsági jelentőségét és az alkalmazható kockázatkezelési módszereket veszi számba.

Kulcsszavak: Érintés nélküli; információbiztonság; kockázat.

Abstract: Although increasingly advanced IT rules, technology and cyber security team are supporting our virtual world, even so it became part of our day the notice and prevention of incidents, which come from the different parts of the World. More and more governments are now requiring the use of contactless solutions in defence technology as well. According to some theory, it is essential the connection the surveillance camera database to the official, defence or other banking databases, because data could even be used in a unified law enforcement system.

According to the governments of some countries, real-time identification of people on the public area can reduce crime and the stricter identification system can also reduce the number of Internet incidents. The idea raises a question, so how does potentially excessive control affect other citizens living a normal life? Is it really control excessive, or is there a country where this care is absolutely necessary and now has it impact on other countries as well? How is the biometric data collected and used and what is its purpose?

What are the limitations and challenges of biometric data collection? From the point of view of risk management, how important is the possibility of data entry that may be more exposed to a cyber attack?

The physical boundaries are starting to blur, different types of groups of people are migrating around the World. We are able to get from one end of the world to the other in almost a few hours. There are no boundaries in the virtual world, so useful and useless or harmful messages, programs, or information can travel largely without boundaries. Our IT devices need to be protected in an increasing number of ways, including servers, applications, routers, endpoints and applications running on endpoints, and even personal information. The multi company often guarantees the information security of the contactless, data recording and sending IT devices, which are situated in a public area or which are made by robotics. In some technologies, the secret of the successful application is recognizing human faces, is identifying the characteristics of real people behind the faces and is realistic copying of a live person's character, where the virtual character has significantly more information than a retouched picture or a text description. The technology can be integrated into a country's digital identification system, which can also be linked to another authority. Thus, for example, law enforcement can be extended to other area. Some private companies, automotive multinationals firm or financial institutions can help in similar project. In recent years, the number of incidents and the extent of the caused material damage also show that the cooperation is essential between different projects, if the projects are used based on the information security principles. The transfer of personal data between two systems requires a high degree of security technology and supervision, with unique statuses of trust. Therefore following of the information security principles and standards are an extremely complex task that requires a high level of technical and manpower. Compliance, verification of capabilities, and information security standards are performed by companies, which certify. The audit are identified the gaps, vulnerabilities and risks of the information system. There are some disadvantages of creating and using databases, which store of data of contactless IT devices. During the development of IT devices, which overcame the interoperability difficulties, we are facing data protection issues, ethical principles and information security standards. Crime prevention and law enforcement are important, but the development of methods requires the application of the technical and the information security standards, the user awareness, the analysing of legal and data protection options, and the compliance with the standards. This paper is concerned with the information security and risks assessment methods of contactless data recording.

Keywords: Contactless; information security; risk.

Bevezetés

A biometrikus adatok gyűjtésével és felhasználásával a személyes adatok palettája kibővül, ezáltal többféle informatikai technológia válik elérhetővé a vállalkozások és az állami intézmények, valamint a felhasználók számára. A szükségszerű technológiai fejlődés által megjelenő új információs felületek használata feltételezi a technológiai, az adatvédelmi és az információbiztonsági minimális követelmény teljesülését. Ugyanakkor a biometrikus adatok jellegüktől fogva nem változtathatók meg, ezért a védelmük is kiemelten kezelendő. Sérülésük vagy veszélyeztetésük az érintettre vonatkozóan helyrehozhatatlan károkat okozhat. Az információbiztonsági és az adatvédelmi alapelvek és előírások meghatározzák az adatkezelés technológiai és jogi követelményét. Általában a nagyobb munkavállalói létszámmal rendelkező vállalkozások, valamint a közszolgálatot teljesítő intézmények információbiztonsági ajánlásoknak, valamint a törvényi rendelkezéseknek megfelelően szabályozzák a személyes adatkezelést. Az információbiztonsági szabályozás önmagában kevés, az információbiztonsági kockázatok feltárásával, elemzésével és kezelésével meghatározható az adott információs rendszer információbiztonsági szintje, hatékonyabbá tehető az incidenskezelés. A kockázatkezelési módszerek, mint a kockázatmenedzsment eszközei segítenek azonosítani és meghatározni a rendszer gyenge pontjait, a kezelendő sebezhetőségeket, a kockázat mértékét és az elfogatható tartományt, valamint a maradványkockázat értékét.

Semmi sem eredményezi az innovációt jobban, mint a válság

Biometrikus módszereket több, mint száz éve használjuk, többek között a bűnüldözési eljárásokban. 1901-ben Angliában és Írországbán használtak először ujjlenyomatot a bűnügyi nyilvántartásokban, később 1903-ban az Amerikai Egyesült Államok, New York Állami Börtönében bevezették az ujjlenyomat-alapú azonosítást. Nagy fejlődést jelentett az FBI¹ ujjlenyomatok feldolgozására létrehozott részlege 1924-ben, amely 1946-ban már közel 100 millió ujjlenyomattal rendelkezett. 1936-ban pedig Frank Burch lefektette az íriszvizsgálat alapjait. 1960-as évektől számíthatjuk az arcfelismerő berendezéseket, mivel Bledsoe, Goldstein és Lesk munkássága alapján megszülettek az első gépi arcfelismeréssel kapcsolatos eredmények. Ugyanakkor azt is el kell mondani, hogy a biometrikus módszerek elterjedésének korai szakaszában a pontos meghatározás alulmaradt a vártnál, nagy volt a hibalehetőség is. 1965-ben létrejött az első automatikus aláírás-ellenőrző rendszer, de csak az 1977-es szabadalmazást követően fejlesztették tovább. 1974-ben megjelent az első kereskedelmi forgalomba állított kézgeometria-olvasó, és 1976-ban kifejlesztették a hangalapú azonosítás rendszerét.

Később, 1998-ban bevezetésre került a DNS-alapú azonosítás és 1994-ben magyar fejlesztés alapján megalkották az első tenyérleNyomatot használó rendszert. Arcfelismerő rendszert 2001-ben telepítettek először az Egyesült Államok egyik stadionjában, majd a 2001. szeptember 11. után az amerikai hatóságok az összes reptérre biometrikus rendszert vezettek be.² A biometrikus módszerek kifejlesztése és alkalmazása nagy utat tett meg mostanáig, ugyanakkor az elmúlt években, valójában az elmúlt száz év eredményeire alapozva szintén nagy előrelépést tapasztalhattunk meg, immár a pontos és gyors azonosítást igénylő érintés nélküli biometrikus adatgyűjtés terén. A Covid-19 világjárvány következtében az érintés nélküli technológia, a biometrikus adatgyűjtést alkalmazó rendszerek használata iránt megnőtt a kereslet. Az egyik legújabb és legszembetűnőbb érintés nélküli azonosításra szolgáló rendszer a norvég reptereken alkalmazott biztonságos áthaladást segítő technológia, amely biztosítja a fizikai kontaktus, személyzettel vagy beléptető berendezésekkel való elkerülését a poggyászfeladástól kezdve a beszállásig. A norvég Avinor vállalat ezzel az intézkedéssel is szeretne volna növelni az utasok bizalmát, akik már az online regisztrálás során megkapják a beszállókártyát. Az intelligens poggyászfeladó automata segítségével elkerülhető a személyes kontaktus. Budapesten, a Budapest Airport is alkalmaz hasonló technológiát. Egyes repterek robotok használatával is kísérleteznek, mint például a dél-Koreai Incheon nemzetközi repülőtér egy utasokat arcmaszk viselésére emlékeztető robot tesztelését végzi. A robot rögzíti és továbbítja azon személyek arcáról készített felvételt, akik nem teljesítik a hatóság által előírt feltételeket. Biometrikus adatgyűjtés alapú IT-rendszerek tesztelésben például Németország, Kanada, Kína, Japán, Izland, Olaszország, Spanyolország is részt vesz, ezáltal jelentősen felgyorsítják az ellenőrzési folyamatot.³ További kezdeményezés, hogy az utasok egy speciális rendszerbe feltöltik a szelfiket, az arcukról készített képet, a hitelesített útlevéllel ezt ellenőrzik, majd a beszállókapunál egy képernyő segítségével megtörténik az azonosítás és beszállhatnak. Az automatizált határellenőrzési rendszer valószínű a Covid-19 után is velünk marad.

További érintés nélküli megoldások születnek abból a célból, hogy biztosítsák a személy és vagyonvédelmet:

- társadalmi távolságtartás mérése a reptereken és egyéb forgalmas helyeken,
- arcmaszkok viselésének detektálása,
- testhőmérséklet-mérés,
- légszűrő rendszerek alkalmazása, fertőtlenítés,
- utasérkeztetés,
- szigorúbb takarítási eljárások alkalmazása.

A járványhelyzet idején a mobilalkalmazások használata elősegíti a szükséges személyes kapcsolattartás csökkentését az oktatási intézményekben, a munkahelyeken, valamint a hivatali ügyintézés során, tehát az applikációkkal az ügyintézési és a személyes érintkezési kötelezettségek a minimálisra redukálhatók.

A mesterséges intelligencia technológiáját alkalmazó rendszerek esetében bármely érintés nélküli vezérlés hozzáadható minden egyes kiszolgáló eszközhöz, amely további fejlesztési lehetőséget biztosít. Ilyen például az emberi reakció érzékelése, feldolgozása és értelmezése, mint például a mozgáskorlátozott személyek esetében a fej mozgásával irányítható az adott önkiszolgáló eszköz. A hangfelismerő eszközök fejlesztésével, egyedi hangok, hangminták gyűjtésével, alkalmazásával és az egyének felismerésével betegellátó intézmények munkája is támogatható, vagy forgalmas helyeken az adott egyén hangja felismerhető. A gyors azonosításra szolgáló technológia segíthet megakadályozni a kórokozók terjedését a repülőtereken, ezáltal a közlekedés is biztonságosabbá válik. A biometrikus adatok tárolása, feldolgozása és felhasználása különböző célokat szolgálhat. Az adatok használhatók kereskedelmi szolgáltatások (pénzfelvétel, hozzáférés-ellenőrzés, számlázás, vásárlás, beléptetés) nyújtására, hatósági (személyazonosság igazolása, határátlépés, katonai programok, adathozzáférés) alkalmazások használatára, bünygyi és bírósági ügyek intézése és eljárások lefolytatására (halottazonosítás, bűnözők felderítése és azonosítása, emberi és vérszerinti kapcsolatok meghatározására, elveszett személyek felderítésére, büntetővégrehajtásra és nyomkövetésre: börtön, javítóintézet, házi őrizet). Az előbb felsoroltakon kívül számos helyen magán és közérdekből egyaránt alkalmaznak különböző biometrikus adatok detektálására információs eszközöket, így videó- és hangrögzítő, arcfelismerő, érintés nélküli ujjlenyomatolvasó, retinabeolvasó és felismerő rendszereket, amelyek alkalmasak a személyes adatok (arcpontok, kézlenyomat, hang, mozgás) rögzítésére és továbbítására, feldolgozás és további felhasználás céljából. Az alkalmazási területeket többféleképpen csoportosíthatjuk, mint például videófelügyelet (személy és vagyonvédelem céljából), egészségügy, illetve gyógyítás (orvosi kezelések, kórtörténet meghatározása, személyazonosítás, egészségügyi rendszerhasználat során azonosítás), munkáltatói ellenőrzés (munkaidőnyilvántartás: be- és kilépés ellenőrzése, azonosítás, rendszerhozzáférés ellenőrzés) céljából alkalmazott információs eszközök.

Biometrikus adatok rögzítésének és tárolásának korlátai és kihívásai

A reptereken használt biometrikus rendszerek magukban foglalják a személyhez kapcsolódó regisztrációt, azonosítást és a tanúsítást, amellyel egy bizonyos személy azonossága igazolható. Az azonosítási folyamat alkalmazásával különböző információs rendszerek együttműködését, interoperabilitását biztosítani kell, így a hatósági és a reptéri adatbáziskezelő valamint alkalmazásrendszerek közti folyamatok biztosítását, a feladatainak biztonságos elvégzését és a kommunikáció biztonságát. A tudomány jelenlegi állása alapján az ujjlenyomat, az írisz, az arcvonalak vagy az emberi hang tulajdonsága ugyan megváltoztatható, de nagyon költséges és bonyolult orvosi módszer segítségével, ezért a biometrikus adatok kicserélése tömeges formában a jelenlegi orvosi gyakorlat szerint nem megoldható. Tehát a biometrikus adatok tárolása, kezelése és továbbítása különleges biztonsági védelmet kíván, mivel azok egyediek, nem lecserélhetők és nem pótol-

hatók. Ugyanakkor egy személyi igazolványszám vagy egy információs rendszer felhasználói adata bármikor megváltoztatható. A biometrikus rendszerek iparági szakértői az érintés nélküli biometrikus adatokat kezelő rendszerek három formáját látják reálisnak: az arc-, írisz-felismerést vagy ujjlenyomat-azonosítást.

Az arcfelismerő rendszerek jellemzően négy komponensből állnak:

- képi rögzítésre alkalmas fényképezőgép,
- algoritmus az arc template létrehozására,
- adatbázis a képek tárolására,
- algoritmus a képek összehasonlítására (a készített képre és a tárolt képekre vonatkozóan).

Az arcfelismerő rendszerek működését az úgynevezett gépi tanulás (MI), a mélytanulás (deep learning) és a neutrális hálózatok technikai támogatják.⁴ A Google Picasa képszerkesztő szoftvere például arcfelismeréssel, nevekkel címkézi az adatbázisba betöltött fotókat. A Facebook szintén arcfelismerési funkciókat használ a kapcsolatok azonosításához és képes 98%-os pontossággal azonosítani a fényképeken szereplő személyeket az adatbázisuk segítségével. Az Amazon Prime Photos szoftvere nemcsak az arcokat vagy a tárgyakat ismeri fel, hanem az arcanalízis technológiával az egyének érzelmi állapotát tudja elemezni, úgymint a száj, szemöldök állása, tehát az egyén mosolyog vagy sem. Az Apple 3D arczszkenelő funkciója ujjlenyomat-beolvasó helyett az arcfelismerő technológiát használja a bejelentkezéshez. Bár az emberi arc alakja, vonalai változnak, az arcfelismerő rendszereket ki kell egészíteni az öregedési folyamat által létrehozott változásokkal, illetve annak valószínűségével. További hibátényező lehet az arc megvilágítása, az éppen aktuális arckifejezés, vagyis mimika, a szemüvegviselés és az elmosódás.

Az érintés nélküli ujjlenyomat-technológia (CFT) használata potenciális előnyökkel jár az állami és a magánfelhasználók számára. A CFT-technológiák nagymértékben javíthatják a LFT által eddig begyűjtött adattartalom minőségét és drasztikusan növelhetik a gyűjtemény méretét is. Előnye az előző technológiához képest a gyorsabb és higiénikus adatrögzítés, mivel nincs szükség a fizikai érintésre. A CFT alkalmazása során 3D ujjlenyomat-kép készül, ami a korábbi tinta, papír, lemez, optika vagy érzékelő felület-alapú technológiával szemben hatékonyabbnak tűnik. A 3D technológia kiküszöbölheti a 2D technológiák által okozott eredmények torzulását, a szennyeződések (úgymint tintaelfmosódás, papíron vagy érintőképernyőn lévő szennyeződés) által okozott hiányosságokat, ezáltal gyorsabb a feldolgozás és hatékonyabb az azonosítás is. Az FBI által használt adatbázisából az ujjlenyomat-egyeztetés algoritmusát alkalmazó NGI⁵ rendszere a kezdetekben 92%-ban, ma már viszont 99,6%-os arányban azonosíthatók az egyének.⁶ Hasonló arányban változott az arcillesztési és –felismerési technológia eredményessége is, amely 92%-ról 97,5%-ra nőtt. A CJIS adattár központi rendszeréhez online kapcsolódó mobil azonosító eszköz gyors ujjlenyomat keresésre szolgál, amivel a tisztviselők másodpercek alatt megkaphatják az egyén összes elérhető adatát. Ez a RISC gyorskereső szolgáltatás hatékonysága többek között abban rejlik, hogy képes ke-

vesebb, mint 10 mp alatt adatot szolgáltatni. Az adattár hozzáfér a Nemzeti Bűnügyi Információs Központ adattartalmához (1. ábra) is, beleértve az elítéltek, jogsértők, terroristák, vagy gyanúsítottak, illetve eltűnt személyek adatbázisát.

1. ábra. FBI, NGI adatai, 2021.⁷

March 2021 Next Generation Identification (NGI) System Fact Sheet

NGI System Daily Records					
Receipts	Totals	Date	Closeouts	Totals	Date
Receipts	364,617	06/25/2015	Closeouts	363,907	06/25/2015

Unless noted otherwise, the Receipt and Closeout numbers provided in this monthly report are based on Tenprint Fingerprint totals only, and do not include transactions involving other biometric types.

NGI System Fingerprints in the Identification and Slap Segmentation Fingerprint (idFP) Repository	
Repository Type	Repository Count for idFP Composite Fingerprints
# Civil Fingerprint Repository	59,573,538
# Criminal Fingerprint Repository	78,812,631
Repository for Individuals of Special Concern	3,536,611

an additional 14,141,473 Civil Fingerprint composites are in both the criminal and civil repositories and consist of mostly Office of Personnel Management and Military related fingerprints.

A biometrikus adatokat, úgymint ujjlenyomat, arcvonal-felismerő technikákat felhasználják a laptop, telefon bejelentkezéséhez, majd további applikációhoz, úgymint banki tranzakciókhoz, szavazásokhoz, és egyéb ügyviteli rendszerekhez.

Az elmúlt tíz évben számos olyan biometrikus azonosításon alapuló információs rendszert vezettek be, amelyek esetlegesen nem a kellő módon teljesítik a magas információbiztonsági és adatvédelmi követelményeket. Sok esetben az állami felügyeleti szerveknek nincs rálátásunk a nagyvállalatok által alkalmazott rendszertechnológiára⁸, és esetenként az egyedi technológia jogvédelem alá esik. Az állampolgárok kevésbé járatosak a technológiák megbízhatóságában, megfelelő működésében és a biometrikus adattárolásra és kezelésre vonatkozó információbiztonsági tanúsítvánnyal is rendkívül kevesen rendelkeznek.

Mivel a kormányzati szervek és a kereskedelmi szervezetek egyre inkább arra törekszenek, hogy rögzítés és adatbázisokban tárolják a biometrikus adatokat, amelyek természetükből adódóan nem változhatnak így aggodalomra ad okot, hogy a felhasználásra vonatkozó előírások megfelelő szigorral rendelkeznek vagy sem. Mivel a biometrikus információinak tárolására adatbázisokat hoztak létre, és a kormányzati szervek ezt egyre jobban használják, ezek az adatbázisok immár célponttá váltak, és növekszik az adatvédelmi jogsértés kockázata. Az adatvédelmi jogokkal foglalkozó szervezet, az Elektronikus Adatvédelmi Információs Központ (EPIC)⁹ felszólította a TSA-t, hogy fontolja meg a TSA Pre-Check alkalmazás biometrikus azonosítóinak gyűjtésének bővítési lehetőségeit.¹⁰ Az EPIC aggodalmát a különleges adat-

nak minősülő biometrikus adatokhoz kapcsolódó azonosítók fokozottabb fenyegetettsége miatt fejtette ki, amelyeket az előzetes ellenőrzésre való jogosultság meghatározásán kívül más célokra is használnak. Jelenleg a TSA és a kormányzat számára is rendelkezésre áll a technológia, de figyelembe kell venni a magánélet befolyásolására vonatkozó tényezőket és az adatvédelmi előírásokat, mint például az adatok tárolásának minimalizálására vonatkozó adattakarékosság elvét¹¹, valamint a kormányzati adatbázisok biztonsági kockázatait is. Mivel a biometrikus adatok elérése egyre vonzóbb a hackerek számára, ezért növelheti a jogsértések számát, és az okozott kár mértékét. Számos napi használatban lévő informatikai eszköz, tehát telefon vagy laptop használ biometrikus azonosítást, aminek védelmi technológiája rejtett a nagyközönség, illetve esetlegesen a hatóságok számára is. A hitelesítési forgatókönyv szerint a hozzáférés biztonságos, mivel a biometrikus adat a felhasználó saját mobiltelefonján tárolható, és a használata önkéntes és korlátozható, ugyanakkor a biztonság érdekében a hozzáférés és az üzemeltetés szabályain is szigorítani kell. Kormányzati szintű alkalmazások terén több millió személy biometrikus információja kerülhet veszélybe, tehát a kockázat jelentős. A biometrikus adatok minden egyes ember számára egyediek, ha ezek veszélybe kerülnek az egyének magánéletét és biztonságát örökre befolyásolja. A biometrikus adatok gyűjtését, tárolását, felhasználását és továbbítását hazai és nemzetközi rendelkezések szabályozzák. Tekintettel arra, hogy a biometrikus adat, olyan különleges adat,¹² amely egy természetes személy fizikai, fiziológiai vagy viselkedési jellemzőire vonatkozó olyan, sajátos technikai eljárásokkal nyert személyes adat, amely lehetővé teszi vagy megerősíti a természetes személy egyedi azonosítását, ezért az adatvédelmi szabályokon túl célszerű az adott információs rendszerre a kockázatkezelést elvégezni. A kockázatkezelés a kockázatmenedzsment részét képezi.

A biometrikus adatkezelés információbiztonsági, kockázatértékelési és kockázatkezelési módszerei

A kockázatmenedzsment egy proaktív módszer a biztonsági kockázatok csökkentésére és a biztonsági teljesítmény javítására. A hatékonyan végrehajtott kockázatmenedzsment dokumentált, folyamatalapú és elősegíti a biztonságos működést. A legtöbb ágazat törekedett arra, hogy a saját területén a kockázatmenedzsment folyamatát (2. ábra) alkalmazzák. A kockázatmenedzsment célja, hogy a kockázatértékelés következtében megállapított kockázat, valamint a maradványkockázat a lehető legkisebb mértékű legyen.

A kockázatmenedzsment fázisai a következők:

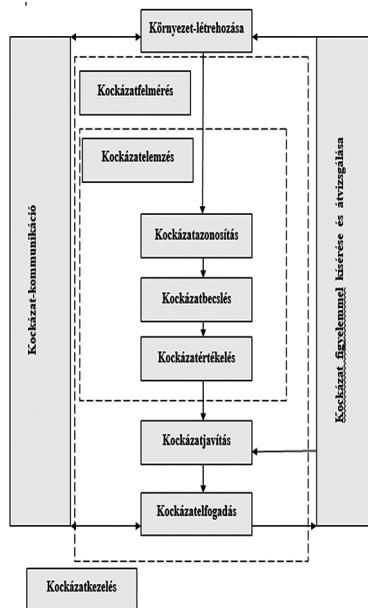
- bevezetés,
- előzetes kockázatfelmérés,
- kockázatelemzés (Analyze): kockázatazonosítás (Identify), kockázatbecslés és kockázatértékelés,

- hatásellenőrzés és -mérés, kockázatkezelés és -javítás, beavatkozás (Control),
- kockázatfogadás, eredmények közzététele (Communicate, Deliberate, and Document).

A kockázatkezelés folyamata mindaddig tart, amíg a kockázatot a lehető legkisebb mértékűre sikerül csökkenteni.

A kockázatmenedzsmentnek nem célja, hogy felderítse az összes lehetséges körülményt és befolyásoló tényezőt, de törekedni kell arra, hogy a lehető legtöbb fenyegetés¹³ típusait (veszélyforrást¹⁴) számba vegyünk.

2. ábra. ISO 31000 Risk management – guidelines alapján (saját szerkesztés)



A kockázatkezelésnél figyelembe kell venni a szervezetre ható input folyamatokat, az erőforrásokat, amelyeket a rendszer a folyamatok és műveletek végrehajtása során felhasznál, valamint a környezeti hatásokat és kapcsolatokat, tehát a rendszeren kívülről érkező hatásokat, amelyek meghatározzák

- a szervezetek fenyegetéseit és a potenciális veszélyek jellegét és szintjét, – a káros hatások bekövetkezésének valószínűségét,

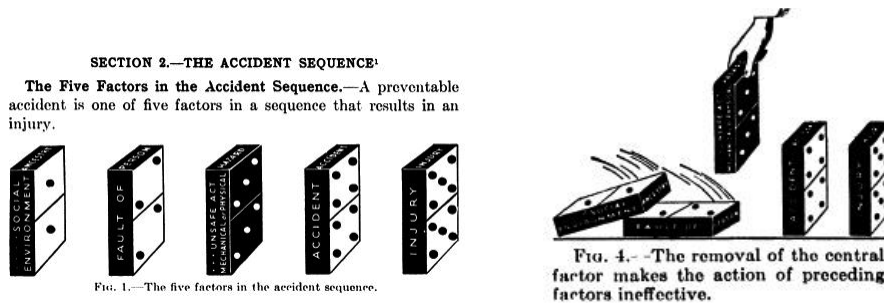
- a megszakítások (rendellenes rendszerleállás, szolgáltatáskimaradás) szintjét és a helyreállítás költségét,
- a kockázatok kezelését szolgáló ellenőrzések hatékonyságát.

A szükséges intézkedések típusa befolyásolja a kockázatkezelést, ami lehet megelőző (preventív), észlelő (detektív) és javító (korrektív). Az intézkedések típusának megválasztása esetfüggő, ezért az alkalmazható általános érvényű elveket, módszereket érdemes figyelembe venni. Az incidensek sokféle méretben, megnyilvánulási formában fordulnak elő, az általuk okozott veszteség is különböző, így lehet ipari, gazdasági és személyi vonatkozású kár. Az incidenst kiváltó okokat, az incidens folyamatát és lehetséges kimenetét többféle módszerrel lehet vizsgálni. Mivel a legtöbb incidens összetett, ezért elemzésükhöz is többféle módszer szükséges. Minden egyes elemzés során számba kell venni az elemzéshez alkalmazott eszközöket, módszereket és egy-egy új modell bevezetése nem azt jelenti, hogy a régié elavultak, inkább csak azok továbbfejlesztett változata alkalmasabb a feladat elvégzéséhez. Az alkalmazott módszerek és az általuk megfogalmazott elvek kiegészítik egymást. Minden kockázatkezelés egyedi, amely egyedi vagyontárral, sebezhetőségekkel, esetleges incidensekkel és intézkedésekkel rendelkezik, ezért a kockázatkezelés során a meglévő kockázatkezelési módszerek megfelelő kiválasztásával tökéletesíthető a kockázatkezelési intézkedési terv, és csökkenthető az esetlegesen bekövetkező incidensek száma és hatása, valamint a kockázat, hiszen az esemény bekövetkezésének valószínűsége is csökken, így növelhető a kockázatkezelés hatékonysága. Az alábbiakban felsorakoztatott módszerek segítenek abban, hogy a kockázatot a lehető legpontosabban határozhassuk meg.

Kockázatkezelésnél az alábbi modelleket, módszereket lehet figyelembe venni:

- *Biztonsági modellek*: amelyek a biztonságra (elsősorban ipari, informatikai, nemzet) vonatkozó tulajdonságokkal és feltételezésekkel kapcsolatos hipotéziseket rögzítik, amelyek igazolásával hozzájárulnak a rendszer biztonságának növeléséhez.
 - *Baleseti modellek*: baleseti hipotézisek összessége, amely az előfordulásának módját és az elveket rögzítik, ilyen például a Seveso-irányelv¹⁵ vagy az irányelv alapjául szolgáló modellek.
- A baleseti modellek egyik alapja az egyszerű, véletlenszerű, szekvenciális modell, a H. W. Heinrich dominómodellje, amelynek lényege, hogy a balesetek egy meghatározott sorrendben előforduló, mechanikus események sorozatából származnak és a baleset megakadályozható azzal, ha eltávolítjuk a dominó-sorozat egyik meghatározó tényezőjét (3. ábra).

3. ábra. H. W. Heinrich dominómodellje, 1941.¹⁶



A dominóelv a kockázatkezelés során jól alkalmazható, hiszen a megfelelően kiválasztott gyenge pont kezelése és a biztonsági rés befoltozása. A sebezhetőség kijavítása a sebezhetőséget kihasználó incidensek számát csökkenti, mivel a támadás során ezen a helyen rést nem, csak bezárt ajtót találnak. A modell előnye, hogy egyszerű és könnyen átlátható, megérthető és lehetőséget nyújt a lényeges ok-okozati tényezők azonosítására, amelyek hozzájárulnak a baleset vagy incidens bekövetkezéséhez. A dominóelv a viselkedésalapú biztonsági programok együttes alkalmazásával lehetőséget nyújt a gyenge pont előtérbe helyezéséhez, az esetleges emberi hibák, hibás emberi teljesítés észleléséhez.

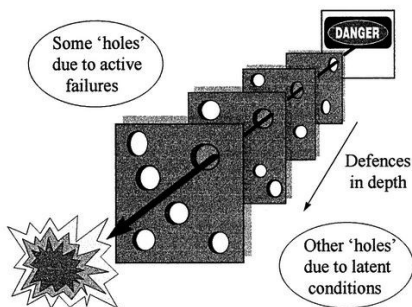
A dominóelv alapján az alábbi kockázati faktorok állapíthatók meg:

- kiváltó ok (eredő tulajdonság) és társadalmi környezet (hanyagság, makacsság, kapzsiság; nem kívánatos vagy zavaró tényező),
- személy hibája (veszély okozása, bármely emberi tulajdonságból vagy viselkedésből eredően),
- nem biztonságos cselekmény és / vagy mechanikai vagy fizikai veszély,
- baleset (főként emberi hibából eredő esemény, ami sérülést okoz),
- sérülés (sértés vagy sérülés, ami a baleset következménye).

A H. W. Heinrich másik baleseti modellje a „biztonsági piramis” vagy „baleseti háromszög”, amelynek teóriáját tovább gondolva megállapítható, hogy a kisebb események bekövetkezési gyakoriságának csökkentése, csökkenti a súlyos balesetek bekövetkezésének valószínűségét.

A következő modell, James Reason svájci sajtmodellje szerint, ha feltételezzük, hogy az incidenst az aktív hibák (hibás biztonsági magatartás) és látens, környezeti tényezők kombinációja okozza, a korlátok megerősítésével megelőzhető a baleset. A biztonság menedzselése ebben az esetben is megköveteli a teljesítmény-mutatókat.

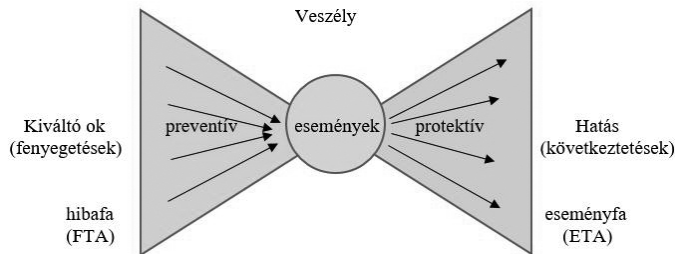
4. ábra. svájci sajtmmodell¹⁷



A módszer (4. ábra) lényege, hogy az incidens csak akkor következhet be, ha minden „sajt” rétegen, azaz minden védelmi rendszeren átcúszik a nemkívánt támadási kísérlet. Számos iparág, mint például a közlekedés (légitársaságok) vagy az egészségügy használja ezt a modellt. A védekezési lehetőségeket a különböző méretű és elhelyezkedésű lyukakkal ellátott „sajtszeletek” jelentik. Lehetőleg azonos méretű és elhelyezkedésű lyukakkal ellátott „sajtszeletek” ne kövessék egymást. Amennyiben ez mégis előfordul, az adott védelmi réteg nem éri el a szükséges hatást és a támadás átjut rajta. Szükséges a gyenge pontok minimalizálása és a biztonsági rések befoltozása.

A következő modell, a Bow-tie, ami egy vizuális csokornyakkendő módszer a veszély hatásainak, a kockázatok, a következményeknek megértésére. A módszer lényege, hogy a bal oldalon fel kell sorolni az összes olyan lehetséges okot vagy fenyegetést, amely az esemény bekövetkezéséhez vezethet. A jobb oldalon pedig az összes olyan következményt fel kell tüntetni, amely az esetlegesen bekövetkezendő eseményhez kapcsolható. A kiváltó okok és a lehetséges következmények egymással összekapcsolhatók és az ok-okozat között felállítható a logikai folyamatábra. A csokornyakkendő módszer leegyszerűsített ábrázolása az 5. ábrán található.

5. ábra. Bow-tie modell alapján (saját szerkesztés)



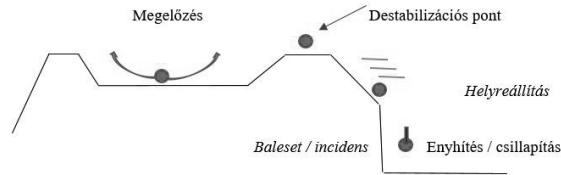
A Bow-tie modell továbbfejlesztett változata a HSE-módszer, ami egy strukturált ellenőrzési modell a kockázatok kezelésére, aminek egy részletét a következő ábra mutatja.

6. ábra. Health, Safety and Environment (HSE) Management System, TÜV



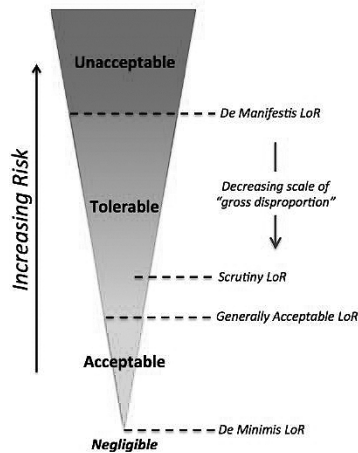
A Loss of control accident model (7. ábra), az irányítás elvesztése – a nem lineáris baleseti modell (French BEA) feltételezése szerint, az incidens egy nem várt esemény és a normál teljesítményváltozás kombinációjából adódik. Az incidens megelőzése a folyamatok megértésében és a monitoringolásában rejlik. Az elvárt védelmi szint megköveteli a bekövetkező események detektálhatóságát és a megfelelő reagálás képességét. A modellt repülőgép balesetek analizálásához és megelőzéséhez használják.

7. ábra. Loss of control accident model (French BEA) alapján (saját szerkesztés)



- A mentális modellek alkalmazása befolyásolhatja az adott rendszer tervezését és működését, az operatív döntéseket és a munkavállalók viselkedését. A humán modell szerint a biztonság jelentősen javul, ha eltávolítjuk a „megromlott almát”, tehát az olyan alkalmazottakat, akik emberi hibákat, úgynevezett „human-error”-t generálnak, mivel sorozatos gondatlan magatartásuk áterjed a többi munkavállalóra is. Bármely rendszerben az emberi viselkedést a környezeti tényezők, korlátok befolyásolják, mint például a nyereséges és veszteséges típusú folyamatok, biztonsági műveletek, munkaterhelés. A „human-error” menedzsment rendszerszemlélet híveinek egyik jelentős álláspontja, hogy egy olyan átfogó irányítási programra töreksszenek, amely több különböző célra irányul: a személyre, a csapatra, a feladatra, a munkahelyre és az intézményre.¹⁸

8. ábra. ALARP-alapelve²⁰



– Kockázatkezelést segítő módszerek:

- ALARP (As Low As Reasonably Practicable) kockázatkezelési keretrendszer, amely a kockázatcsökkentés meghatározására szolgáló alapelveket rögzíti.

Az Európai Unió számos döntési mechanizmusához,¹⁹ vagy repülésbiztonság rendszereihez használják az ALARP-keretrendszert, amelynek kritériumai.

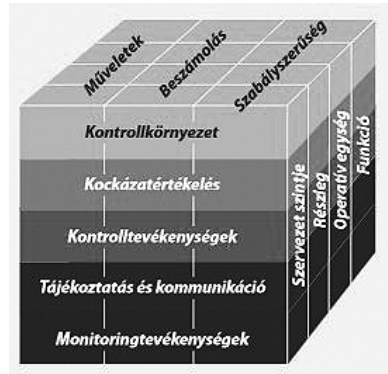
- az elfogadhatatlan tartomány, amely a tevékenységekhez kapcsolódó előnyöktől független.
- az elfogadható ALARP-tartomány, a kockázat vállalható, de van következménye. A javasolt kockázatkezelést akkor kell végrehajtani, ha a veszteség nincs aránytalanul nagyobb arányban a végrehajtás által elért előnnyel szemben. A kockázati szintet a lehető legnagyobb mértékben csökkenteni kell, figyelembe véve költségének mértékét.
- általánosságban elfogadható tartomány, ebben az esetben nincs szükség további kockázatcsökkentési módszer alkalmazására.

- COSO-keretrendszer, amelyet 1992-ben vezettek be az Amerikai Egyesült Államokban, legfőképp nagyvállalatok használták, azóta széles körben elterjedt és számos közszolgálati intézmény is alkalmazza az egységes belső ellenőrzési keretrendszert.

A COSO ERM kockázatviselési szintje azt mutatja meg, hogy a vezetőség mekkora kockázatot tart elfogadhatónak, milyen mértékű lehet a kockázati tolerancia elsődlegesen a gazdasági tevékenységek figyelembe vételével. Az események lehetnek negatív hatásúak, amelyek kockázatot jelentenek és megakadályozhatják az értékteremtést, illetve ronthatják a meglévő értéket, valamint lehetnek pozitív hatásúak, amelyek ellensúlyozhatják a negatív hatásokat vagy lehetőséget jelenthetnek a szervezet számára. (9. és 10. ábra)

Pozitív és negatív hatású események kombinációja is előfordulhat. A keretrendszer pozitívan befolyásolja a célok elérését, támogatja az értékteremtést és az értékmegőrzést.²¹ Rávilágít arra, hogy a lehetséges események milyen mértékben befolyásolhatják a vállalkozások és intézmények célkitűzéseit. Két szempont alapján értékeli a kockázatokat: valószínűség és hatás. A keretrendszer kvalitatív és kvantitatív kockázatértékelési módszerek kombinációját alkalmazza.

9. ábra. COSO kocka²²



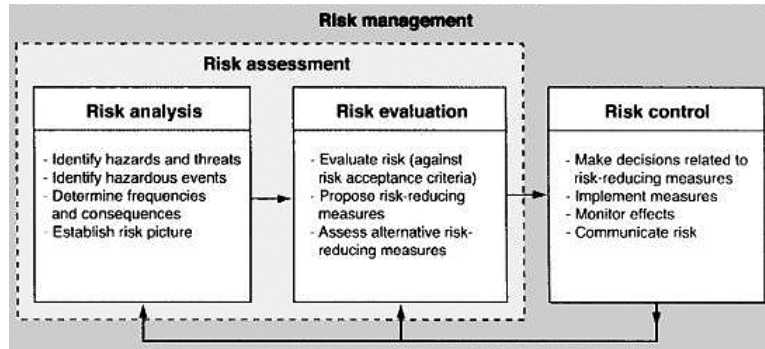
11. ábra. COSO kockázattértékelés

I M P A C T	High	<u>Medium Risk</u> • Loss of phones • Loss of computers	<u>High Risk</u> • Credit risk • Customer has a long wait • Customer can't get through • Customer can't get answers
	Low	<u>Low Risk</u> • Fraud • Lost transactions • Employee morale	<u>Medium Risk</u> • Entry errors • Equipment obsolescence • Repeat calls for same problem
		Low	High
		PROBABILITY	

– Kockázattértékelés matematikai számítási módszerei

A kockázatelemzés a rendelkezésre álló információk szisztematikus felhasználása a veszélyek azonosítására és az egyénekre, a tulajdonságokra, és a környezetre gyakorolt kockázat becsléséhez. (IEC 60300-3-9, 1995). A kockázattértékelés a kockázatelemzés és a kockázatszámítás vagy becslés átfogó folyamata. (11. ábra)

12. ábra. Kockázatértékelés, IEC 60300-3-9, (1995.) alapján



A kockázatértékelés alapelvei szerint a kockázatszámítás leegyszerűsített számítási módja:

$$R=C \times F$$

ahol az R (Risk) kockázat egy adott incidens vagy esemény bekövetkezésének gyakorisága vagy valószínűsége (F), valamint a következmény súlyosságának (C) a szorzata²³.

Komplex rendszerek esetén:

$$r = \sum_{t \in T} (p_t \times d_t)$$

Ahol

r – a rendszer biztonsági kockázata,

T – a releváns fenyegetések halmaza,

p_t – egy adott fenyegetés bekövetkezésének valószínűsége (gyakorisága)

d_t – egy adott fenyegetés bekövetkezéséből származó kár.²⁴

A kockázatszámítási módszerek lehetőséget biztosítanak a kockázat gyors értékelésére, ezáltal lehetőség van egy adott rendszer információbiztonsági kockázatainak felmérésére és meghatározására, valamint szükséges intézkedések megvalósításával az elvárható és elfogadható kockázati szint beállítására.

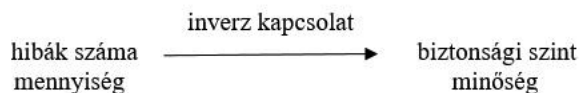
A kockázati mátrix (1. táblázat), olyan kétdimenziós számhalmaz, melynek függőleges tengelyén a veszélyeztető hatás következménye, vízszintes tengelyén a veszélyeztető hatás bekövetkezési valószínűsége (gyakorisága) található, és amelynek eredményeként megállapítható, hogy egy adott veszélyeztető hatás mekkora kockázatot jelent az adott rendszerre, objektumra.²⁵

1. táblázat. Kockázati mátrix táblázatos formában²⁶

Hatás	Bekövetkezési gyakoriság			
	Ritka	Nem gyakori	Gyakori	Nagyon gyakori
Nagyon súlyos	II. osztály	II. osztály	I. osztály	I. osztály
Súlyos	III. osztály	II. osztály	II. osztály	I. osztály
Nem súlyos	III. osztály	III. osztály	II. osztály	II. osztály
Alacsony	III. osztály	III. osztály	III. osztály	III. osztály

Számos matematikai számítási módszer szolgál a kockázatelemzésre, a maradványkockázat, valamint a hiba számosságának, úgymint a hibás folyamatok számosságának megállapítására, amely a rendszer biztonsági szintjének kvantitatív értékelését eredményezi. Ezekhez a módszerekhez összetettebb matematikai számítási folyamatok szükségesek, úgymint a HAZOP – Hazard Operability Analysis, a HACCP – Hazard Analysis and Critical Control Points, az FMEA – Failure Mode Effects Analysis, vagy az FMCEA – Failure Mode, Effects and Criticality Analysis. Általánosságban megállapítható, hogy a hibák száma fordítottan arányos a biztonsági szint értékével (13. ábra).

13. ábra hibák száma és a biztonsági szint kapcsolata (saját szerkesztés)



Összegzés

A személyes adatok gyűjtése során a legújabb, érintés nélküli biometrikus adatgyűjtésnél ugyan úgy figyelembe kell venni az adatvédelmi és az információbiztonsági előírásokat egyaránt. Tekintettel arra, hogy a biometrikus adatok az adatvédelmi szabályok és alapfogalmak szerint különleges adatnak minősülnek, fokozottabb védelmet kívánnak. A biometrikus adatokat gyűjtő vállalatok esetenként elhanyagolják az információbiztonsági kockázatok felmérését és kezelését, ezáltal az intézkedési tervekből és a fejlesztésekből kimaradhatnak azok a tényezők, legfőképp a jelentős biztonsági rések kezelésére irányuló intézkedések, amelyek az incidensek elkerüléséhez, megelőzéséhez vezethetnek. Információbiztonsági szempontból te-

hát elengedhetetlen azon információs rendszerek kockázatkezelése, amelyek személyes adatokat gyűjtenek, tárolnak és kezelnek. A kockázatok felmérésére, analizálására és kockázatértékelésre az elmúlt évtizedek során többféle, sokszor egymásra épülő, az információs technológiával szinte együtt fejlődő, számos modellt és módszert találhatunk, amely az adott információs rendszer esetében releváns lehet. A megfelelő, de speciális módszer kiválasztása mindig rendszerfüggő vagy iparágfüggő, hiszen más módszereket alkalmaz a repülésbiztonság, az egészségügy vagy a közigazgatás. A tanulmányban azon módszereket emeltem ki, amelyek alapelveit tekintve számos területen lehet használni, ezáltal alkalmazásuk is népszerűbb. Azon alapelvek és módszerek, amelyek legjobban szolgálják az alkalmazásuk célját, elősegítik a hatékony információbiztonsági kockázatkezelést, költséget takarítanak meg az adott szervezet számára, mivel az incidensek bekövetkezésének valószínűségét is csökkentik, közvetett módon pedig az információs rendszer biztonsági szintét növelik. Amennyiben a vállalatok és intézmények támogatják az információbiztonsági kockázatkezelési intézkedések hatékony végrehajtását, az érintett a biometrikus adatát is nagyobb biztonságban tudhatja.

Jegyzetek

¹ Federal Bureau of Investigation

² Czuni László (2015): *Biometria a számítógépes személyazonosításban – vizuális módszerek*. Egyetemi jegyzet. Pannon Egyetem, Műszaki Informatikai Kar, Képfeldolgozás Kutatólaboratórium.

³ How COVID Is Revolutionising Touchless Airport Technology, <https://simpleflying.com/touchless-airport-technology-covid/>. Letöltés: 2021. május 12.

⁴ Lawrence, Steve, et al. (1997): „*Face recognition: A convolutional neural-network approach*.” IEEE transactions on neural networks 8.1 Pp. 98–113.

⁵ Next Generation Identification (NGI)

⁶ FBI, Advanced Fingerprint Identification Technology (AFIT), <https://www.fbi.gov/services/cjis/fingerprints-and-other-biometrics/ngi>. Letöltés: 2021. május 5.

⁷ FBI, <https://www.fbi.gov/file-repository/ngi-monthly-fact-sheet/view>. Letöltés: 2021. május 5.

⁸ Nemzeti Adatvédelmi és Információszabadság Hatóság (NAIH), Az Adatvédelmi Hatóság vizsgálja a Facebook hangfelvétel leírási gyakorlatát (közlemény), <https://naih.hu/files/2019-08-14-kozlemen-y-FB-Messenger.pdf>. Letöltés: 2021. május 12.

⁹ Electronic Privacy Information Center, <https://epic.org/>

¹⁰ Transportation Security Administration (TSA), <https://www.epic.org/apa/comments/EPIC-TSA-Pre-Check-Expansion-Comments.pdf>. Letöltés: 2021. május 5.

¹¹ Európai Bizottság, https://ec.europa.eu/info/law/law-topic/data-protection/reform/rules-business-and-organisations/principles-gdpr/what-data-can-we-process-and-under-which-conditions_hu, letöltés: 2021. május 12.

12 2011. évi CXII. törvény az információk önrendelkezési jogról és az információszabadságról.

¹³ Fenyegetés (Threat): olyan lehetséges művelet vagy esemény, amely sértheti az elektronikus információs rendszer vagy az elektronikus információs rendszer elemei védetségét, biztonságát, továbbá olyan mulasztásalapú cselekmény, amely sértheti

az elektronikus információs rendszer védetségét, biztonságát.

¹⁴ Veszély: olyan személy vagy körülmények együttese, mely biztonsági esemény okozója lehet.

¹⁵ Seveso-irányelv I-III. Veszélyes anyagokkal kapcsolatos súlyos balesetek, <https://eur-lex.europa.eu/legal-content/HU/TXT/HTML/?uri=LEGISSUM:l21215&from=HU>, 1976-ban történt seveso-i katasztrófa következtében a balesetek megelőzésére és kezelésére vonatkozó jogszabály.

¹⁶ H.W. Heinrich (1941): *Industrial Accident Prevention, A Scientific Approach*. Second edition. New York–London: McGraw-Hill Book Company.

¹⁷ James Reason (1997): *Managing the Risk of Organizational Accident*. Routledge.

¹⁸ James Reason (2000): *Human error: models and management*.

¹⁹ Health and Safety Executive, <https://www.hse.gov.uk/managing/theory/index.htm>, letöltés: 2021. május 9.

²⁰ Reece, A.–Clothier, B. P.–Williams, N.–Fulton, L.–XunGuo Lin (2013): *ALARP and the Risk Management of Civil Unmanned Aircraft Systems*.

²¹ COSO Enterprise Risk Management – Integrated Framework, <https://www.coso.org/pages/erm-integratedframework.aspx>. Letöltés: 2021. május 9.

²² Az Európai Bizottság szervezetirányítási rendszere: helyes gyakorlatok?, www.eca.europa.eu/lists/ecadocuments/sr16_27/sr_governance_hu.pdf, 2016. Luxembourg. Letöltés: 2021. május 9.

²³ Marvin Rausand, Risk Assessment Theory, Methods, and Applications, New Jersey, 2011.,

„Frequency analysis. This step will usually involve a deductive analysis to identify the causes of each hazardous event and to estimate the frequency of the hazardous event based on experience data and/or expert judgments.”

„Consequence analysis. Here, an inductive analysis is carried out to identify all potential sequences of events that can emerge from the hazardous event. The objective of the inductive analysis is usually to identify all potential end consequences and also their probability of occurrence.”

²⁴ Muha Lajos (2009): *Az informatikai biztonsági kockázatok elemzése*. Robothadviselés.

²⁵ NASA, Risk Management Reporting, 2009.

²⁶ 234/2011. (XI. 10.) Korm. rendelet, a katasztrófavédelemről és a hozzá kapcsolódó egyes törvények módosításáról szóló 2011. évi CXXVIII. törvény végrehajtásáról, 2021. május 9-i állapot.

Köszönetnyilvánítás

A közlemény megjelenését az EFOP-3.6.1-16-2016- 00003 „K+F+I folyamatok hosszú távú megerősítése a Dunaújvárosi Egyetemen” című projekt támogatta.

Galéria

Duma Bálint fotói



