

A Debreceni Fejlesztői Hálózat

BÁTFAI NORBERT

Debreceni Egyetem, Informatikai Kar, Információ Technológia Tanszék
batfai.norbert@inf.unideb.hu

MOLNÁR PÉTER

Debreceni Egyetem, Egyetemi és Nemzeti Könyvtár
pmolnar@lib.unideb.hu

MOLNÁRNÉ NAGY MÁRIA

Debreceni Egyetem, Informatikai Kar, Debreceni Informatikai Kutató-Fejlesztő Központ Non-profit Kft.
nagymaria1@gmail.com

RÁBAI BÁLINT, SZITHA KRISTÓF, KOVÁCS ZSOLT, HUDÁK LÁSZLÓ, RÁK JÁNOS

Debreceni Egyetem, Informatikai Kar
rabai.balintos@vipmail.hu, {szithak, kovacs.zsolt.85, hudak.laszlo.42, wilson90kbf}@gmail.com

Kulcsszavak: DDN, közösségépítés, mobil játék, Java, Kolmogorov-bonyolultság, hasonlósági metrika, viselkedés API, nyílt forráskód

Ebben a cikkben egy egyetemi szoftverfejlesztői közösség, a Debreceni Fejlesztői Hálózat, röviden DDN, kialakítását, működésének megszervezését mutatjuk be. Megismerjük a DDN tipikus használati eseteit, bemutatjuk a formálódó közösség kezdeti, illetve jelenleg futó fejlesztési projektjeit.

1. Bevezetés

Minden területre igaz, hogy meghatározó dolgok ritkán születnek, mert ezeknek a dolgoknak ez a természetük. De nagyon érdekes, hogy ehhez viszonylag mégis milyen gyakran tesznek nagy dolgokat hallgatók az informatikában. Például 1987-ben megjelenik a Minix [1], a helsinki egyetemen operációs rendszerekből, BSc hallgatóként már ezt tanulja Linus Benedict Torvalds, majd 91-ben [2] megszületik a Linux. Bill Joy MSc hallgatóként részt vállalt a BSD disztribúció összeállításában, később ő a Sun egyik alapítója. Számos nevezetes, hasonló példát olvashatunk [3]-ben. Ez az élmény az, ami miatt különösen fontosnak tartjuk a hallgatókkal való szoros kapcsolatot, közös munkát.

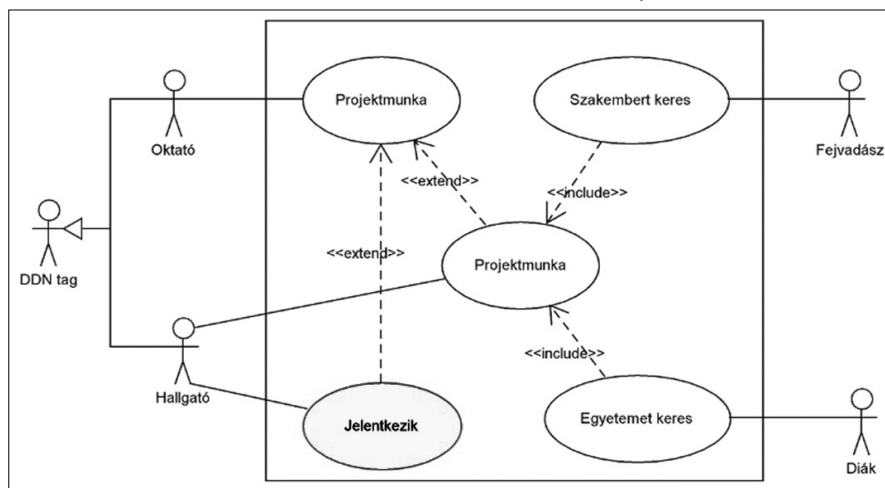
Mi tehet informatikai nagyhatalommá egy országot? Egyáltalán mi kiket tartunk nagyhatalomnak? Finnországot igen, a Nokia és a Linux okán. Hollandiát igen, a Minix és mondjuk a hasonlósági metrika kapcsán. Ám az USA-beli neves egyetemek adhatnak egy objektívebb definíciót: nagyhatalom generátora az olyan egyetem, ahol az ott keletkező források mellé tett licenc általánosan elterjedté válik. Erre példa a – jól ismert – BSD licenc [4], a Berkeley-i egyetem kifejlődött engedély.

Jelen közleményünk harmadik motiváló tényezője, hogy 2008 decemberében az Év Informatikai Oktatója [5] díjátadó ünnepségen tett

bejelentésnek megfelelően 2010 januárjára (az első szerző disszertációja [6] kapcsán) már elérhetőek az Eurosmobil [5,6] megnyitott játéka [8,9]. De valóban élő informatikai projektek létrehozásához nem elegendő csupán a források megnyitása, szükség van egy fejlesztői közösségre, aminek tagjai gondozzák ezeket a megnyitott forrásokat. Ez az életre hívott közösség a Debreceni Fejlesztői Hálózat [11], vagy röviden a DDN. Ezt az egyetemi egységet szeretnénk bemutatni ebben a dolgozatban.

Jelen közleményünk felépítése a következő. A második szakaszban röviden bemutatjuk a DDN kialakítását, tervezett használati eseteit, majd a harmadikban a jelen pillanatban futó témákat villantjuk fel, végül a cikket az összefoglalás zárja le.

1. ábra
A DDN tipikus használati esetei



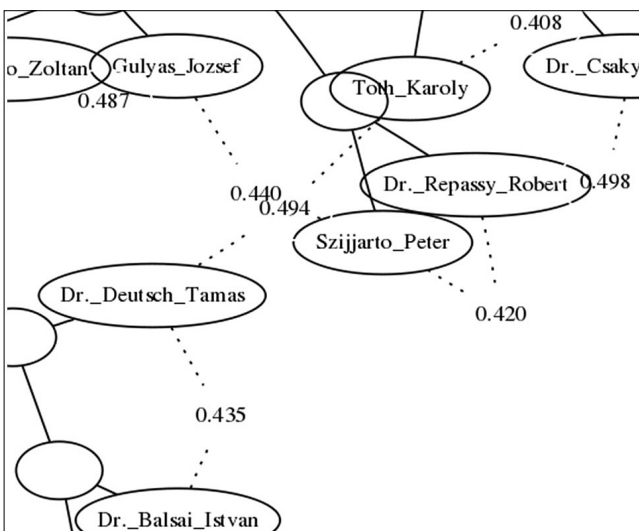
2. A DDN kialakítása és kezdeti projektjei

A felsőoktatásban nem ritka, hogy neves intézmények kutatás-fejlesztési csoportokat hoznak létre, mely csoportok természetes módon vállalják fel az egyetemi háttér biztosította hallgatói bázis megszervezését is. Hazai példát említve: a BME Automatizálási és Alkalmazott Informatikai Tanszékének AMORG (<http://amorg.aut.bme.hu>) csoportja egy olyan csoport, amit példaképnek tekinthetünk.

A DDN szervezésének előkészítése a 2008/2009 tanév őszi félévének elején megkezdődött, az alapító első szerző célja egy széles hallgatói tömegebázison nyugvó, egyetemi szoftverfejlesztői közösség kialakulását inkubálni képes szervezet kialakítása volt. A szerveződések céljait az 1. ábra használatieset-diagramja mutatja: a hallgató érdekes projektekből szeretne részt venni és referenciaként felhasználható tapasztalatokat gyűjteni; a középiskolás diák jó egyetemet keres; a fejlődés cégek képviselői megfelelő, esetleg végzős hallgatókat keresnek; végül az oktatók olyan hallgatókat, akik szeretnének szoftverfejlesztésekhez kapcsolódni, s így a – remélhetőleg rezonálni képes – kör bezárult.

Jelen pillanatban a hallgató akkor csatlakozhat a DDN-hez, ha két egyszerű feladatot elkészít. A kezdetektől mostanáig ez a feladat egy Java ME és egy Google Android projekt (a [18] könyv „NehogyMár7” és „NehogyMár3” feladatainak) felélesztése a népszerű NetBeans, illetve a „rivális” Eclipse fejlesztői környezetekben. Ennek teljesítése után még annyi a hallgató feladata, hogy a DDN Évkönyvének megfelelő, saját magát jellemző XML állományt kitölti az adott félévben rá vonatkozó adatokkal, majd az adott XSL fájl segítségével legenerálja az évkönyv DocBook 5.0 XML változatát, amiből végül elkészíti az évkönyv aktuális pdf formáját [12]. Ezzel a „beléptető” rendszerrel azt biztosítjuk, hogy a sikerrel jelentkezett hallgató képes (legalábbis az egyszerű esetekben) a fejlesztői környezetek és egy verziókezelő (Subversion, <http://subversion.tigris.org>) rendszer alapvető használatára.

2. ábra Egy parlamenti gráf részgráfja



2009-ben kilenc, a DDN Évkönyvben részletesen bemutatott belső projekt merült fel. Legjobban a Magyar Parlament képviselői gondolatainak filogenetikája című keltette fel a hallgatók érdeklődését, de önálló publikációig egyelőre nem érett meg ez a kezdeti projekt. Itt a Kolmogorov-bonyolultságra alapozott hasonlósági metrikát [13] implementáló CompLearn [14] csomagot használtuk fel arra, hogy a Magyar Parlament képviselőit, a <http://www.parlament.hu/> lapról „kibányászott” adataik alapján jellemezzük.

A CompLearn csomaggal készített gráf egy részgráfját mutatja a 2. ábra. A gráf kapcsolatainak helyes értelmezése az lehet, hogy mely képviselőket foglalkoztatnak hasonló gondolatok, ettől természetesen a megoldásaik lehetnek még teljesen különbözőek. Az említett CompLearn csomagot számos összehasonlítási feladatnál használják, például számítógépes fájlípusok, mitokondriális DNS szekvenciák vagy emberi nyelvek törzsfájának meghatározására [15]. Mi magunk, a politikai gráfokon túl – amelynek ötletét egyébként magának a csomagnak a dokumentációja is sugallja –, nagy, nyílt forráskódú projektek forrásainak klaszterezésére próbáltuk ki még. A [10]-ből származó 3. ábra mutatja, hogy a csomag képes azonos klaszterbe sorolni ugyanazon szoftver különböző verzióit.

A CompLearn csomag felhasználása doktori témák keretében jelenleg is aktuális, ezt mutatja be a következő alszakasz.

2.1. Hasonlósági mérték használata a szoftver reengineering területén

A cél a Java forráskódból megtalálni a szoftver azon részeit, amelyek nem megfelelően tervezettek, a források elemzését a CompLearn csomag [13] segítségével elvégezve a kritikus pontok azonosítása. Ezzel a módszerrel jellemzően felderíthető problémák a kódmásolás, a hasonló felépítésű osztályok külön-külön – nem osztályhierarchiában – történő implementálása. A kezdeti eredményekből előadás [26] született és cikk is készült a témában.

Időközben az Ars Poetica Informaticae (API) [11] portál alportáljaként elkészült a DDN WebSynergy, <https://portal.dev.java.net/public/Downloads.html> – jelenlegi nevén GlassFish Web Space Server – portálmegoldásra alapozott website-ja [16]. Ennek a portálmegoldásnak a használata azért fontos a számunkra, mert lehetőséget ad hallgatóinknak Java EE tapasztalatok szerzésére: saját portletjeiket, azaz a portálra „drag and drop” módszerrel is ráhúzható szoftverkomponenseket tudnak telepíteni a futó API portálra, akár közvetlenül a NetBeans fejlesztői környezetből is.

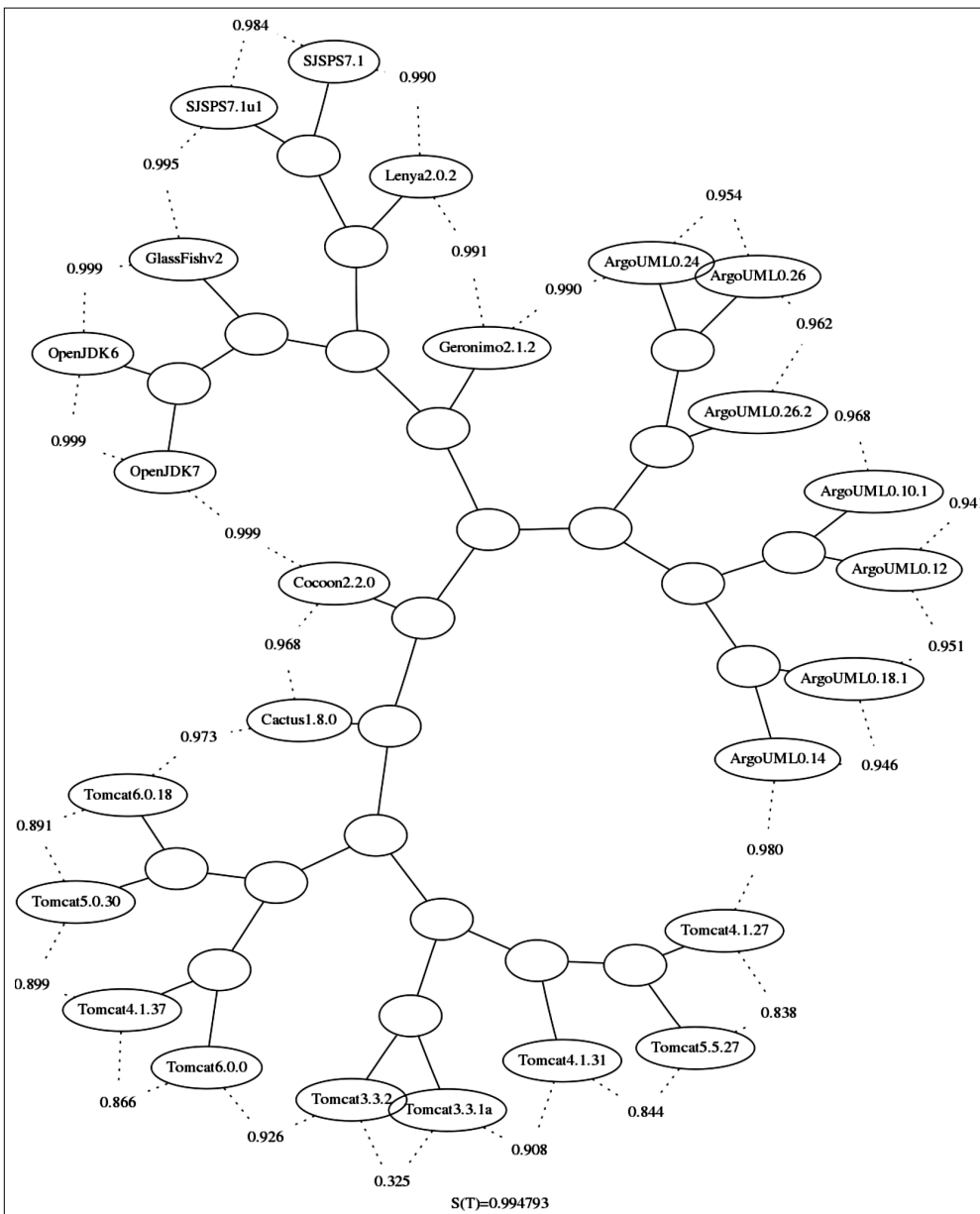
A szóban forgó alportálon megtalálható minden eddig említett DDN specifikus dokumentumunk: a DDN Évkönyv, a felvételi feladatok részletes leírása, a fejlesztett projektekhez kapcsolódó anyagok, dokumentációk. Például a Magyar Parlament képviselői gondolatainak filogenetikája könyvtárban nem csupán a legenerált gráfok, hanem ezekből készített posztterek is elérhetőek.

3. A DDN kutatás-fejlesztési irányai

Mindhárom Java platformon: a Java ME, Java SE és a Java EE környezetekben, azaz a mobiltelefonok, asztali gépek és vállalati szerverek világában is otthonosan mozgunk. Ennek megfelelően van gyakorlatunk MIDlet-ek (objektumok, amik a mobiltelefon virtuális gépébe töltődnek be), Applet-ek (objektumok, amik a böngésző virtuális gépébe töltődnek be), Servlet-ek (objektumok, amik a webszerver virtuális gépébe töltődnek be), bean-ek, portlet-ek, Swing-ek vagy teljes képernyős Java alkalmazások írásában. Érdemes megemlíteni, hogy 2006 óta nincs Java csapda, azaz nem fenyegeti a nyílt forráskódban fejlesztőket egy zárt környezet árnya, mert a Sun mindhárom platformot megnyitotta. (A GNU GPL licenc hatálya alatt, ugyanez a licenc védi például a Linux kernelt is.)

3. ábra

Nyílt forrású Java projektek forrásainak összehasonlítása a CompLearn csomaggal



A Java platformon belül intenzíven kezdünk foglalkozni az NFC (Near Field Communication) technológiával, azaz – a Java világából – a JSR 257 API-val. Ebben a világban a JSR (Java Specification Request, <http://jcp.org/en/jsr>) dokumentumok formájában történik a szabványosítás. Például, ha Bluetooth alkalmazást akarunk fejleszteni a telefonunkra, akkor nem elég, ha a készülékünk Bluetooth készülék, a Java fejlesztéshez az is kell, hogy a telefon Javából tudja azt, azaz implementálja a JSR 82 (a Java APIs for Bluetooth) szabvány interfészt.

Az elsős mérnök informatikus hallgatók gyakorlati programozási szemléletét LEGO NXT robotok Java programozásával szélesítjük. Itt tipikusan a processzortéglán az operációs rendszer Java virtuális gépre cserélésével (tehát nem az eredeti NI-os LabView adatfolyam nyelvvel), a leJOS: <http://lejos.sourceforge.net/> Java rendszeren belül rendelkezésre álló viselkedés [17] API-t használjuk. Ez tökéletes egy rovarszerű intelligenciát mutatni képes rendszer megvalósítására. A „gyakorlati szemléletet” fogalom jelentését akkor érezhetjük át, ha elképzeljük a különbséget az első 100 prímszámot kiíró program és a robot versenyautót a pályán sikerrel végigvinni képes program és a programhoz vezető fejlesztés között.

De meg szoktuk jegyezni, hogy a LEGO NXT azért nem robotika! Legalábbis számunkra csak az iménti értelemben vett szemléletformáló lehetőség. Hiszen amíg a (már piaci ciklusát kifutotta) korábbi RIS (Robotics Invention System) 2.0 csomag 12 éves kormegjelöléssel került forgalomba, az új, az NXT csomag magát már 10 éves kortól ajánlja.

Azt még a RIS 2.0 kapcsán érdemes volt megemlíteni, hogy milyen kapcsolatban van az MIT média laborjában készült cikk [19] és a csomagba került „LEGO Constructopedia” nevű kereskedelmi termék. Itt meg kell, hogy jegyezzük, hogy sajnos ehhez hasonló minőségű dokumentációs anyag az NXT csomagban nincs.

Azt még a RIS 2.0 kapcsán érdemes volt megemlíteni, hogy milyen kapcsolatban van az MIT média laborjában készült cikk [19] és a csomagba került „LEGO Constructopedia” nevű kereskedelmi termék. Itt meg kell, hogy jegyezzük, hogy sajnos ehhez hasonló minőségű dokumentációs anyag az NXT csomagban nincs.

3.1. A DDN eszközparkja

Jelen pillanatban több mint 20 darab Nokia, Motorola és Sony Ericsson Java ME mobiltelefonnal rendelkezünk, illetve rendelés alatt van néhány Android, OpenMoko és Maemo készülékünk is. NFC fejlesztéseinket egy „Gemalto Prox Development Kit” hívhatja szolgálni, emellett számos NFC kártyával és matricával rendelkezünk. Asztali gépek tekintetében 2 PC állomással és egy lappal számolhatunk, szerver oldalon van lehetőségünk akár az egyetemi tűzfalon kívülről is elérhetően futtatni GlassFish vagy TomCat szervert. Mindkettő nyílt forrású megoldás, az első a Sun-hoz, a második az Apache-hoz köthető. A mérnök informatikus hallgatók gyakorlati programozási szemléletét két csomag LEGO NXT programozásával tudjuk szélesíteni.

3.2. Aktuális projektek

A következő alszakaszokban néhány mondatban és képben felvillantjuk az aktuálisan éppen futó projektjeinket. Ezekre tipikusan igaz, hogy az első szerző kezdte meg a munkát, az ő ötletén alapul a teljes vagy a helyi téma. (Kivéve a robotversenyes most induló projektet, ahol a versenyautó építését egy hallgatója, Hudák László vetette fel, az első szerző az egységes platform kialakítását javasolta és a témát koordinálja. Illetve a mobil játékokat az első szerző Bátfai Erikával közösen fejlesztette az Eurosmobil-ban, a munka pontos megosztása a [10]-ben van megadva.)

3.2.1. Jávácská One

A „Jávácská One” a bevezető szakaszban említett, megnyitott (horgász, foci és ezoterikus) játékok projektjének kódneve. A játékok forrásai elérhetőek az első szerző <http://www.inf.unideb.hu/~nbatfai/> honlapján és a SourceForge portálon, a <https://sourceforge.net/projects/javacska> lapon. Mindhárom játékból megmutatunk egy tipikus pillanatképet a 4-6. ábrákon. Részletesen olvashatunk róluk a [8-10] munkákban.

3.2.2. A labdarúgás-szimulációs jelölőnyelv

Álmunk itt a labdarúgásban használható döntéstámogató és szimulációs rendszer kifejlesztése. Ötletünket a [21] cikkben mutattuk be, a magyar szakmai közönség elé pedig [22]-ben tártuk. A jelenleg futó szimulációink (<https://sourceforge.net/projects/footballerml>) teljesen a korábbi pont „Focijáték Neked NYFK” című játék foci-szimulátorának modelljén és kódján alapulnak.

3.2.3. Kriptográfia mobiltelefonon

A TARIPAR3, teljes nevén a „Lokális és regionális tartalomipar fejlesztése, innovatív értéknövelt szolgáltatás keretrendszerének kialakítása adatvagyonok hasznosítására” című pályázat keretein belül mi egy Bouncy Castle alapú, hibrid kriptorendszeres Java ME - Java EE demó alkalmazást készítettünk, melyben a különböző funkcionális részek (mint például az RSA vagy ElGamal kulcsgenerálás a telefonon) futási idejét vizsgáltuk meg [20]. S például azt a részeredményt kaptuk, hogy néhány adott készülék esetén a kulcsgenerálás már a telefonkészülékeken is sikerrel elvégezhető. A Bouncy Castle (<http://www.bouncycastle.org/>) egy MIT jellegű licenccel ellátott, azaz nyílt forráskódú, Java ME mobiltelefonra is alkalmazható kriptográfiai csomag.

A teszteleseket 11 telefonon végeztük. A tesztek során nagyon különböző eredmények születtek. A 7. ábrán összefoglaltunk néhány tipikus mérési eredményt. Volt olyan telefon, amelyen a tesztelés órákat vett igénybe (például a Nokia 2600 típusú készüléke) és volt amelyeken csupán néhány percet, mint a Motorola U9 telefonja. Ez a látványos eltérés a teljesítménybeli különbségeknek, az eszköz kategóriájának tulajdonítható.

3.2.4. A Kolmogorov-bonyolultság és a „Szorgos Hód”-probléma

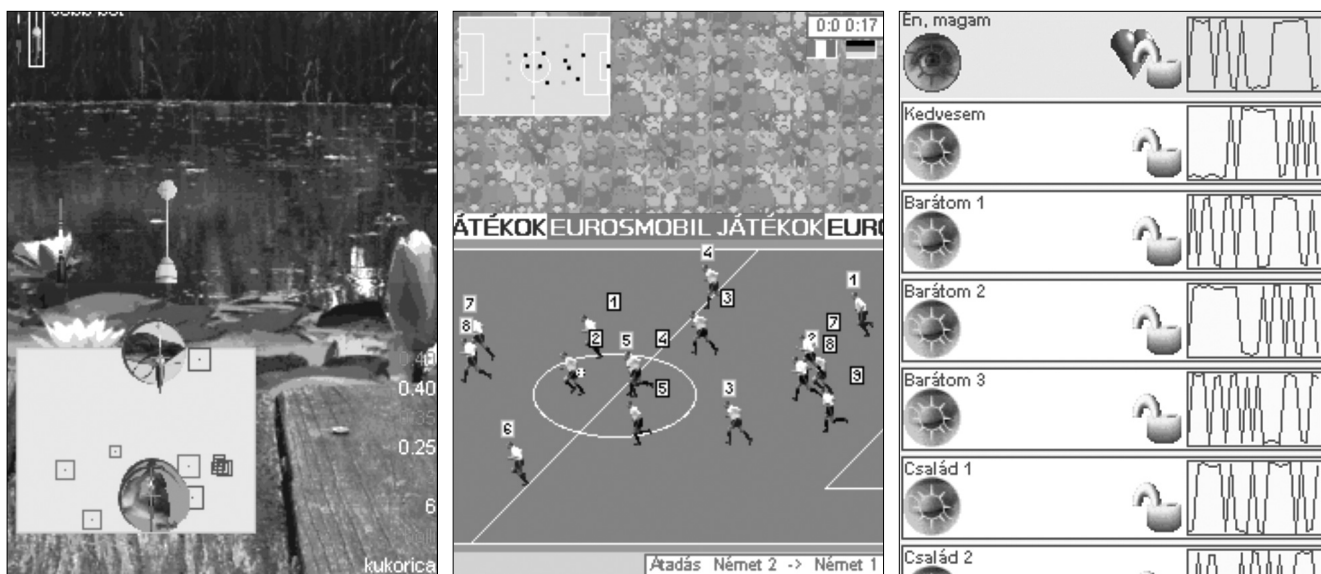
Ez egy inkább elméleti háttérű terület, maga a (Radó Tibor által 40 évvel ezelőtt felvetett) probléma arról szól, hogy maximum hány darab egyes számjegyű tud

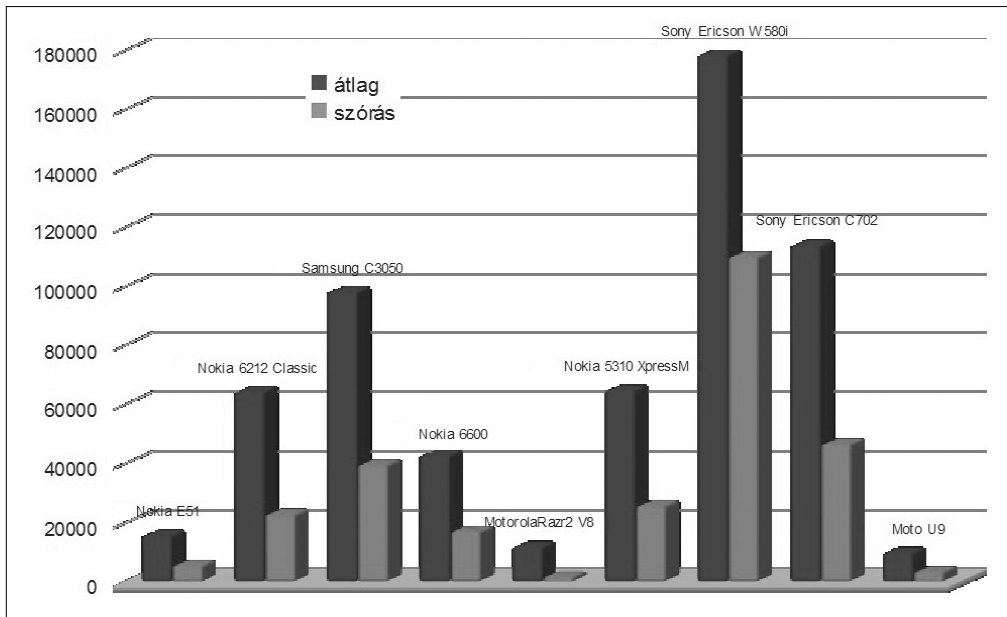
4-6. ábra „Jávácská One”-játékok pillanatképei

„110% Nyári Kapitális NYFK” (horgász)

„Focijáték Neked NYFK” (foci)

„Hetedik Szem NYFK” (ezoterikus)





7. ábra
RSA kulcsgenerálás ideje
milliszekundumban

kiírni egy input nélkül indított bináris Turing-gép. Az első szerzőnek jelenleg is futnak kereső programjai a témában, az eddigi munkát a [23] és a [24] cikkekből ismerhetjük meg.

3.2.5. Felegyenesedett operációs rendszerek

Energiafelhasználás szempontjából nagyon rossz ötlet, hogy a nyílt forráskódú kernelek IDLE folyamatát (vagyis amikor a processzor semmit nem csinál, sőt, emiatt éppen egy energiatakarékos módba kapcsolja magát) átírjuk, hogy valamilyen számítást hajtsunk végre ezekben a periódusokban. Viszont programozóként ez igen izgalmas gondolat, a [25]-ben mutattuk be.

3.2.6. Az országgyűlés kettős spirálja

A 2. szakaszban már bemutatottuk ezt a témát. Most a pártok választási programjait hasonlítottuk össze. Ennek egy korai gráfját mutatja be a 8. ábra, itt a szereplő pártok listája még nem teljes, illetve a szereplők adatai néhány esetben csak töredékesek, hiányosak! A gráf helyes értelmezése ez lehet: kit mi foglalkoztat? De természetesen, ha valakik ugyanazon gondolkodnak, a válaszaik még lehetnek eltérőek.

3.2.7. A Jávácaska Kupa

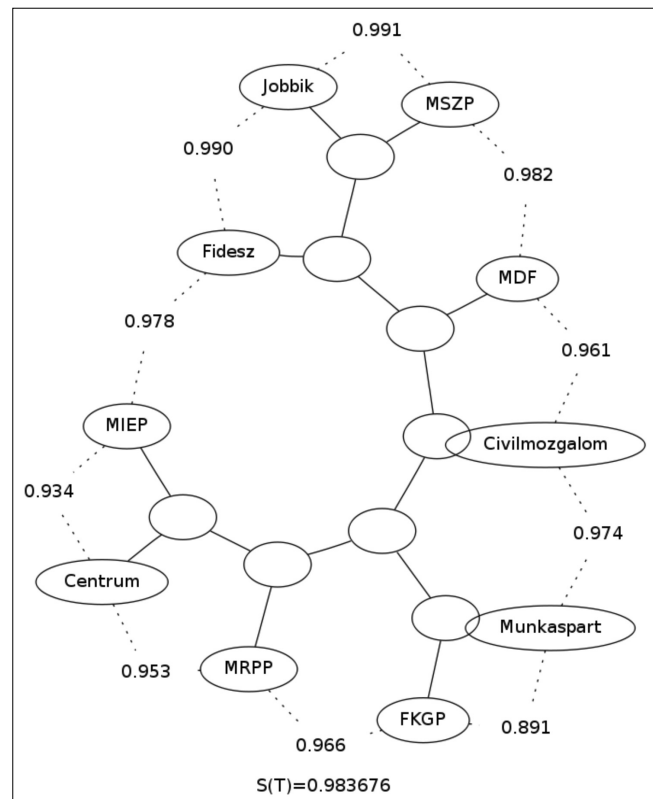
A Jávácaska Kupa keretében egy egységes LEGO robot autóverseny platform felállításán dolgozunk. Célunk egy olyan egységes versenyfelület kialakítása, melybe bárki, aki rendelkezik egy LEGO NXT csomaggal, sikerrel becsatlakozhat. Jelen pillanatban saját autóink és a pálya elkészítése van a középpontban, utóbbi esetén fontos, hogy bármely más egyetemen nagyon kis munkával reprodukálni tudják a versenypályát. Mivel csak két LEGO NXT dobozzal rendelkezünk, így két autónk van: az „MI-s” (mérnök informatikus) és a „PTI-s” (programtervező informatikus) autó.

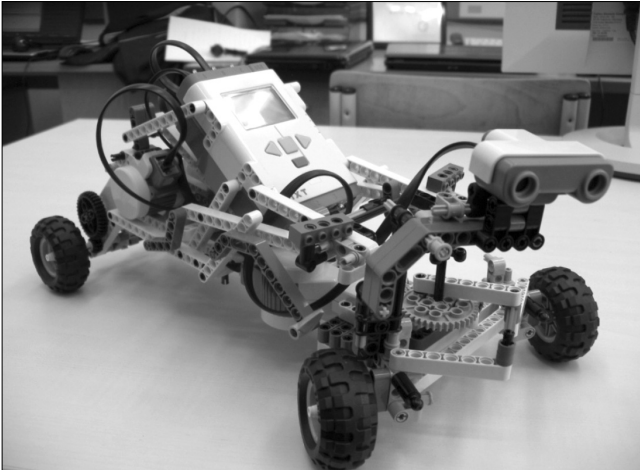
Jelen pillanatban, ezen a két kocsin verseng saját szoftverrel hat istálló. A két autót a 9. és 10. ábrán láthatjuk.

A „PTI-s” robot kialakításánál nagy hangsúlyt fektettünk a szerkezeti stabilitásra és a könnyű karbantarthatóságra. Két motor hajtja a hátsó kerekeket, egy harmadik pedig a kormányzást valósítja meg. Az autó sebességét fogaskerék-áttétellel növeltük. A jármű egy rögzített ultrahang szenzor segítségével állapítja meg hogy milyen távolságra van a hozzá legközelebbi objektumtól.

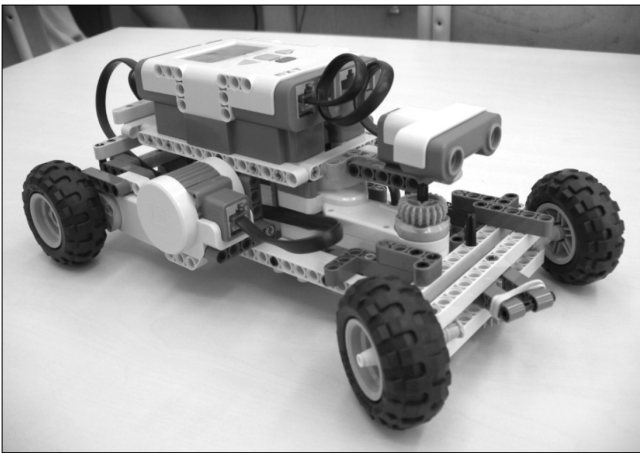
Ami az „MI-s” autót illeti, a legelső típusnál kiütközött néhány hiba, amit a második autónál javítottunk. Az új modellnél figyelniünk kellett többek között a fokozott

8. ábra
Egy korai gráf a pártok választási programjairól





9. ábra Az „MI-s” autó



10. ábra A „PTI-s” autó

stabilitásra, a hátsó kerekek áttétellel való gyorsítására, illetve a kormányműre. A szoftver írása közben ugyanis, az élet azt igazolta, hogy olyan szerkezetre van szükségünk, amellyel robotunk minden helyzetben döntésképes. Így megalkottunk egy úgynevezett „szem a keréken” technológiát.

4. Összefoglalás

A DDN nagyban segíti azzal az oktatást, hogy számos ismeretet, tapasztalatot, know-how jellegű tudást nem az oktatóknak személyesen kell átadniuk a hallgatóknak, hanem a képzés támaszkodni tud a DDN önszervező erejére, aminek során a hallgatók spontán, saját kapcsolati hálózatukban osztják meg ezeket az alapvető információkat egymással. Minél több érdekes projekttel hajtjuk meg ezt a kapcsolati hálót, az annál szélesebb és mélyebb lesz.

Köszönetnyilvánítás

Az első szerző köszönetet mond Jeszenszky Péter kollégájának, aki felhívta figyelmét a [3] irodalmi forrásra. Továbbá a szerzők szeretnének köszönetet mondani a következő hallgatóknak, akik szervesen részt vettek a két robot autó megépítésében, lehetővé téve ezzel, hogy az istállók szoftvereikkel megjelenhessenek az autókkal a Jávácska Kupán: Szabó Károly, Debreczeni Attila, Fábíán Ákos, Doszpoly Dávid és Veres Ferenc.

A szerzőről



BÁTFAI NORBERT kitüntetéses okleveles programtervező matematikus, diplomáját 1998-ban a Kosuth Lajos Tudományegyetemen, Debrecenben szerezte. Jelenleg tanársegédként dolgozik a Debreceni Egyetem Informatikai Karán, az Információ Technológia Tanszéken. 1999-ben megnyerte a Java Szövetség (Sun, IBM, Oracle, Novell és IQSoft) Java Programozási Versenyét. 2004-ben cége, az Eurosmobil első helyet ért el a Nokia és a Sun Magyarország rendezte Java ME – Java EE Fejlesztői Versenyen. 2008-ban a Vezető Informatikusok Szövetsége az Év Informatikai Oktatója cím egyikeként választotta.

Irodalom

- [1] Tanenbaum, Andrew S.,
“A UNIX clone with source code for operating systems courses”,
SIGOPS Operation System Rev., Vol. 21, Issue 1,
pp.20–29., 1987.
- [2] Tanenbaum, Andrew S., Woodhull, Albert S.,
Operating Systems Design and Implementation,
3rd ed., Prentice Hall Software Series, Prentice Hall,
ISBN 0131429388, (2006), pp.18.
- [3] “Landmark Contributions by
Students in Computer Science”, 2010.
http://www.cra.org/coc/docs/Student_Achievements.pdf
- [4] The BSD License, 2010.
<http://www.opensource.org/licenses/bsd-license.php>
- [5] “Az év informatikai oktatója” – VISZ-díj, 2008.
<http://www.mvisz.hu/index/visz-dij.html>
- [6] Bátfa N., Bátfa M.E.,
Ha hívsz, támadok! –
Java-alapú játékfejlesztés mobiltelefonra.
Híradástechnika, LX. évf., 2005/1., pp.30–32.
http://www.hiradastechnika.hu/data/upload/file/2005/2005_1/HT_0501-7.pdf
- [7] Bátfa N., Bátfa M.E.,
A mobiljáték-fejlesztés elméleti és gyakorlati momentumai.
Híradástechnika, LX. évf., 2005/5., pp.34–37.
http://www.hiradastechnika.hu/data/upload/file/2005/2005_5/HT_0505-7.pdf
- [8] N. Bátfa,
“Open source mobile games for education”,
8th International Conference on Applied Informatics,
Eger, (conference lecture), 2010.
http://www.inf.unideb.hu/~nbatfai/opensource/ICAI_OpenSourceMobileGamesForEdu.pdf

- [9] N. Bátfai, E. Bátfai, I. Psenáková,
"Jávácska One: Open source mobile games to revolutionize education of programming",
Teaching Mathematics and Computer Science,
(submitted), 2010.
- [10] Bátfai N.,
"Mobiltelefonos játékok tervezése és fejlesztése".
PhD doktori disszertáció, 2010.
<http://www.inf.unideb.hu/~nbatfai/phd>
- [11] Ars Poetica Informaticae, 2010.
<http://dev.inf.unideb.hu:8080>
- [12] A Debreceni Fejlesztői Hálózat Évkönyve, 2010.
<http://dev.inf.unideb.hu:8080/web/ddn>
- [13] M. Li, X. Chen, M.L. Xin, B. Ma, P.M.B. Vitányi,
"The similarity metric",
IEEE Transactions on Information Theory,
pp.863–872., 2003.
- [14] Cilibrasi, R., Vitányi, P.M.B.,
"The Google Similarity Distance",
DBLP:journals/corr/abs-cs-0412098, 2004.
<http://arxiv.org/abs/cs/0412098>
- [15] Cilibrasi, R., Vitányi, P.M.B.,
"Clustering by compression",
IEEE Transactions on Information Theory, 51/4,
pp.1523–1545., 2005.
- [16] Debreceni Fejlesztői Hálózat, 2010.
<http://dev.inf.unideb.hu:8080/web/ddn>
- [17] Brooks, R.A.,
"Intelligence without representation",
Artificial Intelligence, 47, pp.139–159., 1991.
- [18] Bátfai, N.,
Nehogy már a mobilod nyomkodjon Téged!
DEENK, 2008.
<http://www.eurosmobil.hu/NehogyMar>
- [19] Martin, F.G.,
"The Art of LEGO Design",
The Robotics Practitioner:
The Journal for Robot Builders, 1(2), 1995.
<http://www.cs.uga.edu/~potter/robotics/artoflego.pdf>
- [20] N. Bátfai, P. Molnár, B. Rábai, I. Tari,
Cryptographic measurements on java-enabled
mobile phones.
8th Int. Conf. on Applied Informatics,
Conference lecture, 2010.
- [21] N. Bátfai,
"Footballer and Football Simulation Markup Language
and related Simulation Software Development",
Journal of Computer Science and Control Systems
(beküldve), 2010.
- [22] Bátfai N.,
Bevezető számítások a labdarúgás szimulációs
jelölőnyelv kialakításához.
Híradástechnika, 2010/5-6. (jelen számban)
- [23] Bátfai, N.,
"On the Running Time of the Shortest Programs",
DBLP:journals/corr/abs-0908-1159, 2009.
<http://arxiv.org/abs/0908.1159>
- [24] Bátfai, N.,
"Recombinations of Busy Beaver Machines",
DBLP: journals/corr/abs-0908-4013, 2009.
<http://arxiv.org/abs/0908.4013>
- [25] Bátfai, N.,
"A Conceivable Origin of Machine Consciousness
in the IDLE process",
DBLP:journals/corr/abs-0909-5064, 2009.
<http://arxiv.org/abs/0909.5064>
- [26] M. Molnárné Nagy,
"Providing software reengineering technical expertise
based on similarity metric",
8th International Conference on Applied Informatics,
Eger, Conference lecture, 2010.