

**Az XML technológia alapjai –
XML az oktatásban, különös tekintettel a programozás oktatására**

Menyhárt László Gábor

Doktori értekezés tézisei

Eötvös Loránd Tudományegyetem
Média- és Oktatásinformatika Tanszék



Témavezető: Dr. Remzső Tibor

Informatika Doktori Iskola
Dr. Csuhaj Varjú Erzsébet

Az informatika alapjai és módszertana doktori program
Dr. Horváth Zoltán

2018

Bevezetés

Az informatika számos területén megtalálható a kiterjeszhető leíró nyelv, vagyis az eXtensible Markup Language, röviden XML [HL1, HL2]. Ezt a nyílt formátumú nyelvet először 1998. február 10-én publikálták, és az 1.0-ás verziója 2008. november 26-a óta W3C ajánlás. Az XML strukturált adatokat képes leírni szöveges formátumban, ezért is jelenik meg nagyon sok helyen. Előfordul a szabványos webes szolgáltatások kommunikációjától elkezdve a konfigurációs állományokon keresztül a végfelhasználói fájlformátumokban, vagyis megtalálható a legkevésbé sem szakértő felhasználó számítógépén is. Azért kezdtem el foglalkozni az informatikai világ „latin”-jával, mert az XML is – a latin írásmódhoz hasonlóan – rugalmas és praktikus definiálást tesz lehetővé, és hamar elterjedt. Térhódítása tapasztalható minden informatikai területen. Megjelenik az összes nagy információtechnológiai cég fejlesztésében, a webes kommunikációban, az integrációs problémák megoldásában és konfigurációs feladatok strukturált leírásánál, mégsem kap akkora figyelmet az informatikusok oktatásában, mint egy-egy – néha csak rövid időre – felkapott újdonság.

Előzmények és a témaválasztás indoklása

Már a 2003-ban informatika tanár szakos hallgatóként írt egyetemi szakdolgozatom alaptémája is az XML volt. Amikor is az XML-ről szóló és natív XML adatbázisban tárolt tananyagot készítettem el, a hozzá tartozó webes felülettel, ahol a tartalomban való keresést és az XML köré épülő technológiákat használtam fel.

A téma annyira megtetszett, hogy a doktori iskolába való jelentkezésemnél is ezt a témát választottam. „Az XML technológia alapjai” címmel kezdtem kutatásaimat az XML, mint szöveges fájlok feldolgozása, parse-olása és az XML tömörítésének lehetőségei terén.

A jelenlegi Média- és Oktatásinformatika Tanszéken végzett oktatói munkám során egyre többször kezdtem el használni az XML-t, majd publikálni módszertani eredményeimet. Mostanra forrott ki bennem, hogy az oktatás, kutatás és fejlesztés terén elért eredményeimet összegyűjtve ebben a disszertációban bemutassam.

Az oktatás módszertanban mindig is közel álltak hozzám az életszerű példák, de így az XML-lel összekötve még inkább szívesen foglalkoztam a témával. A technológia mélyebb megismerését teszi lehetővé, ha már létező függvényosztályok helyett saját implementációt

készítünk. Mivel az XML dokumentumokat szöveggként is fel lehet dolgozni, és a szövegfeldolgozás amúgy is előfordul a programozó képzésben, egyenesen következik, hogy egy gyakorlatias példának megfeleljen az XML.

A tanítási módszerek fejlesztésekor is segítségül hívhatjuk az XML-t. Az Eötvös Loránd Tudományegyetem (ELTE) Informatikai Karán futó *Programozási alapismeretek* tárgy technológiával támogatott oktatása egy újabb távolodó lépés lehet az ösztönös programozástól.

Az említett szakdolgozatomban elkészített XML alapokra épülő tartalomkezelő, oktatási anyagot tároló alkalmazás már nem elérhető, de a beletöltött tananyagot azóta is használom az évenként meghirdetett kurzusomon. A Média- és Oktatásinformatika Tanszéken részt veszek más kurzusok oktatásában is, így az általam kedvelt XML-t próbálom beiktatni más órákon is. Oktatási, módszertani szempontból olyan új ötletek fogalmazódtak meg bennem, amelyek hasznosak lehetnek.

Célkitűzés

Disszertációmban összefoglalom az XML oktatása, az XML felhasználása az oktatásban és más módszertani újítások terén tett erőfeszítéseimet. A dolgozat fő témája az XML-re és a köré épülő technológiákra alapoz. Bemutatok több oktatási segédeszközt, amit én magam fejlesztettem, és ki is próbáltam. Írok ezen szoftverek alkalmazási módszertanáról.

Tézisek

1. Tézis az XML oktatásáról

Az XML-ről és a köré épülő technológiákról szóló tananyagommal már a jelenlegi egyetemi informatikus-képzés korai szakaszában is bemutatathatóak lennének az XML technológiák alapjai, melyekre építhetnének a hallgatók a későbbi tanulmányaik során.

Kutatást végeztem, aminek során információkat gyűjtöttem össze az oktatással foglalkozó szervezetekben előforduló XML tananyagokról. Az esettanulmányom adatgyűjtési eredménye függelékben található meg a disszertációban, és szövegesen is részleteztem, hogy milyen

ismereteket szereztem, amikor utánanéztem annak, hogy hol milyen formában hogyan tanítják az XML-t.

Az adatgyűjtés módszere online kutatás volt, így azok az információk szerepelnek itt, amiket az Interneten publikáltak, még elérhetőek, és a keresésem eredményeképpen rájuk találtam. A kutatás alapjául a Google-t használtam; kezdetben az elinduláshoz szabadszöveges kereséssel, majd tematikusan az egyetemek, illetve azok oldalain belüli szűréssel. Ezek a kiinduló pontok is szerepelnek a linkek listájában. Amikor elérhető volt a website-okon a saját adatbázisukban a keresés, akkor ott folytattam. A tantárgylistákból elérhető leírások, tárgytematikák végigolvasása is előfordult. Az egyetemek oktatási anyagai között sok helyen és többször jelenik meg az XML. Néhol csak egy tárgy néhány óráján van szó róla, máshol egy teljes kurzust kitesz. Valahol csak adatrepresentációs lehetőségként említik, de strukturált adatbázisokkal is összehasonlíthatják.

Több kollégámmal észrevettük, hogy a mai informatikus-képzésben kikerülhetetlen az XML megemlítése, így újra-és-újra szóba hozzák, összehasonlításokat tesznek vele, építenek rá. De a kutatásom során talált, az XML-vel foglalkozó tárgyak közül csak pár érinti az összes XML technológiát, és nincs jelen kötelező alapozó tárgyként.

Az ELTE-n speciális kollégiumként, az ELTE-s NEPTUN-ban IKI-AXFG és TANM-INF-300-XML kódokon meghirdetett „Adatkezelés – XML” című tárgyat mutatom be, amit 2002/03-as tavaszi féléve óta tanítok az általam kidolgozott tematika és tananyag alapján.

A hallgatók visszajelzései azt mutatják, hogy szeretik a tárgyat és érdekes, hasznos dolgokat tudtak meg.

2. Tézis az XML szövegfeldolgozás oktatásához történő felhasználásáról

Az XML jó példát ad a szövegfeldolgozás oktatásához és a projektmunka használatához.

Több kurzus tematikájában előfordult a szövegfeldolgozás. Ezek gyakorlati oktatása során legtöbbször olyan egyszerű problémák megoldása a feladat, melyek nem elég látványosak. Az élmény helyett a probléma algoritmikus megoldására koncentrálnak, és mondvacsinált problémákat vagy már absztrahált feladatot próbálnak megoldani. Ilyen például a kurzushoz

ajánlott jegyzetből [HP1] egy szövegfájl szekvenciális feldolgozásához szükséges szóolvasás algoritmus implementálása.

Ebbe a témakörbe találtam ki egy új problémát, melyben XML-t kell beolvasni, és HTML-t kell gyártani az adatokból. Kidolgoztam hozzá az óravázlatot, a bemutatandó és kiosztandó tananyagot, illetve a keretrendszert, majd kipróbáltam az egyik kurzus oktatása során, és felmérést végeztem a résztvevő diákok körében. Az [SP1] cikkemben publikáltam ezt a témát.

Tehát létezik példa, ami szövegfeldolgozást oktat XML-lel. Ez egy jó példa, hiszen a visszajelzések alapján a hallgatóknak tetszik. Ráadásul a projektmunkában dolgozó második csoport jobb értékeléseket adott a kérdőív kitöltésekor.

3. Tézis egy algoritmus leíró nyelvről

Kezdő programozók oktatását segítheti egy algoritmus leíró nyelv (AML), amely lehet az XML nyelv kiterjesztése saját típusok definiálásával.

A programozás oktatása nemcsak az algoritmikus gondolkodásról szól, hanem forráskód készítésről is, természetes módon eljutva annak fordításáig, futtatásáig, teszteléséig, dokumentálásáig a hibakeresést is beleértve. Különböző iskolák mást és mást választottak ki „első nyelv”-ként, amit oktatnak. Korábban legelterjedtebb a Pascal volt, amit legtöbb helyen már felváltott valamilyen C típusú nyelv. Közben megjelentek olyan grafikus felületek, melyek próbálják látványosabbá, játékosabbá és gyerekek számára könnyebben érthetővé tenni az algoritmusokat. Vannak köztük olyanok, amelyek speciális szöveges parancsokkal működnek, és grafikus elemek jelennek meg a felületen, mint például a Logoban. De vannak már olyanok is, ahol maguk a parancsok grafikus elemek, melyeket fogd-és-vidd (drag-and-drop) módszerrel építhetünk össze. Például a Scratch, Google Blockly, Snap! vagy App Inventor 2. Ezekkel a gyerekek játékokat tudnak készíteni számítógépre és mobil eszközökre.

Az általam kidolgozott új leírónyelvvvel, amiről az [SP2] cikkemben írtam, bemutatható az oktatási folyamat elején a strukturált programozási nyelvek felépítése. Megérthetőek az egyes vezérlési szerkezetek részei. A szintaxis könnyen ellenőrizhető, sőt már szerkesztéskor a felajánlott parancsszavak használhatóak. Generálhatunk belőle forráskódot és egyéb segítő fájlokat. Mivel az XML fájl az igazán hasznos adatokon kívül több extra tag-nevet is tartalmaz, ezért sokat kellene írni, de szerencsére már a legtöbb XML szerkesztő alkalmazás létezik, mely támogatja a helyes szintaxis gyors írását már szerkesztési időben is. Ezzel az oktatást is segíti, mert anyanyelven is működhet, ellentétben az angol nyelvhez korreláló

forráskódokkal. Az XML szerkesztők kódkiegészítés mellett lehetőséget adnak az annotációkban lévő dokumentációk megjelenítésére is, ami segíti a megértést. Ez a nyelv főleges lépés lenne a haladó programozóknak, akiknek inkább már a tesztelés, nyomkövetés a fontos, amit az AML nem támogat, mert csak egyirányú kódgenerálásra van lehetőség. Úgy tekintünk erre a lehetőségre, eszközre, mint az általános iskola első osztályában a számolópálcikákra vagy tojástartóra a tízes átlépés oktatásakor, hogy csak pár bevezető órán van rá szükség. Például a *Programozási alapismeretek* tárgy első pár óráján lehetne használni bevezetésként, hogy az első forráskód írásakor ne rögtön a C-s forráskód szintaxisa rémítse el a hallgatókat, hanem kódkiegészítéssel, a forráskód strukturális felépítésének megértésével tegyék meg az első lépéseket. Mivel legtöbbször HTML szerkesztést már korábban tanulnak - hiszen érettségi témakör -, így a leíró nyelv tag-es felépítése ismert számukra, amihez kötni tudják az újonnan megjelenő vezérlési utasításokat és szerkezetüket. Így tehát, ha az AML szintaxisa nem is egyszerűbb, de már ismert felépítésű, a szerkesztők kódkiegészítő képessége pedig segít az adott helyen használható lehetőségek kiválasztásában az anyanyelven (vagy angolul) megjelenő rövid leírással.

4. Tézis az algoritmus vizualizációs alkalmazásomról

A tanulás eredményességét növelheti az algoritmusok vizuális megtekintését és szerkesztését megvalósító alkalmazásom.

A bemutatott AML tehát egy XML, amiből XSL transzformációval egyéb kimeneteket tudunk készíteni, de a létrehozáskor XML szintaktikának megfelelő szöveges fájlt kell létrehozni forrásként. Bizonyos esetekben ennél jobb lenne egy olyan alkalmazás, amiben grafikus felületen tudnánk az algoritmusainkat összeállítani. [SP3] cikkem szól erről a web-es alkalmazásról.

A strukturált programozás oktatásának elején kihagyhatatlan az algoritmikus gondolkodás és szemlélet elmélyítése. Az algoritmusok reprezentációjára több lehetőségünk is van [HP2], köztük a folyamatábra [HP3, HL3], Jackson-diagram [HP4, HP5], pszeudokóddal [HP6] és struktogrammal más néven Nassi-Shneiderman diagrammal [HP3]. A grafikus megjelenés nem feltétlenül elég, sokkal jobb lenne, ha már a szerkesztés is grafikus nézetben történne.

Először Scalable Vector Graphics (SVG) formátumú megjelenítéssel foglalkoztam, majd utána HTML formátumú ábrázolással folytattam a munkát, hogy azt még JavaScript segítségével tovább tudjam bővíteni. Elkészítettem a prototípusát egy AML

kódszerkesztőnek, mely a vizualizált NSD-ből indul, és azt frissíti minden egyes szerkesztési lépés után.

A Nassi-Schneiderman diagramok (NSD) jól használhatóak eljárások vizualizációjára anélkül, hogy egy nyelvet meg kellene tanulni [HP7, HP8, HP9]. Már több eszköz is lehetővé teszi az algoritmusok megjelenítését NSD formában, például az egyik ilyen a Structorizer [HL4]. Készült már több ilyen eszköz is BsC-s szakdolgozatként és Tudományos Diákkörben bemutatott hallgatói kutatásként, lásd [HL5, HL6]. A platformfüggetlen webes megoldások érdekelnek, ezért ilyet hoztam létre, hogy minden számítógép bármilyen mai böngészőjében működjön. Szerver-kliens platformot választottam a fejlesztés alapjául, az algoritmus a szerver oldalon van tárolva a korábban bemutatott AML nyelven. Itt történik az XSL transzformáció is, hogy az NSD HTML-ként pontosabban `div`-ekként és `table`-öként jelenhessen meg a kliens oldalon.

Az XML alapú AML-ből adott a lehetőség forráskódot is generálni, de most csak az algoritmusok megjelenítésére és módosítására koncentráltam. A bemutatott eszközzel úgy lehet szerkeszteni az AML formátumban tárolt algoritmust, hogy nem kell ismerni ezt az új nyelvet. Egérkattintásokkal indítják a szerkesztési utasításokat a felhasználók. Egy grafikusan megjelenített algoritmus könnyebben megérthető, és a szerkesztési lehetőség segíti a tanulási folyamatot is.

Van már néhány eszköz, amivel Nassi-Schneiderman diagramot lehet készíteni. Néhányukkal csak teljes algoritmust lehet rajzolni. Másokat szerkesztésre is lehet használni. A Paint vagy bármilyen más grafikai program használható, akár csak a táblázatok a Microsoft Word-ben vagy bármelyik másik irodai alkalmazásban. Microsoft Visio [HL7] egy egyszerű sablonnal támogatja ezt. Míg a SmartDraw [HL8] vagy Structorizer [HL4] alkalmazások egy algoritmus szerkesztésére is jók. Így az algoritmus közepe is bármikor módosítható, és a diagram újrarajzolódik minden alkalommal, amikor az szükséges. Ezen alkalmazások használatához legtöbbször le kell tölteni és telepíteni azokat. Mostanában többen is elkezdtek web-es alkalmazásokat kidolgozni e területhez.

Az általam készített eszköz egy platformfüggetlen kliens-szerver megoldás, amivel a felhasználók telepítés nélkül, böngészőben tudják megjeleníteni és szerkeszteni az algoritmusokat. A képernyőn megjelenő NSD képét akár a dokumentációban is elhelyezhetik. A diákoknak ez jó választás lehet, hogy könnyen tudják az NSD-eket generálni. Hasznos lehet azoknak a tanároknak is, akik egyszerű eszközzel szeretnék algoritmusokat megjeleníteni.

Természetesen a weboldalt olyan funkciókkal bővítettem, mint a felhasználó-menedzsment és képként való mentés.

A tézis elejét már mások is publikálták, például Törley Gábor a disszertációjában [HP10], míg a saját fejlesztésű alkalmazás a szerkesztést támogató alkalmazásomról bemutattam, hogy az algoritmusok strukturáját NSD formában megjeleníti és szerkeszteni is lehet.

5. Tézis a programozási folyamat oktatását támogató alkalmazásomról

A szakmában elterjedt, módszertanok által használt eszközök analógiájára épülő alkalmazásom támogatja a módszeres programozás oktatását.

A *Programozási alapismeretek* című tárgy óráin olyan informatikai feladatokat kell a hallgatóknak megoldani, amihez általában szöveges leírást kapnak, ebből készítik el a specifikációt, algoritmust majd implementálják a kódot. Azon kezdtem el gondolkodni, hogy hogyan tudom támogatni a hallgatók gondolkozását, és segíteni a megoldandó feladatok nyilvántartását, esetleg gyorsítani az implementáción, illetve a dokumentációkészítésen. Ennek a problémának a megoldásán az előző algoritmusvizualizáló és szerkesztő programmal párhuzamosan kezdtem foglalkozni. Majd később összeépítettem a két ötletet.

Az egyetemen az informatikushallgatók programozás-oktatása a teljes fejlesztési folyamatot le kell, hogy fedje, de a módszeres programozás folyamatában nagy hangsúlyt kell fektetnünk az algoritmizálásra. [HP11] Azt tapasztaltam, hogy a hallgatóim legtöbbször csak a kódolás lényegi része után érdeklődik, ami a feladat szövegéből jön, és próbálja elkerülni a specifikációkészítést és algoritmusrajzolást. Az volt az ötletem, hogy nagyobb figyelmet szentelnének a hallgatók a specifikációra és az algoritmusra, ha azzal segítenénk a későbbi feladataik megoldását. A támogatás lényege az algoritmusból történő kódgenerálásból és a dokumentáció fő elemeinek előkészítéséből áll. Első lépésben egy Excel táblázat-ban kell összegyűjteni az információkat, amiből a dokumentáció legenerálható a program használatának bemutatása vagyis képernyőképek nélkül. Először az [SP4] cikkemben számoltam be erről részletesebben.

A második lépésben már web-es platformon futó összeépített alkalmazást az [SP5] cikkemben mutattam be. Hallgatóimmal kipróbáltattam az alkalmazás használatát. Ekkor még csak a programozási nyelv alap szintaktikájáról tanultak. Volt egy csoport, ahol angol nyelven folyt

az oktatás, mert a hallgatók több különböző országból érkeztek (Anglia, Brazília, Irán, Kamerun, Macedónia, Németország, Szaud-Arábia, Törökország, USA, Vietnám, ...). Itt különösen jó volt, hogy az algoritmust nem pszeudokóddal írtuk, mert mindenki anyanyelvét nem ismertük, angol nyelven pedig zavart okozhatott volna, hogy nagyon hasonlít a forráskódban megadandó programozási nyelv kifejezéseire. [HP12]

Nem önmagában a programozás eszközt (programozási nyelv és környezete) kell megtanítanunk, hanem magát a programozást. Ki kell alakítanunk a hallgatóságban egy gondolkodási módot, egy követhető programozási stílust. A módszeres programozás folyamatában nagy hangsúlyt kell fektetnünk az algoritmizálásra. [HP11] Fontos, hogy a hallgató értse az algoritmust, és azt valamilyen nyelven le is tudja írni. Meg kell, hogy értsék az alapalgoritmusokat (programozási tételeket), és azokat bonyolultabb algoritmusok elkészítéséhez építőelemekként fel tudják használni. Célszerű ezért ezeket az algoritmusmintákat elemezni, ellenőrizni és számon kérni. Az oktatás során fontos, hogy csak az algoritmizáló készségek kialakítása után kezdjünk programot írni, de a gyakorlati kipróbálás miatt szükséges, hogy párhuzamosan fejlesszük a hallgatóság programozó képességeit is.

Bemutattam a megvalósított ötleteimet, a vonalvezetőként használható Excel alapú támogató csomagot. Majd ezt építettem össze egy Web-es platformon az előző tézisben szereplő algoritmust Nassi-Shneidermann diagramként szerkesztő alkalmazással. Megválasztam a közben felmerült kérdéseket, a tapasztalataimról és a visszajelzésekről írtam. A PFW pozitív fogadtatása és a visszajelzések alapján állíthatom, hogy segíti a módszeres programozás oktatását.

A szerző publikációi

- [SP1] Menyhárt L.: Teaching string processing by using XML in: Nagib Callaos , Hsing-Wei Chu , Belkis Sánchez , Michael Savoie , Andrés Tremante (szerk.) Proceedings of the 7th International Multi-Conference on Engineering and Technological Innovation . Konferencia helye, ideje: Orlando , Amerikai Egyesült Államok , 2014.07.15 -2014.07.18. Orlando: International Institute of Informatics and Systemics (IIS), 2014. pp. 73-76., ISBN:978-1-941763-07-0
- [SP2] Menyhárt L.: Can a language be before "the first programming language"? in: TEACHING MATHEMATICS AND COMPUTER SCIENCE 9:(2) pp. 209-224., 2011

- [SP3] Menyhárt L.: Nassi-Schneiderman diagram in HTML based on AML In: ACTA DIDACTICA NAPOCENSIA 6:(3) pp. 19-26., 2013, ISSN: 2065-1430
- [SP4] Menyhárt L., Pap Gné.: How can we get our students to think while we help their work too?: Document based development In: Nagib Callaos , José Vicente Carrasquero , Ángel Oropeza , Andrés Tremante , Fridrich Welsch (szerk.) Proceedings of the 7th International Multi-Conference on Society, Cybernetics and Informatics . Konferencia helye, ideje: Orlando , Amerikai Egyesült Államok , 2013.07.09 -2013.07.12. Florida: International Institute of Informatics and Systemics (IIS), 2013. pp. 97-102., ISBN:ISBN-13:978-1-936338-83-2
- [SP5] Menyhárt L., Pap Gné.: Presentation of improved version of guide application for teaching programming fundamentals In: Nagib Callaos , Hsing-Wei Chu , Belkis Sánchez , Michael Savoie , Andrés Tremante (szerk.) Proceedings of the 7th International Multi-Conference on Engineering and Technological Innovation . Konferencia helye, ideje: Orlando , Amerikai Egyesült Államok , 2014.07.15 - 2014.07.18. Orlando: International Institute of Informatics and Systemics (IIS), 2014. pp. 77-82., ISBN:978-1-941763-07-0

Hivatkozások

Publikációk

- [HP1] Pap, Gné., Szlávi, P., Zsakó, L.: Módszeres programozás – Szövegfeldolgozás, 14. Mikrológia, 2000
- [HP2] Furman, B. J.: Notes on Algorithms, Pseudocode, and Flowcharts, 2010, http://www.engr.sjsu.edu/bjfurman/courses/ME30/ME30pdf/Notes_on_Algorithms.pdf
- [HP3] Nassi, I., Shneiderman, B.: Flowchart techniques for structured programming, ACM SIGPLAN Notices, Volume 8 Issue 8, 12–26, 1973
- [HP4] Jackson, M.A.: Principles of Program Design, Academic Press, 1976
- [HP5] Jackson, M.: A system development method In Cambridge University Press, 1982, http://www.ferg.org/papers/jackson--a_system_development_method.pdf
- [HP6] Cormen, T. H., Leiserson, C. E., Rivest R. L.: Algoritmusok, Műszaki Könyvkiadó, Budapest, 1997, p 2-4.
- [HP7] Hendrix, T.D., Cross, J.H.: "Language Independent Generation of Graphical Representations of Source Code", Proceeding CSC '95 Proceedings of the 1995 ACM 23rd annual conference on Computer science Pages 66-72, 1995

- [HP8] Csepregi, Sz., Dezső, A., Gregorics, T., Sike, S.: Automatic Implementation of Service Required by Components, Workshop on Property Verification for Software Components and Services, PROVECS, <http://lina.atlanstic.net/provecs/2007/provecs2007proceedings.pdf>, 2007
- [HP9] Averbukh, V., Bakhterev, M.: The analysis of visual parallel programming languages, ACSIJ Advances In: Computer Science: an International Journal, Vol.2, Issue3, No. 4, 2013, ISSN : 2322-5157
- [HP10] Törley, G.: Vizualizáció a Programozástanításban, doktori disszertáció, 2014, <http://www.doktori.hu/index.php?menuid=193&lang=HU&vid=12890>
- [HP11] Szlávi P., Zsakó L.: Módszeres programozás: Programozási bevezető, 18. Mikrológia, 1996
- [HP12] Weise, M.: A Model for Teaching Informatics to German Secondary School Students In: English-language Bilingual Education, Proceedings of the 6th International Conference ISSEP 2013; Oldenburg, Germany, February 26–March 2, 2013/Diethelm et al. (Eds.)/ Potsdam: Universitätsverlag Potsdam, 2013/ S.127-137

Linkek

- [HL1] W3C - Extensible Markup Language, <http://www.w3.org/XML/>
- [HL2] XML - W3C Recommendation, <http://www.w3.org/TR/xml/>
- [HL3] Flowchart, <https://en.wikipedia.org/wiki/Flowchart>
- [HL4] Fisch, B.: Structorizer, <http://structorizer.fisch.lu/>, 2009-
- [HL5] Molnár T.: StukiMania, <http://stukimania.hu>, 2012-
- [HL6] Németh B.: DAVIK, (Dinamikus Algoritmus Vizualizációs Keretrendszer) Dinamic Algorithm VIsualisation Framework, OTDK: http://otdk2013.gdf.hu/sites/default/files/OTDK_2013_Informatika_Tudomanyi_Szekcio_GDF_Programfuzet.pdf, szakdolgozat: http://www.inf.elte.hu/karunkrol/szervezet/dekanihivatal/to/diplomamunkak/diplomamunkak/Lapok/11122_diplomak.aspx
- [HL7] Microsoft Visio Template for Nassi-Schneiderman Diagram, <http://users.tm.net/tonietienne/>
- [HL8] SmartDraw, <http://www.smartdraw.com/resources/tutorials/nassi-shneiderman-diagrams/>