

Vizuális Programozás

szigorlati dolgozat, MOME, 2007. készítette Nagy Ágoston
iv., évfolyam, vizuális kommunikáció, média design.
opponens: Babarczy Eszter

Háttér

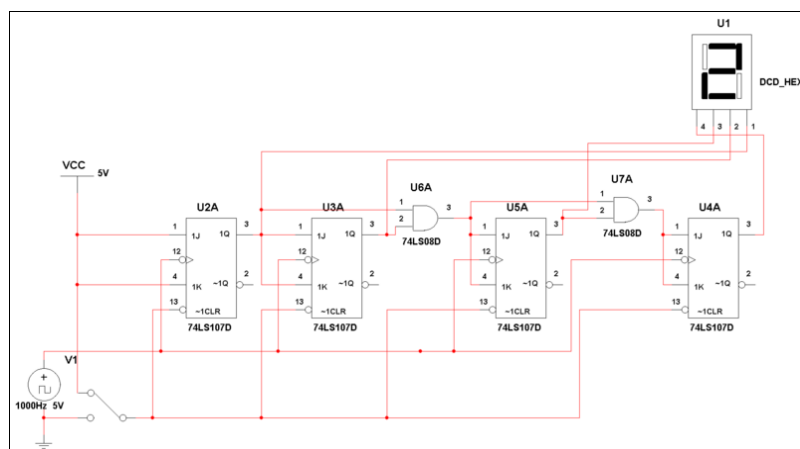
A számítógépekkel való kommunikáció alapja a bináris értékek alakításából, programozásából áll. Általában ezek a gépi kódok előlünk rejtve maradnak, egy absztrahált, felhasználóbarát felülettel érintkezünk, amikor szöveget, képet, hangot, vagy más szerkesztünk. Ezek a tevékenységek a felszín alatt számok manipulálását jelentik, és ennek kulturális aspektusai lassan-lassan a professzionális programozói körökből a mindennapokba is átszivárognak. A programozás alapvetően egy hierarchikus rendszerre épül, ahol a legalsó szinten található a gépi kód (a sokat hangoztatott egyesek és nullák...), és erre épülnek az egyre összetettebb (egyúttal az emberi nyelvre is egyre inkább hasonlító) mesterséges nyelvek. A jövőben várhatóan mindenki más aspektusból is meg fog ismerkedni a programozással, a kultúra, tervezés szerves részévé alakul egy-egy nyelv valamilyen szintű ismerete.

A hagyományos programozói nyelvek egymást követő sorokból állnak, amelyek utasításokat tartalmaznak. Feltételek, ciklusok, elágazások nagyon szigorú nyelvtanba rendezve, amely igen szabad, tág lehetőséget biztosít annak, aki alkalmazza, ám rengeteg időt és energiát igényel a használatukhoz elegendő tudás elsajátítása. Valaki, aki nem olyan jártas a matematikában, a logikában és a programozásban, viszont érdeklik a technológia adta lehetőségek, és integrálni szeretné ezeket a munkájába, egyszerűbb, áttekinthetőbb felülettel könnyebben boldogul. MILLER PUCKETTE¹ a MAX (akkor még Patcher) névre hallgató vizuális nyelvet eredetileg azon megfontolásból kezdte fejleszteni, hogy a zenészek számára egy áttekinthető keretrendszert biztosítson ahhoz, hogy kipróbálhassák elméleti tanulmányaikat a gyakorlatban, mindenféle előzetes programozói ismeret nélkül.

A dolgozatban főként a képekre és a hangokra specializált nyelveket ismertetem, természetesen ezek a rendszerek fizikai programozásra, tiszta adatok szervezésére is használhatóak: az utóbbi években kialakultak standard protokollok, jellemző a szabad átjárhatóság, összekapcsolhatóság, technológiai, társadalmi, kulturális viszonylatban egyaránt.

Vizuális Paradigma

A vizuális programozás azt a módszert jelenti, amikor a hagyományos szöveg helyett grafikai illusztrációkkal hozzuk létre az adott programot. A gyökerei a DSP² algoritmusok grafikai megjelenítésében, illetve analóg eszközök (áramkörök) technikai leírásához alkalmazott diagrammokban, vagy az analóg szintetizátorok moduláris, „kötögetős felületeiben” keresendők. Ily módon kézenfekvő jelfeldolgozórendszerek, „adatfolyamok” algoritmusainak reprezentálására és tervezésére, vizuálisan nagyon egyszerűen nyomon követhető az adat haladási iránya, akár az áram útja az áramkörben, vagy a véré az érrendszerben.



1. ábra: négy bites számláló elektromos diagramja



2. ábra: Robert Moog³ moduláris szintetizátora

Az első grafikus programozói nyelvet WILLIAM ROBERT SUTHERLAND készítette 1966-ban. Ám a legjelentősebb fejlesztések jobbára a nyolcvanas években történtek, a processzorok teljesítménye és az eszközök elérhetősége lényeges szempont. B.A. MYERS *Vizuális Programozás Taxonómiája és Programvizualizációk* című cikkében a következőképpen definiálja a fogalmat:

A Vizuális Programozás egy olyan rendszert jelent, ahol a felhasználó két (vagy több) dimenzióban szerkesztheti a programot. [...] a konvencionális szöveg-alapú nyelvek nem tekintendők két-dimenziósoknak, mivel a kompilerek⁴ hosszú, egydimenziós folyamatokként kezelik őket.

Az adat áramlásának követhetősége, a modulok egymáshoz rendelése, összekötése kifejezetten alkalmas prototípusok gyors készítésére. Ebben az esetben egy szerkezeti vázat létrehozunk, módosítanak, fejlesztenek, majd a kész verziót megírják hagyományos programnyelven, például C-ben. Ez azért van így, mert a magas szintű vizualizáció (az összetett architektúra, grafikai felület kezelése) egyrészt lassít a rendszeren, az alap rendszer-funkciók csak áttételesen érhetőek el. Másrészt egy kész alkalmazásnál már nem feltétlenül van szükség a szerkezet láttatására. Harmadrészt ezek a nyelvek interpreterek, vagyis olyan programok (a szoftver-technológia klasszikus értelmében), amelyek nem alakítják a forráskódot egy közvetlenül futtatható fájlba, hanem folyamatosan olvassák, analizálják és futtatják a kódot.

A prototípus-készítés esetében tehát a rendszer folyamatosan működik, és közben valós időben módosíthatunk a grafikus felületen, elvehetünk, hozzáadhatunk, variálhatunk a tartalom, és így közvetlen visszacsatolást kapunk a rendszer aktuális állapotáról, tehát a (kisebb) hibák működés közben korrigálhatóak. Feltehetően ez az interaktivitás a legnagyobb előnye e nyelveknek. A tervezés vitathatatlanul egyszerű velük, de le kell szögeznünk, hogy összességében messze állnak attól, hogy a hagyományos szöveg-alapú nyelvekkel (azok flexibilitásával, hatékonyságával) versenyre keljenek.

GÜNTER GEIGER (a Pure Data programnyelv egyik jelenlegi fejlesztője) szerint a legjobb megoldás egy hibrid rendszer lehet, ahol szöveg-alapú szubrutinok, objektumok integrálhatóak a vizuális környezetbe. Több példa is létezik már erre: részint integráltak a Pure Datába Python, Ruby

nyelv kiterjesztéseket, a MAXMSP nevű vizuális programnyelvbe implementálták a Java-t.

„A keverés, a hozzárendelés, a ritmus, a frekvencia, az összeillesztés, a folyamat, a kölcsönhatás a tudományos eljárások alapjai, amelyek bármire vonatkoztathatóak, mikro és makro szinten egyaránt. Emiatt a zene és a hang kitüntetett szereppel bír a médiaarcheológiában.”

Állítja SIEGFRIED ZIELENSKY *Deep Time of the Media* (A média ideje) című könyvében. Nem véletlen, hogy a korábban említett nyelvek eredeti funkciója is a zenével és a hangokkal, a hangok modellezésével foglalkozott, és filozófiai gyökerei a szintetizátorokhoz és az áramkörökhöz köthetők. A zene kapcsolata a tudományokkal mindig az élvonalbeli technológia mentén körvonalazódott: MARINNE MERSENNE⁵ a kombinálást és a komponálást ekvivalens kifejezésnek tekintette, és már a XVI-XVII. században mutatkozott igény kompozíciós gépek előállítására.

A vizuális programnyelvek korai generációi is zenei, akkusztikai problémák modellezésére szolgáltak. MILLER S. PUCKETTE az IRCAM⁶-nél megírta a Patcher nevű vizuális programnyelvet, amely MIDI⁷ adatokkal kommunikált elektronikus hangszerekkel és más gépekkel. A rendszer később a MAX nevet kapta, MAX MATHEWS után (Ő alkotta a MUSIC nevű programot 1957-ben. Ez az első ismert digitális szintézisprogram).

A lényege a már említett, valós idejű koncepción alapult: az értékek, hozzárendelések szabad variálhatósága, a zenei produkció folyamatával egyidejűleg. A komponálás és az előadás egyesültek, és a számítógépes zenében is teret nyert az improvizáció fogalma. 1989-ben az IRCAM egy konkurens rendszert bocsátott kereskedelembe, MAX/FTS (Faster Than Sound) néven, amely a későbbi, jelenleg világszerte alkalmazott multimédia-rendszer, a MAX/MSP (itt már: Multimedia Signal Processing) elődje.

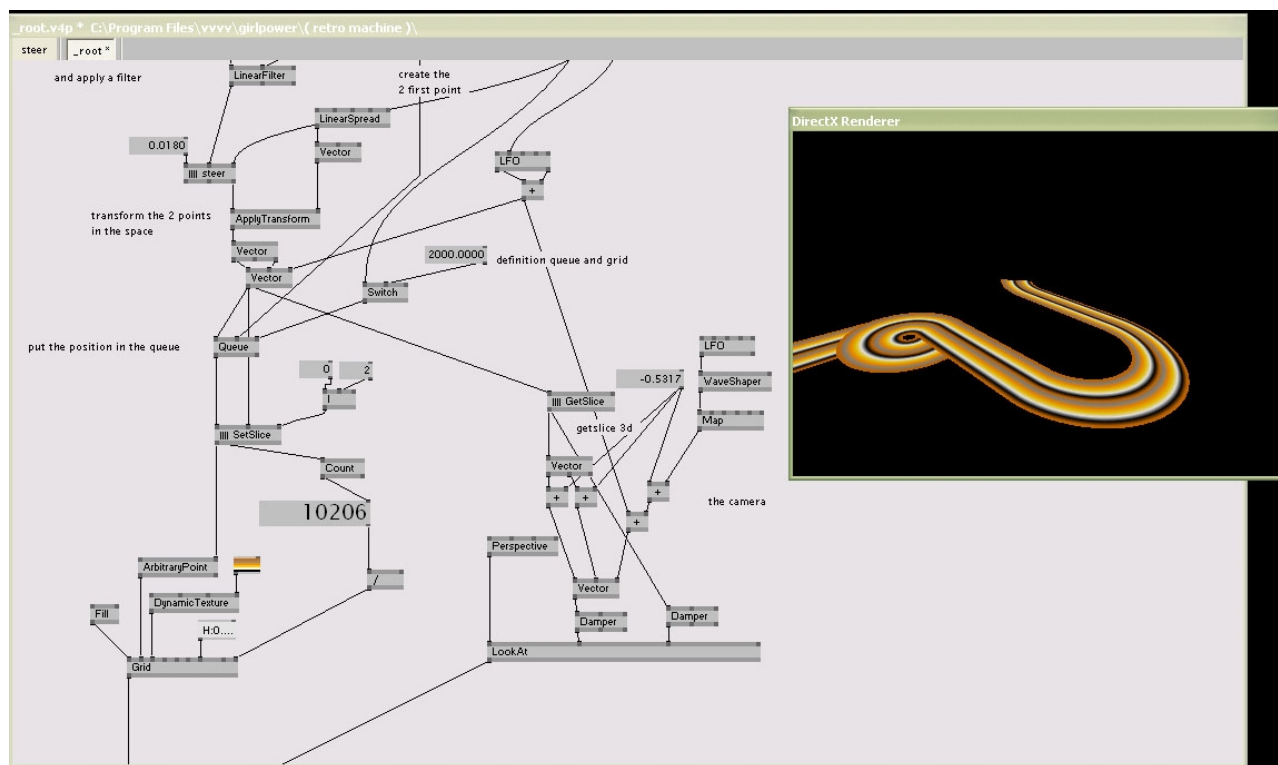
A jelenlegi MAX programot DAVID ZICCARELLI fejleszti és forgalmazza csapatával, a Cycling 74-el. MILLER PUCKETTE viszont úgy döntött, hogy a kereskedelmi szemlélettel ellentétben újraírja az egészet, és közzé teszi fejlesztései forráskódját. Így egy másik ágon megszületett a nyílt forráskódú, ingyenes Pure Data.

Léteznek még más zenei vizuális programnyelvek, például a Native Instruments által fejlesztett Reaktor, vagy a szintén az IRCAM-nál fejlesztett, ingyenes OpenMusic, de a dolgozat inkább a korábban említettekre koncentrál, egyrészt, mert a logikai és a filozófiai különbség nem túl nagy köztük, másrészt mert a MAX-hoz és a Pure Data-hoz később készültek képpel foglalkozó rendszerek is.

A MAX/MSP-hez eredetileg egy Nato nevű kiterjesztést kezdett fejleszteni NETOCHKA NEZVANOV⁸. A quicktime rendszeren alapuló objektumok valós idejű video manipulására szolgáltak. A projekt 2003-ban leállt és ma már nem létezik (a jelenlegi operációs rendszereken nem fut). Később, már a Cycling 74-nél bemutatták a Jitter nevű kiterjesztést, és mind a mai napig ezt használják standard video-modulként MAX alatt.

A fejlesztés mondhatni gőzerővel történik, a jelenlegi (1.6-os) verzióban a három dimenziós animációk széles tárháza, mátrix-alapú videófeldolgozás, effektek és szűrők, procedurális rajzolásmódok, szkriptelés, kép/hang konverzió, adattárolás és hozzárendelés, Java implementáció (saját objektumok írásának lehetősége Java nyelven) egyaránt megtalálhatóak.

A legrégebbi, és legstabilabb képi kiterjesztés a MARK DANKS által fejlesztett GEM (Graphic Environment for Multimedia). Ez leginkább három dimenziós, OpenGL-alapú⁹ animációkat képes renderelni, valamint textúráként alkalmazhatóak filmek és az élő kamera által közölt jelek. Ezen kívül a linux rendszeren alapuló PDP (Pure Data Packet) az elsőszámú videofeldolgozó kiterjesztés, amely a GEM-mel kombinálva elméletileg hasonló lehetőségeket biztosít, mint a Jitter, ingyen. Természetesen a gyakorlatban a Jitter klasszissokkal erősebb és sokoldalúbb, de hát ennek meg is kéri az árát.

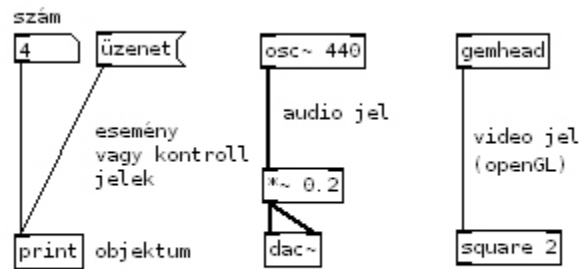


5. ábra: VVVV

Más, videofeldolgozással foglalkozó nyelvek közül figyelemre méltó az Apple Quartz Composer-e, amelyet az OSX operációs rendszerbe integráltak, és alapszintű műveleteket lehet végezni egy vizuális felületen. Ezen kívül az utóbbi idők termése a Windows operációs rendszeren alapuló VVVV névre hallgató programja, amelyet a német Meso cég fejleszt, ez is ingyenesen letölthető és alkalmazható. Főként videóra és animációra specializált nyelv, amely gyorsabban renderel a Jitter sokszorosan áttételes, robosztus architektúrájánál.

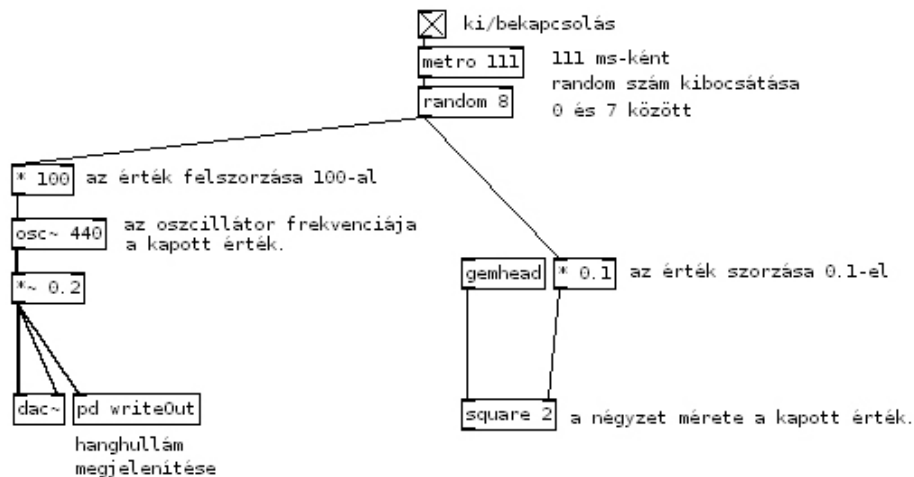
Működési elvek, közelebbről

A vizuális programok működési mechanizmusa némiképp eltérhet, de az alapkoncepció lényegében azonos. A Pure Data (Pd) mechanizmusát szemléltetném, annak érzékeltetésére, hogy mit is jelent a korábban tárgyalt, valós idejű működés. A Pd adatfolyam gráf-elven működik. A gráf-elméletben gráfnak nevezik azt az ábrát, amely pontokból és ezek közül némelyeket összekötő vonalakkól áll. A pontok a gráf pontjai, vagy csúcsai, a csúcsokat összekötő vonalak a gráf élei. A Pd-ben a gráf műveletek együttese, amelyek ábrázolása dobozokat és más grafikai elemeket összekötő vonalakkal történik. A gráfban két egymástól független adattípus működik: az egyik a jel szintje (hang, vagy képi), a másik az esemény szintje (üzenetek, számok). Működés közben egy ütemező folyamatosan összegyűjti és szekvenszálja, majd futtatja az üzeneteket. Ennek kialakítása esszenciális lépés volt a valós idejű programozás felé, így válhatott a komponálás (gráf szerkesztése, „patching”) az előadás részévé.



6. ábra: A Pure Datában alkalmazott üzenettípusok, és néhány objektum. A [print] kinyomtatja egy ablakba a kapott üzenetet. Ez jelen esetben a számra kattintva az aktuális érték (kurzorral állítható), az üzenetre kattintva pedig az üzenet-doboz tartalma. Az [osc~] egy oszcillátor, ami jelenleg a zenei „A” hangon rezeg (440 Hz). A [*~] a hang-jel szorzására (erősítésére) szolgál, a [dac~] pedig a digital/analog converter (d/a átalakító). Ez a hangkártya kimenetére küldi a kapott hangot. A [gemhead] egy openGL alapú, háromdimenziós renderelés „feje”, jelen esetben egy négyzetet renderel egy ablakba.

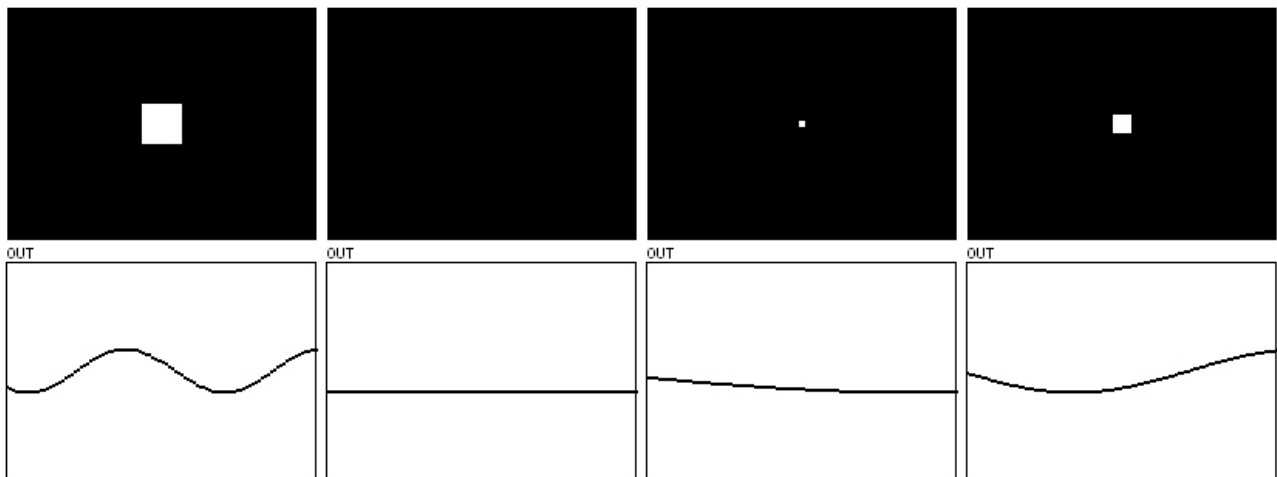
Az objektumok paraméterei, útvonalai szabadon variálhatóak, ilyen módon különböző összetevők között, például a hang és a kép között szoros kölcsönhatás alakulhat ki. Ha az ábrán alkalmazott képletet kicsit átalakítjuk, egy egyszerű, valós idejű animációt készíthetünk. Mivel egyelőre nincsen semmilyen időkezelő egységünk a gráfban, nincs mozgás, egyedül az oszcillátor rezeg (ha elindítjuk a rendszert).



7. ábra: Egy temporális elem, a [metro] beemelése az ábrába.

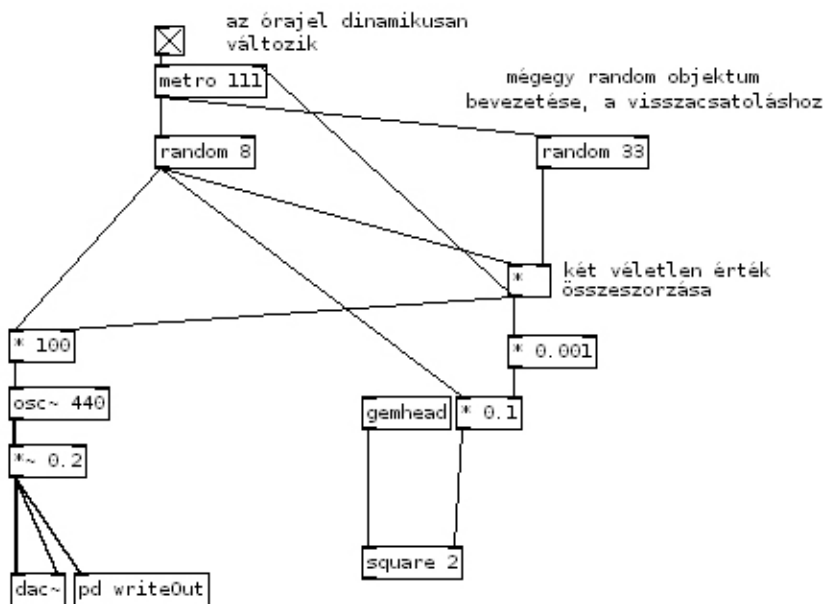
A [metro] objektum adott időközönként (jelen esetben 111 miliszekundum) kiad magából egy impulzust. Ez mozgásba hozza a rendszert, hiszen így folyamatosan változik az oszcillátorunk frekvenciája, valamint a négyzetünk mérete (az eredetileg megadott 440, illetve 2 értékek automatikusan felülíródnak a legutóbb kapott értékekkel). A [pd writeOut] objektum egy „szub-patchot” takar, ami azt jelenti, hogy benne további objektumok találhatóak. A pd előtaggal lehet egymásbaágyazott patch-okat készíteni, és ezáltal komplex műveletek is elférnek a képernyőn, többdimenzióssá válik a szerkesztőfelület. A kapott hangot sajnos a

dolgozatban nem tudom közölni, így a hanghullám és a renderelt négyzet ábrájából közlök néhány állapotot.



8. ábra: A patch által létrehozott képek és hangok. A felső szinten a megjelenített négyzet aktuális állapota, az alsó sorban a vele párhuzamosan kiadott hang hullámképe látható.

A rendszer működése révén előre definiált időközönként kapjuk az értékeket. Egy lehetőség a dinamikus működés felé a visszacsatolás lehetősége. A [random] objektum által kapott értékek visszaetetése a [metro] objektumba a temporális összetevő dinamikus változását eredményezi, ehhez nem szükséges még egy véletlengenerátor beemelése, a példában azért jártunk el mégis így, hogy a kapott értékek skálája kitáruljon (több véletlengenerátor kombinálása nagyobb esetlegességet eredményez), és ezáltal jobban érzékelhetővé váljon a kettő közötti különbség.



9. ábra: Dinamikus időkezelés

A Rendszer, mozgásban

Az eddig említett képi és a hangi vetületek jobbára valamilyen virtuális közegben valósulnak meg, projektorokon, monitorok képernyőin, hangfalakon. Összességében a fejlett országok hétköznapi kellékeit alkalmazzák, amelyek egy cég jól felszerelt irodájában megtalálhatóak. Talán éppen emiatt a szociológiai jelenség miatt válik egyre fontosabbá az alternatív média használata, amelyet a választott anyag szellemiségében is kísérletező, játékos szemléletmód jellemez. Szokás augmentált valósággént definiálni a jelenséget: a virtuális valóság izolációjával ellentétben az augmentált valóság a fizikai valóság részévé teszi a gép által kezelt utasításokat, a gép kimeneti egységei helyett tárgyakban, terekben jeleníti meg azokat. A számítógépet ilyenkor tisztán vezérlésre, analízisra, számolásra használják, és az előállított alkotás megszabadul a beidegződött multimédia preconcepcióktól. A digitális adat egyszerű módszerekkel elektromos impulzusokká, analóg jelekké alakítható, így a gépen alkalmazott rendszer az áramerősség változtatásával képes fényeket, motorokat, robotokat, analóg eszközöket vezérelni. Ez természetesen fordítva is így áll: a szenzorokon keresztül kapott analóg jelek digitalizálásával a numerikus értékek képi, hangi, vagy kontrol-adatokhoz rendelhetőek. Például hasonló dinamikus rendszernek tekinthető a hétköznapi életünkből a fotocellás ajtó.

Az alternatív média megválasztása önmagában is állásfoglalás. Az újrahasznosítás, a meglévő tárgyak újraértelmezése, átkontextualizálása nem újkeletű jelenség, de mindig tartogat érdekességeket. Az elektronikus média esetében a meglévő eszközök (elektromos játékok, régi televíziók, rádiók, motorok) átalakítása, „hackelése”, a szemét felhasználása egyre népszerűbb. A motorok, szenzorok, aktuátorok számítógéppel való vezérlését hívják fizikai programozásnak. Az utóbbi években nagy hangsúlyt fektettek fejlesztők arra, hogy a fizikai (analóg) eszközök és a számítógépen alkalmazott, valós időben számoló programok közötti átjárás könnyebb legyen, és az elektromérnökök és programozók mellett tudják alkalmazni olyanok is, akik nem képzettek ilyen technológiai területen. A hackelés, a „do it yourself”¹¹ mentalitás magában hordozza a finansiális aspektusokat is: olcsó, akár ingyen szerzett alapanyagok átkontextualizálása, integrálása a rendszerbe.

Az eszköz tulajdonságainak ismerete és ezek átfordítása saját igényekre egy kreatív folyamat, amely egy iteratív tervezési, illetve tesztelési diskurzust kíván az alkotó és az anyag között. SOMLAI FISCHER ÁDÁM SZABOLCS és USMAN HAQUE egy mindenki által könnyen értelmezhető cikkben összegyűjti és rendszerezi az olcsó játékok adta kombinációs lehetőségeket, és leírja, miként lehet ezekből egyszerű dinamikus rendszereket, tereket létrehozni. A címe (*Low-tech¹² szenzorok és aktuátorok*) a korábban említett, alacsony költségvetésű mentalitásból ered. Két okból tartom fontosnak a cikket a vizuális programozás szempontjából:

Egyrészt a „nyílt forráskódú tervezés” fogalmával operál, ami ezek olvasatában azt jelenti, hogy az alkalmazott eszközöknek mindenki által elérhetőnek kell lenniük (például hétköznapi használati tárgyak), ezáltal szabadon módosíthatóak és újraformálhatóak. A módszer, amely alapján ezek a módosítások zajlanak, szintén nyílt, hozzáférhető kell, hogy legyen (például online dokumentáció formájában).

Másrészt a konceptuális keret, amelyen ezek a low-tech rendszerek alapulnak, megegyezik a vizuális programozás alapkoncepciójával, ami szerint a különböző objektumok összekötésével egy olyan rendszer építhető, amelyekben alrendszerek, szub-modulok képezhetőek egy-egy objektum együttes által. A cikk az alkotóelemeket három csoportba osztályozza: szenzorok, aktuátorok, és komparátorok (amelyek az előző kettő között átalakítják, továbbküldik a kapott jeleket).

A negyedik komponens a visszacsatolás. Talán a legfontosabb eleme egy dinamikus rendszernek: a folytonosan változó aktuális állapot visszahat a rendszerre, és ezáltal változik a működése. Ez a változás szintén változtat...



10.-11. ábra: Példák a visszacsatolásra. Baloldalt egy video-visszacsatolás, jobboldalt egy csigaház szerkezete. Mindkét esetben látszik, hogy a rendszer aktuális állapota (mérete, formája) a korábbi állapottól függően módosul.

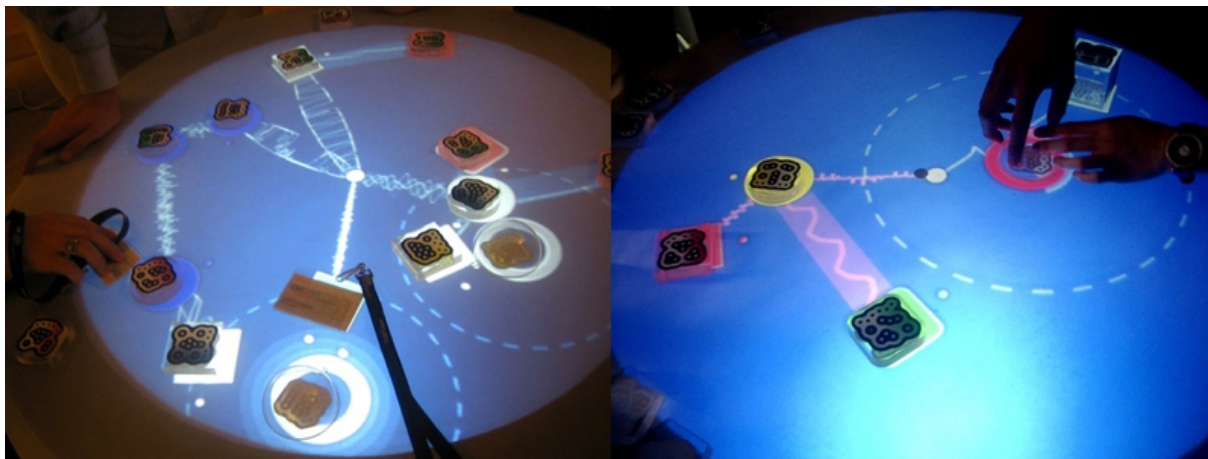
NORBERT WIENER, magyar származású matematikus és mérnök a kibernetika alap-mechanizmusaként tartja számon a visszacsatolást és az entrópia¹³ elvét. Sőt, *Human Use of Human Beings* (Emberekkel bánni, emberien) című, 1954-es esszéjében továbbmegy:

Tézisem, hogy az élőlények fizikai funkcionalitása és az újabb kommunikációs gépek működése rendkívül hasonlít abban, hogy visszacsatolások által kontrollálja az entrópiát.

Hálózatban, többen

Lényeges szempont, hogy a különböző alkotóelemek, modulok kapcsolatban vannak egymással, és adott esetben visszahatnak egymásra. Ez egy programon belül is érdekes, ám akkor válik igazán érdekessé, amikor több különböző, hasonló felépítésű rendszer kommunikál egymással. A multimédia-rendszerek összekötésére különböző protokollok léteznek, az egyik legelső, a már említett MIDI protokoll. Ez igen csekély adatot tud egyidejűleg továbbítani, és ezek mindegyike egy-egy diszkrét szám. Miután az utazó adatok mennyisége növekszik, illetve az említett rendszerek a kontroll adatok (számok, szimbólumok, sorozatok) mellett összetettebb, nagyobb sávszélességű adattípusokkal (kép és hang) is operálnak, szükségszerűen fejlődnek újabb megoldások. Az egyik elterjedt protokoll az OSC (Open Sound Control, nyersen fordítva „nyílt hang vezérlés”), amely szintén kontroll adatok közvetítésére hivatott, ám a sávszélessége sokkal nagyobb, és a MIDI-vel szemben összetettebb adattípusok (szimbólumok, stringek, avagy hosszú karakterláncok) is küldhetőek, fogadhatóak. Az interneten is működő, tcp/ip protokoll alapú stream-technológia pedig a video és audio-jelek streamelését teszi lehetővé fizikai helytől függetlenül.

A spanyol Universitat Pompeu Fabra zene-technológia csapata évek óta fejlesztett installációja egy szép példája a különböző alkotóelemek összekötésének, és egyben a vizuális programozás élményét átülteti a fizikai valóságba. A projekt a *reacTable** névre hallgat, és rengeteg szimpóziumon, kiállításon, fesztiválon szerepel, nagy sikerrel. SERGI JORDÁ koncepcióján, az *FMOL* szintetizátoron alapul a felépítés: zenei élmény, vizuális visszacsatolással, ezáltal mélyebb érzetet kapunk a hangok közötti összefüggésről.



12.-13. ábra: A reactTable* működés közben

Az egész egy asztal, amelyre a ráhelyezett tárgyak egy-egy zenei modulnak felelnek meg. A hangokkal így taktilis módon kerülünk kapcsolatba: a tárgyak közötti vonalak a hang közlekedését jelenítik meg, minden helyen a hanghullám aktuális állapota látszódik. A szűrők, generátorok, hangminták által keltett rezgések vizualizálva vannak az asztalon. Egy-egy tárgy bármikor elvehető és hozzáadható, hiszen a rendszer a már korábban említett, dinamikus módon működik, a folyamatos audio-jel szakadás nélkül jelen van a rendszer módosításakor is. A hangszintézis a már korábban említett GÜNTER GEIGER munkája, amelyet Pure Data nyelvben programozott. Minden egyes tárgynak van egy párja a programban, egy objektum, ami az adott hang műveletet végzi. A szintézis OSC-n keresztül össze van kötve a reactiVision nevű szoftverrel, amelyet külön ehhez az installációhoz fejlesztettek. Ennek a lényege, hogy a tárgyakon található kis ábrák pozícióját precízen azonosítani és követni tudja egy kamera segítségével. Így kapja meg a szintézisprogram az utasításokat a tárgytól, és ugyanígy kerül a vizualizáció (a hanghullámok) az asztalra, alulról, a kamera mellett elhelyezett projektorral.

Az asztal egyik érdekes társadalmi aspektusa, hogy képzetlen, illetve professzionális zenészek egyaránt pillanatok alatt képesek érdekes dolgokat létrehozni. Az életkor szintén nem számít, a spontaneitás, intuíció éppúgy érvényesül, mint az akusztikai, zenei, hangtani tudás. Nincsen szükség semmilyen bemeneti eszközre, kézzel lehet tologatni és helyezgetni a tárgyakat, így ahány ember az asztalhoz odafér, együtt zenélhet kedvére. Koncerteken általában az alkotók zenélnek, amikor installációként szerepel az asztal, akkor a látogatók közül bárki kipróbálhatja az eszközt. Ebben az esetben a komponálás, az előadás és a műélvezés közötti határ is felbomlik és kitágul egy közös játék-élménnyé.

Az eszköz művészeti célok mellett hatékony szerepet tölthet be az oktatásban is, a hangok szerkezete, relációja, típusai és a kortárs elektronikus zene alapvető összetevői könnyűszerrel megismerhetőek általa.

A vizuális programozás hasonló szerepet tölthet be az oktatási intézményekben, amennyiben a „dataflow” koncepciót kiterjesztik a hang-, kép-, kontrol-jelekről általánosabb témákra. Érdekes lehet dinamikus rendszerekkel érzékelteni biológiai, fizikai, matematikai folyamatokat, az „építsd fel, győződj meg” szemléletében.

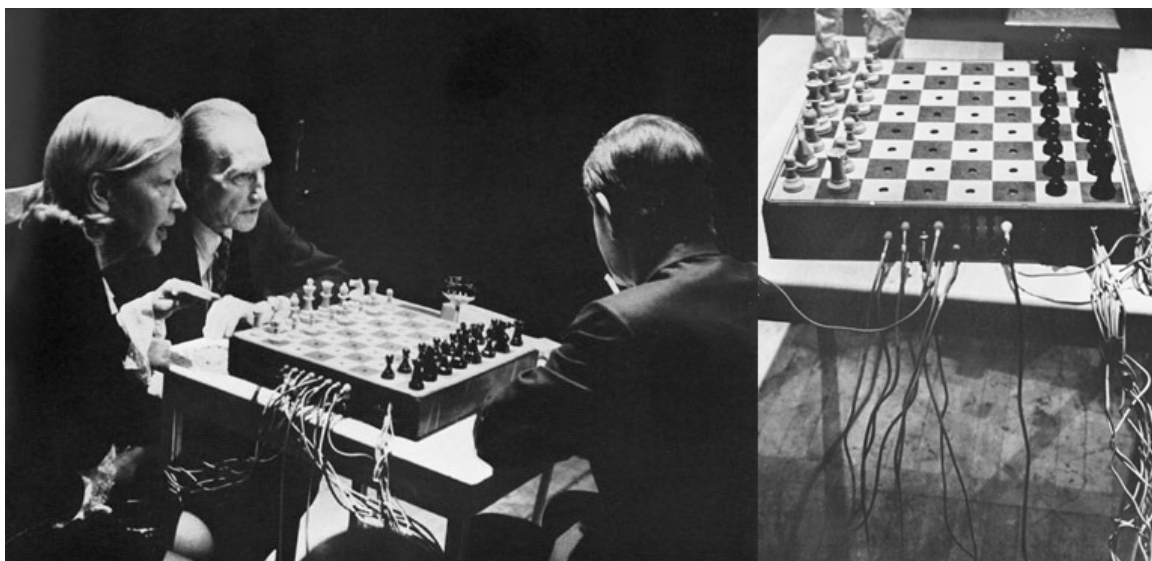
Társas lehetőség

A játék élménye összehozza az embereket. A személyes részvétel és a közvetlen visszacsatolások segítik kizökkenteni a résztvevőt a megszokott állapotból, ezáltal egy nem szokványos, kitüntetett állapotba kerül. A művészi alkotások szintén ezt a célt hivatottak szolgálni, akár pozitív, akár negatív értelemben. A játék és a művészet, a játék és az élet különböző kultúrákban más hangúlyal bír. MÉRŐ LÁSZLÓ szavaival:

A go a távol-keleti go-játékosok számára sokkal inkább kifejezi az életet, mint a sakk az európaiaknak: kevésbé érzik absztraktnak. A go a keleti gondolkodás terméke a világ egységét fejezi ki. [...] A japánok kultúrájuk szerves részének tekintik a gót. Úgy érzik, hogy alapvető értékrendszerüket, esztétikai elveiket, gondolkodásmódjukat fejezi ki.

A játék alatt megváltozik az idő. A pszichológiai folyamatok a lépések köré szerveződnek, a kapcsolat a játékosok között egy, a hétköznapiétól eltérő szintre lép. Akár a csoportos zenélés esetében, a zenekar tagjai a megszokott, verbális kapcsolattól független, másik szinten kommunikálnak egymással. A táblás játék által kiváltott élmény és a zenei játék kollektív jellege hasonló, ám a temporális észlelés mégis más, hiszen egy táblás játék, (például malom, vagy go) egymást követő menetekből áll, amelyek eredményeinek összessége adja a végső szerkezetet, a makro-szintű lépéseket. A verseny az, ami motiválja a szekvenciát. Egy zenemű esetében egymás segítése, kiegészítése a fő motiváció, ebből adódóan más a dramaturgiája.

Egy (vagy több) táblás játék és egy hanginstalláció kombinálása előidézheti ezeket a hibrid érzeteket. Másrészt összehozza a játékosokat, akik a helytől függően más közegből kerülhetnek ki. Egy kávézóban, egy galériában, egy parkban, vagy egy közterületen, vagy a hálózaton keresztül, egyidejűleg két különböző országban egészen más beállítottságú emberek találkozhatnak egymással.



14.-15. ábra: Reunion

A hangok a közös játék során fejlődnek, mintegy melléktermékként. Egy koncert-szituációban a játék is lehet a hangok mellékterméke, bár ez kevésbé valószínű. Már volt példa arra, hogy hangokkal ötvözzenek táblás játékot, 1968. március ötödikén JOHN CAGE és MARCEL DUCHAMP (feleségével) közös sakk-koncertet adott a közönségnek, Torontóban, *Reunion* néven (később még készítettek egy video sakkot is, ennek alapján). A DAVID TUDOR, GORDON MUMMA és DAVID BEHRMAN közreműködésével adott koncerten egy bedrótolt sakktáblát építettek, ami a lépéseknek megfelelően hangokat szintetizált. A „koncert” nagyon sokáig tartott, a közönség lassan elszivárgott.

Egy egyszerű, gyors lefutású játék esetében viszont a résztvevők gyakrabban cserélődhetnek, lazábban viszonyulhatnak a játékhoz. Alkalmas lehet egy malom, vagy annak térbeli változata. Ez utóbbi lényege, hogy egy 4x4x4-es táblán (rudakon) játsszák a résztvevők. Felváltva helyezik bábuikat egy-egy oszlopra, amely így lentől felfelé telítődik be. Az nyer, aki előbb kirakja egy sorba a bábuait. Akár vízszintesen, akár függőlegesen, akár átlósan.

Amennyiben ezt az egységet zenei szekvenszerként alkalmazzuk, a játékosok kezdetben feltehetően a játékbeli lépéseikre koncentrálnának, és a hangok mintegy melléktermékként kapcsolódnának a tevékenységükhöz. Elképzelhető, hogy amennyiben statikusak a játékhoz rendelt hangok, egy idő után a hangok konstellációja, hangulata befolyásolná a játék menetét, visszacsatolásként működhetne. Ugyanakkor akár lehetne egy olyan hangi tartalom, amely menetről menetre, az előzőek állapotától függően alakul, ezáltal makro-temporális szinten is egységet alkothatna a kompozíció (a játék hatására).

A mikrotemporális (egy-egy meneten belüli) összetevők alapesetben úgy épülhetnének fel, hogy az oszlopokon elhelyezett bábuk, az elhelyezés sorrendjében, tehát alulról felfelé befolyásolnák a felettük elhelyezett bábu által megszólaltatott hangot. A legalsó a tiszta szintézis szintje, az első réteg egy egyszerű frekvenciamoduláció, a harmadik réteg egy ritmikai szervezés, és a negyedik egység pedig szűrés lehetne. Minden egyes oszlopon ez a módszer. Az oszlopok elhelyezkedése pedig az alaphangok frekvenciáját és a ritmikai tagolások tempóját határozhatná meg. Például a külső sorok mélyebbek és lassabbak, a belsők magasabbak, és gyorsabbak.

Technikailag a legkézenfekvőbb rendszer LED-ek és érzékelők alkalmazása lenne. Az oszlopokra így nem kell ráhelyezni a gyűrűket (bábukat), hanem az oszlop tetején elhelyezett érzékelő aktivizálja az éppen aktuális LED-et. Minden oszlop minden szintjén két különböző színű LED található, így meg lehet különböztetni a játékosokat, a színek alapján. Aki lép, annak a színe villan fel a válaszott helyen. A tábla a LED-ekkel és az érzékelőkkel együtt össze van kötve egy számítógéppel, egy analóg/digitális feldolgozó eszköz segítségével, ami például lehet pár Arduino¹⁴. Ez az eszköz néhány analóg és digitális ki/be menetet képes kezelni. A jeleket a számítógépen egy dinamikus rendszer feldolgozza (például Pure Data), és végzi a hangszintézist, valamint a ledeket kezeli. Új játék esetén egyszerre két oszlop érintőjét kell megérinteni, és a rendszer újraindul.

A művészeti- és média-projektek, játékok, installációk tervezéséhez hasznos segítséget nyújthatnak a vizuális nyelvek. Mindemellett a fizikai eszközökkel való kapcsolat és a prototípus-készítés átláthatóbbá tehet biológiai, kémiai, fizikai kísérleteket, és más, a természettudományokhoz kötődő tevékenységeket.

A *reactTable** működési elvének mintájára elképzelhető az oktatásban való alkalmazásuk, mint konceptuális folyamatok fizikai tárgyakkal való érzékeltetése, modellezése, de ami talán a legfontosabb: mindenki számára nyitott az empirikus kísérletezgetés élményének lehetősége.

Jegyzetek

- ¹ MILLER S. PUCKETTE jelenleg a komputációs és művészeti kutatás központjának a vezetője (Center for Research in Computing and the Arts).
- ² Digital Signal Processing, digitális jelfeldolgozás.
- ³ ROBERT MOOG az elektromos zene úttörője volt, a legendás Moog szintetizátor feltalálója.
- ⁴ A kompilerek egy-egy kód lefordítását végzik, ezáltal futtatható, önálló applikációkat hoznak létre.
- ⁵ MARINE MERSENNE (1588-1646) teoretikus és zeneszerző volt.
- ⁶ Institut de Recherche et Coordination Acoustique/Musique, a párizsi zenei és akusztikai kutatóintézet központja.
- ⁷ Music Instrument Digital Interface, a nyolcvanas évektől elterjedt szabvány elektronikus zenei eszközök szinkronizálására és vezérlésére.
- ⁸ Netochka Nezvanova „szoftverprogramozó, radikális művész és online bajkeverő” (dir.salon.com/story/tech/feature/2002/03/01/netochka/index.html alapján).
- ⁹ A Linux egy nyílt forráskódú operációs rendszer, eredetileg LINUS THORVALDS fejlesztése.
- ¹⁰ OpenGL: Open Graphic Library (nyílt grafikus könyvtár), három (és két) dimenziós grafikai megjelenítésre alkalmazott specifikáció. Eredetileg a Silicon graphics fejlesztése.
- ¹¹ DIY: Do It Yourself (csináld magad). Eredetileg a nyolcvanas évek punk-mozgalmának mottója. A hangszereket zenei tudás nélkül alkalmazták, a meglévő ruhákat átalakították, használati tárgyakból piercingeket és díszeket készítettek.
- ¹² Low-tech: a HI-FI (high fidelity, magas minőségű) rendszerekkel ellentétben régi, elavult, rossz minőségű, olcsó eszközöket jelent.
- ¹³ Az entrópia egy rendszer rendezettség fokát jellemzi.
- ¹⁴ Az Arduinóról bővebb leírás és dokumentáció található a <http://www.arduino.cc> oldalon

A dolgozatban tárgyalt, vizuális programok

GEM (Graphic Environment for Multimedia): <http://gem.iem.at>

Jitter: <http://cycling74.com/products/jitter>

MAX/MSP: <http://cycling74.com/products/maxmsp>

Nato: <http://en.wikipedia.org/wiki/Nato.0%2B55%2B3d>

Pd (Pure Data): <http://puredata.info>

PDP (Pure Data Packet): <http://ydegoyon.free.fr/pidip.html>

Quartz Composer: http://en.wikipedia.org/wiki/Quartz_Composer

Reaktor: http://www.nativeinstruments.de/index.php?reaktor_us&flash=8

VVVV: <http://vvvv.org/tiki-index.php>

Irodalom

- SIEGFRIED ZIELINSKI: *Deep Time of the Media Toward an Archaeology of Hearing and Seeing by Technical Means*. MIT Press, London, 2006.
- MÉRŐ LÁSZLÓ: *Észjárások*. Budapest, 1989.
- RANDALL PACKER & KEN JORDAN (ed.): *Multimedia from Wagner to Virtual Reality*. W. W. Norton, New York, 2002.
- WILLIAM BURROUGHS: *The Future of the Novel*, 1964. In: *Multimedia*, NY, 2002.
- PIERRE LÉVY: „The Art of Architecture of Cyberspace,” *Collective Intelligence*, 1994. In: *Multimedia*, NY, 2002.
- NORBERT WIENER: „Cybernetics in History,” *The Human Use of Human Beings*, 1954. In: *Multimedia*, NY, 2002.
- SKENDEROVIC LIDIJA & KOVÁCS ZOLTÁN: *Nextlab Freestyle kutatócsoport a kortárs művészet mezőnyében*. In: *Symposion*, 0047-0049, 2006.
- SERGI JORDÁ: *Interactive Music Systems for everyone: Exploring visual feedback as a way for creating more intuitive, efficient and learnable instruments*. Stockholm Music Acoustics Conference, 2003.
- MARTIN KALTENBURGER, GÜNTER GEIGER, SERGI JORDÁ: *Dynamic Patches for Live Musical Performance*. New Interfaces for Musical Expression Conference, Hamamatsu, 2004.
- WINFRIED RITSCH: *Does Pure Data Dream of Electronic Violins?* In: *Bangbook* (<http://pd-graz.mur.at/label/book01>)
- RAMIRO COSENTINO: *Audio & Video Multi-Source Mixing and Streaming: Hack the Media*. In: *Bangbook* (<http://pd-graz.mur.at/label/book01>)
- THOMAS MUSIL & HARALD A. WILTSCHKE: *I'm the Operator with Pocket Calculator*. In: *Bangbook* (<http://pd-graz.mur.at/label/book01>)
- GÜNTER GEIGER: *The Search for Usability and Flexibility in Computer Music Systems*. In: *Bangbook* (<http://pd-graz.mur.at/label/book01>)
- USMAN HAQUE & SOMLAI FISCHER ÁDÁM: *Low Tech Sensors and Actuators* (<http://lowtech.propositions.org.uk/>)
- B.A. MYERS: *Taxonomies of Visual Programming and Program Visualization*. In: *journal jvlc*, 1990.
- Media Art Net – <http://www.medienkunstnetz.de>
- Wikipedia – online enciklopédia. <http://www.wikipedia.org>