Gyimóthy Tibor – Schrettner Lajos – Gingl Zoltán – Kincses Zoltán

Szoftverfejlesztés és műszaki informatika

Szoftverfejlesztés

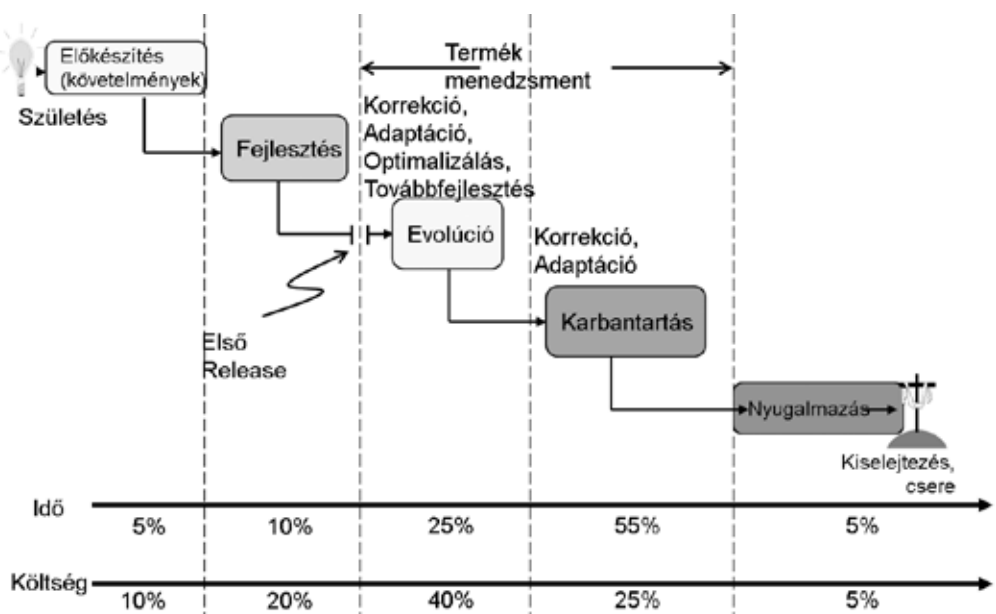
A számítógépek, a számítástechnika rendkívül látványos fejlődésének lehettünk tanúi az elmúlt évtizedekben. A XX. század második felében létrehozott elektronikus számítógép megjelenését követően olyan eszközök és technológiák kerültek kifejlesztésre, amelyek alapvetően befolyásolták, sőt megváltoztatták a modern társadalmak életét. Ennek a mélyreható változásnak az egyik, könnyebben észrevehető aspektusa a számítógépek alkatrészeinek, a hardvernek a fejlődése, amelyet a miniatürizálás egyre magasabb foka és a teljesítmény folyamatos növekedése jellemez. A másik aspektus a szoftver, mely legalább annyira fontos, de nehezebben érhető tetten, hiszen jelenléte nem annyira nyilvánvaló. A kezdeti hatalmas méretű, csak képzett személyzet által működtethető számítógépeket először felváltották a személyi számítógépek, majd az egyre kisebb méretű számítástechnikai eszközök. A méretcsökkenést az internet kialakulása és elterjedése is elősegítette, napjainkban mindennapi használati tárgyaink nagy részét el sem tudnánk képzelni hálózati funkciók nélkül. A számítástechnikai eszközök működését vezérlő szoftverek fejlesztésének módszertana, technológiája együtt fejlődött a hardvereszközökkel. Kezdetben a hardver értéke határozta meg az eszközök piaci értékét, de ez az arány mára sok esetben megfordult, vagyis a működést vezérlő szoftverek az értékesebbek. Mindez annak köszönhető, hogy a felhasználók által elvárt és megszokott színvonalú funkciók megvalósításához rendkívül összetett szoftverre van szükség. A szoftverfejlesztés mára egy nagyon szerteágazó, világgazdasági szempontból is meghatározó iparággá fejlődött. A szoftverfejlesztéssel foglalkozó kutatók és szakemberek célja, hogy ezt a diszciplínát a többi mérnöki tudományhoz hasonlóan olyan szintre emelje, amely lehetővé teszi, hogy szabványosított módszerek és módszertanok alkalmazásával megbízható, a felhasználói igényeknek minél jobban megfelelő termékeket állíthasson elő.

A szoftverfejlesztési folyamat

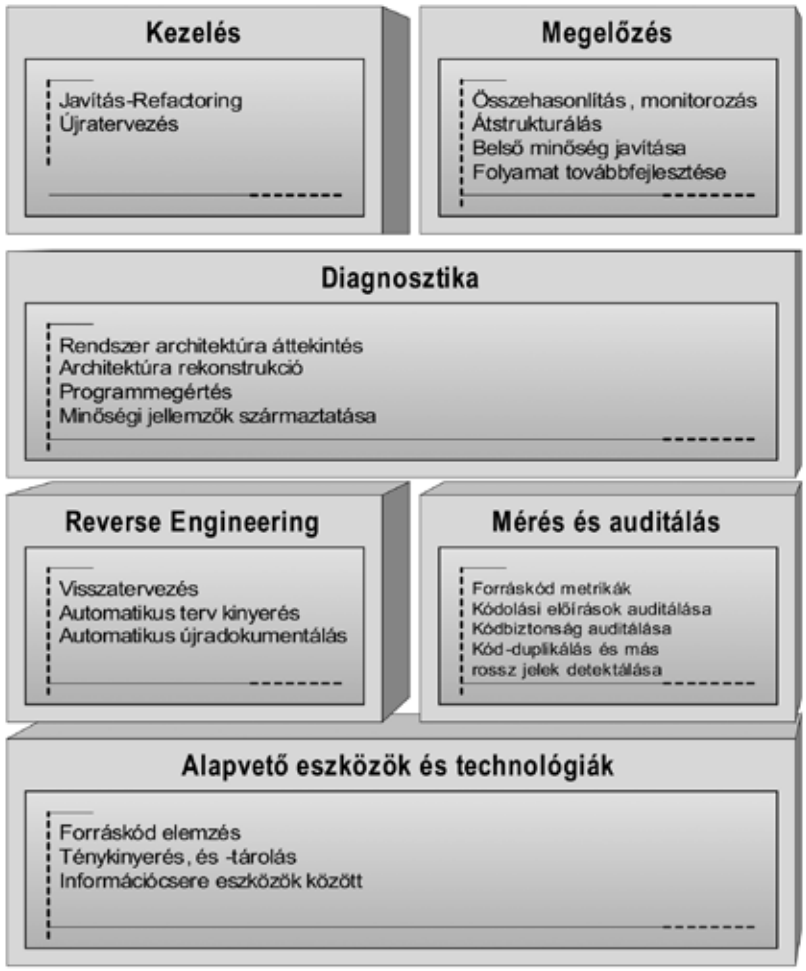
A szoftverfejlesztés széles körben elterjedt, ún. agilis módszertana szerint a fejlesztés egymást követő rövid fejlesztési ciklusokra tagolódik, amelyek

során a szoftverrendszer fokozatosan nyeri el végső formáját. A rövid fejlesztési ciklusok, a gyors változtatási igények és gyorsan változó környezet sok szempontból megváltoztatták a fejlesztési gyakorlatokat, ezért az egyes fejlesztési fázisokban végzendő tevékenységek új megközelítéseket követelnek meg. A szoftverfejlesztési folyamat fő fázisai:

- Követelménymenedzsment: a készítendő szoftverrel szemben támasztott követelmények meghatározása és dokumentálása, a fejlesztés alapját képező specifikáció elkészítése.
- Tervezés: a szoftver architektúrájának, komponenseinek és a komponensek közötti kapcsolatoknak a megtervezése. A legelterjedtebbek az objektumorientált, modell alapú tervezési módszerek.
- Implementáció: A szoftver forráskódjának előállítása. A fejlesztők számos integrált programfejlesztő rendszer közül választhatnak.
- Validáció: Annak ellenőrzése, hogy az elkészített szoftver megfelel-e a specifikációnak. Ebben a fázisban elsősorban nyomkövetés és tesztelés történik, amelyek együtt a teljes fejlesztési költség jelentős hányadát tehetik ki. Korábban a fejlesztés többi fázisánál kevesebb figyelmet kapott a tesztelés, a jelenlegi trendek szerint a tesztelési folyamatok kutatása, és megfelelő, szabványosított elvek szerinti alkalmazása egyre nagyobb hangsúlyt kap.



1. ábra. Tipikus ráfordítások a szoftverfejlesztés során



2. ábra. Forráskódalapú minőségbiztosítás

Szoftverevolúció, minőségbiztosítás

A szoftverfejlesztési fázisokon, illetve az egyes fejlesztési módszertanokon túl beszélhetünk szoftverevolúcióról, amely a kifejlesztett termékkel kapcsolatos hosszú távú továbbfejlesztési és karbantartási tevékenységek összefoglaló neve. Az evolúció során kerülnek előtérbe azok a tevékenységek, amelyek a minőség szinten tartására, a minőségromlás megakadályozására hivatottak. Az 1. ábrán az evolúciót középpontba állítva láthatjuk egy tipikus fejlesztési projekt esetén az egyes kapcsolódó tevékenységekre fordított időt és költséget.

A szoftver minőségének folyamatos mérése lehetővé teszi a hibák korai felfedezését, ezzel együtt a költségek alacsonyan tartását. A 2. ábrán csoportosítva láthatjuk a minőség javítására irányuló tevékenységeket, ezek közül az alábbiakat emelhetjük ki:

- Minőségmérés és monitorozás, amely mögött a szoftvermetrikák és minőség modellek állnak. Mindez monitorozó eszközök bevezetésével valósul meg a gyakorlatban, ezek állhatnak különálló komponensekből (architekturális vizsgálatok, kódolási szabálysértésellenőrző, klóndetektáló, ellenminta-detektáló, alacsony-, illetve magas szintű metrikák kimutatására alkalmas eszközök), vagy végezhető integrált megoldások segítségével is.
- Refaktoring, amelynek során a program szerkezetén úgy hajtanak végre változtatásokat, hogy az jobb minőségű kódot eredményezzen, miközben a funkcionalitást ne befolyásolja. A nem megfelelő minőségű programkód javítása során ezzel a módszerrel biztosítható, hogy egyes hibák javítása ne okozza új hibák megjelenését a rendszerben.

A fejlesztési folyamat csak akkor eredményez jó minőségű szoftvervégterméket, ha a minőségre a teljes folyamat során hangsúlyt fektetünk. A szoftverminőséggel, illetve az ezt megalapozó technológiákkal kapcsolatos kutatások több évtizedes múltra tekintenek vissza. A minőségbiztosítás alapja a minőség mérése, amelyhez a szoftverrendszerek elemzésére, belső szerkezetük részleteinek feltárására van szükség. A programok belső szerkezetének reprezentálására előszeretettel alkalmaznak gráfokat, ezekben a csúcsok a programok egyes részegységeit, míg a csúcsok között futó élek a részegységek között fennálló valamilyen kapcsolatot jelentenek.

A szoftverfejlesztés tanszéki kutatási témái több ponton is kapcsolódnak a fentebb ismertetett fejlesztési fázisokhoz, főleg a szoftver kezdeti fejlesztésének lezárása utáni evolúciós szakaszra koncentrálva.

Forráskód elemzésen alapuló módszerek

A fejlesztési folyamat validációs fázisában végzett nyomkövetés és hibajavítás hatékony támogatására alkalmasak a különböző *programszeletelés*i módszerek, melyek kutatásában a tanszék munkatársai is jelentős érdemeket szereztek [1,2]. A szeletelési technika felhasználásával nagyméretű programok kezelése is lehetővé válik azáltal, hogy ki lehet jelölni a program egy adott probléma szempontjából releváns részét, így a hibák megtalálása és javítása is sokkal könnyebbé válik.

Az utóbbi évtizedekben az objektumorientált nyelvek (például C++, Java, C#) lettek a legelterjedtebb programozási nyelvek. Komoly nemzetközi elismerést vívott ki a tanszék kutatói által kidolgozott *Columbus technológia* [3], amely az objektumorientált programozási nyelven írt programok evolúciója során hasznosítható. Az eredményeket ismertető tudományos cikk 10 évvel a publikálása után, az ICSM2012 konferencián (ez a szoftverkarbantartás vezető nemzetközi konferenciasorozata) megkapta a legnagyobb hatású közleménynek járó nívós díjat. A Columbus technológia hatékony módszert ad nagyméretű programok gráfalapú reprezentációjának létrehozására és feldolgozására. Ennek azért van nagy jelentősége, mert a szoftveriparban a programok méretének – és ezzel együtt komplexitásának – növekedése komoly nehézségek elé állítja a fejlesztőket. Csak úgy lehetnek képesek megfelelő minőségű terméket előállítani, ha a fejlesztés alatt álló rendszert átlátják, a komponenseket egyértelműen azonosítani tudják, a közöttük lévő kapcsolatokat pedig ismerik és a kívánt cél érdekében fel tudják használni, vagy tervezett módon át tudják alakítani.

A Columbus technológia képezi az alapját sok olyan megoldásnak, amely a szoftverfejlesztés evolúciós fázisában alkalmazható a minőség javítására. Ezek között első helyen említhetjük a szoftver metrikák meghatározását, vagyis a szoftverek mennyiségi és minőségi jellemzőinek mérését. E jellemzők mérése fontos információkat szolgáltat a programok állapotáról. Ennek köszönhetően definiálásuk, meghatározásuk, illetve e *metrikák validációja* [4] aktív kutatási terület maradt napjainkig is.

A szoftvermetrikák fontos kiindulási alapot jelentenek a minőséggel kapcsolatos vizsgálatokban, alkalmasak következtetések levonására, ugyanakkor önmagukban csak meglehetősen alacsony szinten jellemzik a programokat. Ebből a felismerésből kiindulva indultak meg a kutatások olyan magasabb szintű, aggregált metrikák irányába, amelyekkel a szoftver minőségét magasabb szinten lehet számszerűen jellemezni. Ezeknek a magasabb szintű metrikáknak egy jól körülhatárolt célból definiált koherens rendszerét *minőségi modell*nek nevezik, melyek kutatása a Columbus technológiára alapozva szintén a tanszék kutatóinak érdeklődési körébe esik [5]. Ennek során megalkottak egy új irányba mutató valószínűségi minőségi modellt, amely nem egyetlen számértékkel próbálja jellemezni a minőséget, hanem egy valószínűségi függvénnyel. Ez a megközelítés lehetővé teszi, hogy a minőség jellemzése során olyan aspektusokat is figyelembe lehessen venni, amelyeket más modellek nem tudnak kezelni, mint például a szakértői vélemények.

A szoftverrendszerek minőségének mérése mellett – de attól nyilvánvalóan nem elválaszthatóan – komoly erőfeszítések történnek a minőség javítását

célzó módszerek kidolgozására is. Ezek között említendő az ún. *hatásanalízis*, melynek célja a programokban végrehajtott módosítások hatásának körülhatárolása abból a célból, hogy a módosítás utáni helyes működés ellenőrizhető legyen. A hatásanalízissel kapcsolatos kezdeti kutatások a korábban említett szeletelés módszerből indultak ki, azonban ez a megközelítés valós programrendszereken túlságosan erőforrás-igényesnek bizonyult. Emiatt szükség volt olyan függőségi relációk definiálására, amelyek a programban fellelhető szerkezeti függőségeknek csak egy – a feladat szempontjából releváns – részét veszik figyelembe. Az egyszerűsítés hatékonyságnövelő hatásával párhuzamosan elkerülhetetlenül csökken a pontosság is, lehetséges azonban olyan megoldást találni, amely elfogadható kompromisszumot jelent. A tanszék kutatói is aktívak ezekben a kutatásokban, ennek keretében kidolgozták a Static Execute After (SEA) relációt, és kimutatták, hogy hatékonyan alkalmazható automatikus hatásanalízis elvégzésére [6].

A minőség javításával kapcsolatban egyre nagyobb figyelmet kap a tesztelési folyamat, melynek során szisztematikusan ellenőrzik, hogy a szoftverrendszer funkciói az elvárásoknak megfelelően működnek-e. A tesztelés legfontosabb céljai közé tartozik, hogy kimutassa a rendszerben esetleg meglévő hibák létezését, illetve – ha lehetséges – okát. A tanszék kutatásai ezen a területen arra összpontosulnak, hogy hogyan lehet minél kevesebb számú teszt végrehajtásával minél jobb hibadetektálási és hibalokalizálási eredményt elérni [7].

Beágyazott rendszerek, nyílt forráskódú fejlesztések

Napjaink meghatározó trendje a mobil számítástechnikai eszközök terjedése, amelyekhez a felhasználóknak minőségileg új szolgáltatásokat nyújtó alkalmazások és szolgáltatások is tartoznak. A mobil eszközök informatikai szempontból kihívást és új lehetőségeket egyaránt jelentenek. A beágyazott rendszereket célzó, illetve a mobil alkalmazások fejlesztési folyamata eltér a hagyományostól, melynek során figyelembe kell venni a céleszközök korlátozott erőforrásait, az újszerű felhasználói felületeket, felhasználási módokat. Ehhez kapcsolódva a tanszék kutatási területeihez tartoznak a beágyazott rendszerek korlátozott erőforrásainak minél optimálisabb kihasználását célzó fejlesztések, mint például olyan programkód-tömörítő eljárás megvalósítása, amellyel a korábbiakhoz képest közel 20%-os helymegtakarítást is el lehet érni [8]. Hasonlóan eredményes volt a UBIFS flash fájlrendszer kifejlesztése, amelynek eredményeképpen a korábbiakhoz képest 90% feletti művelet-megtakarítás vált lehetővé [9]. E fejlesztések eredményei bekerültek a legelterjedtebb nyílt

forráskódú operációs rendszerbe, a Linuxba is. A tanszék aktív tevékenységet fejt ki egyéb erőforrás-optimalizálást célzó kutatásban [11], illetve nyílt forrású szoftverekkel kapcsolatos fejlesztésekben is. Több projekt kapcsolódik például a WebKit böngészőmotorhoz, vagy a GCC fordítóprogramot is tartalmazó fejlesztői eszközkészlethez [10].

Mobil hálózatok

A számítógéphálózatok megjelenése és az internet kialakulása az informatikai eszközök fejlődésének további irányait is jelentősen befolyásolta. Az egymással kommunikáló számítógépek által nyújtott lehetőségek olyan funkciókat megvalósító alkalmazások kifejlesztése előtt nyitotta meg az utat, amely a korábbi technológiákkal nem volt lehetséges. A felhasználói igények kielégítése érdekében az informatikai kutatások részben új problématerületek felé fordultak, mint például a hálózati kommunikáció biztonsági kérdései, vagy a hálózati forgalom által generált nagy mennyiségű adat tárolása, gyors visszakereshetősége és feldolgozása.

A mobil eszközökkel és a hálózatokkal kapcsolatos kutatásokba a Szoftverfejlesztés Tanszék munkatársai is bekapcsolódtak. Ennek keretében vizsgálták a P2P (Peer-To-Peer) hálózatok működését. Ezek olyan hálózatok, amelyek egyenrangú komponensekből állnak, a döntések meghozatala elosztott módon történik, nincs olyan központi elem, amelynek esetleges kiesése zavarokat okozhatna a hálózat működésében. A P2P hálózatok előnyei sajnos rossz célokra is felhasználhatók, így ez a technológia felkeltette a kártékony szoftverek (malware) készítőinek figyelmét is. A P2P hálózatokban üzemelő kártékony programok felderítésének ez idáig ismert leghatékonyabb módja a hálózati forgalomban megfigyelhető mintázatok azonosítása. Ez a technika azért alkalmazható, mert a felderíteni kívánt programok természetüknél fogva jelentős adatforgalmat generálnak. Fontos kérdés, hogy az ilyen jellegű kártékony programok mekkora potenciális veszélyt jelentnek a jövőben, van-e olyan módszer, aminek felhasználásával felderítésük nehezebbé válik. A tanszék kutatói által elért eredmények alapján azt állapíthatjuk meg, hogy a P2P hálózatokban működő programok működése szervezhető úgy, hogy az egyes komponensek csak kevés számú más komponenssel kommunikáljanak [12]. Ez sajnos azt jelenti, hogy a rosszindulatú programok újabb generációja elleni védekezés még nagyobb kihívás elé állíthatja az informatikai biztonsággal foglalkozó szakembereket.

Ahogy korábban említettük, az informatika ma már egyre inkább életünk szerves részévé kezd válni, napjaink minden időszakában számítógépekkel vagyunk körülvéve otthonunkban, munkahelyünkön, utazás vagy éppen szórakozás közben. Ennek a jelenségnek a leírására született meg a ubiquitous („mindenütt jelenlévő") computing kifejezés, melynek szerves részét képezi az a törekvés is, hogy a felhasználókat minél inkább bevonja a rendszerek vezérlésébe anélkül, hogy informatikai szakemberekké kellene válniuk. Ez a törekvés több tudományterületet is érint, kihívást jelent az egyes részterületeken meglévő eredmények adoptálása, integrálása, illetve a legmodernebb eszközök és technológiák képességeinek minél teljesebb kihasználása [13].

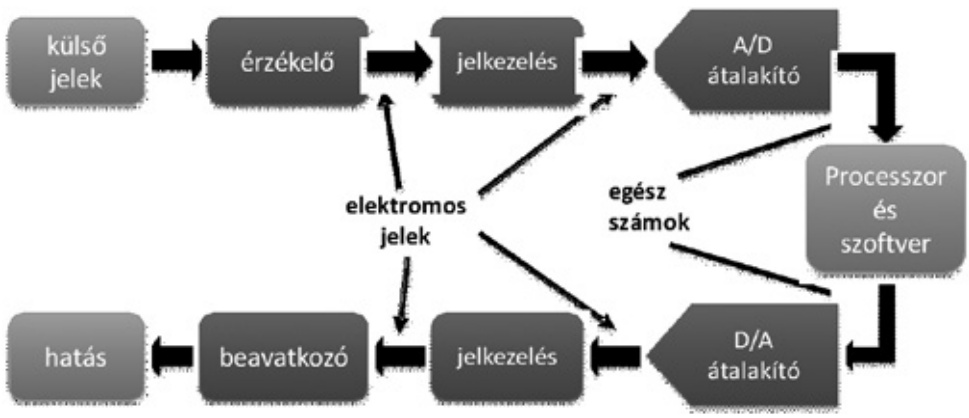
A tanszéken folyó munka eredményességét jelzi az 50 feletti sikeresen befejezett nemzetközi kutatás-fejlesztési projekt, valamint a több mint 200 referált tudományos közlemény. Hatot ezek közül legjobb cikk díjjal ismertek el rangos nemzetközi konferenciákon. Külön megemlítendő, hogy a szoftverfejlesztéssel foglalkozó jelentős nemzetközi konferenciák közül néhány a tanszék szervezésben valósult meg Szegeden, mint például az ICSM 2005, az ESEC/FSE 2011 és a CSMR 2012.

Informatikai megoldások valós rendszerekben

Valódi jelek, hardverek és szoftvereik

A számítógépek hardverei régebben leginkább csak a szoftverek futását, az adatok tárolását, továbbítását és a felhasználóval való kapcsolattartást biztosították. Az elektronika fejlődése és a rendkívüli mértékű miniatürizálás ma már ennél sokkal többre képes: egy komplett számítógép akár lehet milliméteres nagyságú, mérhet és előállíthat valódi jeleket is. Az emberiség régóta készít eszközöket és gépeket saját munkájának megkönnyítésére, ma már rengeteg ilyet találunk a környezetünkben a mosógéptől a telefonokig, a ventillátortól, a légkondicionálótól, az autótól a repülőgépekig. Az ilyen eszközöknek, gépeknek a környezetüket érzékelniük kell, az információt fel kell dolgozniuk, majd szükség esetén hatást kell gyakorolniuk. Így működik a fűtésszabályozás is: a hőmérséklet mérésének eredményétől függően bekapcsol vagy kikapcsol a fűtés. Ezek a megoldások lényegében olyanok, mint az élőlények esetében is: a külvilág jeleit érzékelik, ezek a jelek átalakulnak feldolgozható formába, majd a feldolgozás eredményeként beavatkozás

történik. Ha fázunk, pont azt tesszük, mint egy fűtési rendszer: az alacsony hőmérséklet hidegérzetet vált ki, ez az agyba jut, ami parancsot ad a mozgásra, azaz hő fejlesztésére. Azt is egyszerűen láthatjuk, hogy a leghatékonyabbá akkor tehetünk egy eszközt, ha az információ feldolgozását a lehető legjobbra választjuk – erre pedig ma a számítógép és a szoftver képes. A kérdés tehát az, hogyan juthatnak a valódi jelek egy számítógépbe és hogyan kelthet a számítógép különböző hatásokat?



3. ábra. A megoldás egyszerűen látszik a fenti ábrán

A külső jelek – például hőmérséklet, elmozdulás, fényintenzitás – érzékelők segítségével elektromos jelekké – például feszültséggé, árammá, ellenállássá – alakítható érzékelők vagy más néven szenzorok segítségével. Az elektronikában az információt megfelelő nagyságú feszültség hordozza, így szükséges még egy jelkezelés is. Ezen a ponton számokká alakíthatjuk a feszültséget (analóg-digitál átalakítás) és processzorokra és szoftverekre bízhatjuk a feldolgozást. A folyamat megfordítható, ez látszik az alsó ágban. Érzékelőre példa a mikrofon, a fotocella, a termoelem, beavatkozóra pedig a hangszóró, a LED, a motor. Egyszerűen láthatjuk, hogy a zsebünkben is ilyen eszköz, a mobiltelefon van.

A műszaki alkalmazások mellett a kutatásoknak is nagy lendületet adott ez a rendkívüli hatékonyság. A természettudományok a külvilág jeleinek mérésére és elemzésére, speciális hatások keltésére, egyedi kísérletezésre alapulnak. A Műszaki Informatika Tanszék kutatási területei is ennek megfelelően épültek ki az informatikai, villamosmérnöki, fizikai, kémiai, orvostudományi diszciplínák összekapcsolása révén. Részletes információk: http://www.inf.u-szeged.hu/dti/research/.

Szoftverdefiniált műszerek és kísérletezés

A kísérletezésekhez sok esetben szükséges egyedi műszerek fejlesztését teszi lehetővé a szoftverdefiniált műszerezés, amikor a műszerfunkciók legnagyobb részét szoftverek végzik el. A tanszék kutatói számos hardvert fejlesztettek ki, melyek kellően univerzálisak, sokféle mennyiség mérését teszik lehetővé és processzorokat is tartalmaznak. A szoftverek egy része a műszerben fut, a másik pedig a vezérlő számítógépen. Pusztán a szoftver cseréjével ugyanaz a műszer más mérésekre alkalmassá tehető.

Véletlenszerű jelek elemzése és hasznosítása

A véletlenszerű jeleket – más néven zajokat – legtöbbször károsnak, az információszerzést gátlónak tartják. Ugyanakkor a véletlenszerű jelek szinte mindenhol előfordulnak: a hőmérséklet a részecskék véletlenszerű mozgásának intenzitásával arányos, az oldódás, folyadékok keveredése véletlen jelenségekre épül, és a gazdasági és társadalmi folyamatok mellett a biológiai evolúcióban is meghatározó szerepe van a véletleneknek. A véletlenszerű jelek épp ezért gyakran információforrásnak is tekinthetők, sőt, akár hasznosíthatók rendszerek jobb működtetésére is. Mindehhez igen sok számítási kapacitásra, gyakran kisméretű eszközökre van szükség, amit a mai technika már lehetővé tesz.

Számos eredményt értünk el különféle zajfajták tulajdonságainak megismerésében, különösen a mai napig rejtélyes 1/f-zaj vizsgálatával. Példákat adtunk rá, hogy bizonyos rendszerekben elég hihetetlen módon kevésbé zajos jeleket lehet kapni zaj hozzáadásával, és olyan hardvereket és szoftvereket is készítettünk, melyek zaj segítségével képesek hatékonyan titkosítani az adatátvitelt. Az Alkalmazott Kémiai Tanszék kutatóival, köztük dr. Kónya Zoltánnal és dr. Kukovecz Ákossal nemzetközi együttműködésben vettünk részt, melynek egyik eredmény szerint lehetséges gázszenzorok zajának mérésével hatékonyabb információszerzést megvalósítani. Részletes információk: http://www.noise.inf.u-szeged.hu/Research/.

Multidiszciplináris kutatások

Fizikai és műszaki kutatásaink mellett 1994 óta végzünk közös munkát az Általános Orvostudományi Kar számos munkatársával, többek közt dr. Rudas Lászlóval, dr. Zöllei Évával, dr. Pap Róberttel. Az egyik fontos terület

a szívműködés vizsgálata, amihez EKG, vérnyomás, légzés és ezek összefüggéseinek számítógép-vezérelt méréséhez és elemzéséhez fejlesztettünk ki hardvereket és szoftvereket, illetve jelfeldolgozási módszereket. Az eredmények közel 60 publikációban jelentek meg, melyek jól mutatják a multidiszciplináris együttműködés hasznát. Legújabb területként légzésmechanikai kutatásokba kapcsolódtunk be, ahol a tüdő mechanikai impedanciájának valós idejű méréséhez fejlesztünk precíz mérésekre alkalmas kísérleti rendszert és elemzési módszereket dr. Hantos Zoltán irányítása mellett. Orvostudományi kutatásaink mellett szoftverdefiniált műszereinket használják lézerfizikai, biofizikai és kémiai területeken is. Részletes információk: http://www.noise.inf.u-szeged.hu/med/.

Szoftverdefiniált műszerek az oktatás modernizálásához

Nagy szükség van ma az oktatás fejlesztésére, a diákok sokszor úgy érzik, hogy amit tanulnak, az túl elméleti, távol áll a hétköznapjaiktól. Sokat segíthet ezen is a műszaki informatika: a fentebbi ábrának megfelelően parányi USB portra köthető műszereket fejlesztettünk, melyekkel rendkívül hatékony kísérletezés végezhető el, egyszerre akár több diák is gyakorolhat. Példákat mutattunk arra, hogy számos mérésre megbízhatóan használhatjuk akár a minden gépen elérhető hangkártyát is, így a diákok akár otthon, saját ötleteik alapján is kísérletezhetnek, eredményeiket megoszthatják a közösségi oldalakon is. Részletes információk: http://www.noise.inf.u-szeged.hu/edudev/.

FPGA áramkörök alkalmazása

A nagy sebességű műveletvégzés különösen fontos, ha egy olyan eszközre van szükség, ami igen gyorsan változó valós idejű jelekkel dolgozik, és a számítások eredményére is azonnal szükség van. Ilyen például a valós idejű képfeldolgozás, amikor a kameráról érkező nagy mennyiségű képpontot kell igen gyorsan elemezni azért, hogy az eszköz gyorsan válaszolhasson, és megfelelő döntéseket hozhasson. Mindezt gyakran a lehető legkisebb méretben kell megvalósítani. Ezeknek a követelményeknek a mai processzoroknál akár több százszor nagyobb számítási sebességgel rendelkező néhány centiméter méretű úgynevezett FPGA áramkörök tökéletesen megfelelnek. Az FPGA áramkörökkel megvalósítható az elvégzendő számítási feladatok nagyfokú párhuzamosítása. A tanszéken folyó kutatások egyik iránya a különböző képfeldolgozási felada-

tok FPGA alapú megvalósítása és gyorsítása. Az egyik ilyen feladat például a képalkotó rendszerek (csillagászati vagy nagy hatótávolságú katonai távcsövek) hatékonyságának javítása a fényhullám hullámfrontját torzító hatások valós idejű kiküszöbölésével. Ennek a feladatnak a megoldásához az FPGA áramkörnek 715 Mbyte adatot kell feldolgoznia másodpercenként. Egy másik képfeldolgozási feladat olyan FPGA alapú rendszer kifejlesztése mely segítheti a látássérült emberek tájékozódását. A tanszék FPGA alapú kutatásainak másik iránya olyan rendszer kifejlesztése, mely a végtagamputált betegek művégtagvezérlésében nyújthat segítséget. Részletes információk: http://www.inf.u-szeged.hu/dti/research/fpga/

Vezeték nélküli szenzorhálózatok

Az ujjhegynyi mikroszámítógépek és különféle szenzorok mellé ma már vezeték nélküli adó-vevőket is integrálnak. Ilyen megoldás például a Bluetooth vagy a ZigBee. Ezek a kicsi, komplett áramkörök különböző jeleket – például mozgás, hőmérséklet, fény, páratartalom – képesek mérni és továbbítani azokat valamilyen vezetékek nélküli kapcsolaton keresztül, akár egymással is kommunikálva. Így az ilyen parányi eszközökből, úgynevezett vezeték nélküli szenzormodulokból (angolul mote, azaz porszem) hálózatokat is építhetünk, melyek rendkívül sok helyen alkalmazhatóak a gyógyászattól a robothelikopterek vezérléséig, az autókban a terepen vagy az épületekben való adatgyűjtésig. Mindezeket a feladatokat akár több évig is önállóan és megbízhatóan kell végezni úgy, hogy csupán egy korlátozott kapacitású elem vagy akkumulátor biztosítja az energiaellátást. Az ilyen hálózatok esetében több probléma is felmerülhet, mint például, hogy a szenzorok maguk meg tudják állapítani egymáshoz képesti pozíciójukat, akár mozgásukat, vagy az időszinkronizáció, mely során a hálózat minden egyes elemének szinkronizálni kell az „óráját". Ezen problémák mellett azonban a legfontosabb az eszközök energiahatékony működtetése. A tanszéken folyó kutatások egyik fő célja a módszerek tökéletesítése, optimalizálása különböző alkalmazási területeken, környezeti feltételek mellett. Részletes információk: http://www.inf.u-szeged.hu/dti/research/wsn/

Robotika

A műszaki informatika egyik klasszikusa a robot, melyből rendkívül sokféle létezik, és intenzíven használják is ezeket gyártósoroknál, veszélyes vagy

nehezen megközelíthető terepek felderítésénél, precíziós sebészethez és sok más helyen is. A robot ötvöz három fontos területet: a mechanikát, az elektronikát és az informatikát – szokták az ilyen rendszereket mechatronikainak is nevezni. Sokféle nagyon jól kidolgozott elv, a gyakorlatban működő robotok mellett is szükség van újabb ötletekre az optimalizálás, a hatékonyság növelése és a költségek csökkentése érdekében. Tanszékünkön a robotika fontos területe az oktatásnak, de kutatási feladatok is kapcsolódnak hozzá elsősorban a mobil robotok pályakövetési módszereinek tökéletesítése, aktuátorok átviteli függvényének meghatározása, valamint áttételek optimalizálása területén. Részletes információk: http://www.inf.u-szeged.hu/robotics/

Hivatkozások

1. T. Gyimóthy, Á. Beszédes, I. Forgács: An efficient relevant slicing method for debugging, Proceedings of the Joint 7th European Software Engineering Conference and 7th ACM SIGSOFT International Symposium on the Foundations of Software Engineering (ESEC/FSE99), pp. 303–321, 1999
2. D. Binkley, S. Danicic, T. Gyimóthy, M. Harman, Á. Kiss, B. Korel. A Formalisation of the Relationship between Forms of Program Slicing. Science of Computer Programming, Volume 62(3), pp.: 228–252, 2006.
3. R. Ferenc, Á. Beszédes, M. Tarkainen, T. Gyimóthy: Columbus – Reverse Engineering Tool and Schema for C++, 18th International Conference on Software Maintenance (ICSM), IEEE Computer Society, pp. 172–181, 2002
4. T. Gyimóthy, R. Ferenc, I. Siket: Empirical Validation of Object-Oriented Metrics on Open Source Software for Fault Prediction, IEEE Transactions on Software Engineering, Volume 31, pp. 897–910, 2005
5. T. Bakota, P. Hegedűs, P. Körtvélyesi, R. Ferenc, T. Gyimóthy: A probabilistic software quality model, 27th IEEE International Conference on Software Maintenance (ICSM), 2011
6. L. Schrettner, J. Jász, T. Gergely, Á. Beszédes, T. Gyimóthy: Impact analysis in the presence of dependence clusters using Static Execute After in WebKit, Journal of Software: Evolution and Process, 2013
7. L. Vidács, Á. Beszédes, D. Tengeri, I. Siket, T. Gyimóthy: Test suite reduction for fault detection and localization: A combined approach, IEEE Conference on Software Maintenance, Reengineering, and Reverse Engineering CSMR-WCRE, pp. 204–213, 2014

8. T. Gergely, F. Havasi, T. Gyimóthy. Binary Code Compression Based on Decision Trees, Proceedings of the Estonian Academy of Sciences – Engineering, Volume 11, pp. 269–285, 2005
9. F. Havasi: An improved B+ tree for flash file systems. SOFSEM 2011: Theory and Practice of Computer Science – 37th Conference on Current Trends in Theory and Practice of Computer Science, Novy Smokovec, Slovakia, pp. 297–307, 2011
10. Á. Beszédes, T. Gyimóthy, G. Lóki, G. Diós, F. Kovács: Using Backward Dynamic Program Slicing to Isolate Influencing Statements in GDB, GCC Developers' Summit, 2007
11. Z. Herczeg, D. Schmidt, Á. Kiss, N. Wehn, T. Gyimóthy. Energy Simulation of Embedded XScale Systems with XEEMU. Journal of Embedded Computing, Volume 3(3), pp. 209–219, 2009
12. M. Jelasity, V. Bilicki: Scalable Stealth Mode P2P Overlays of Very Small Constant Degree. In ACM Transactions on Autonomous and Adaptive Systems, Volume 6, pp. 27:1–27:20, 2011
13. V. Bilicki, Z. Rak, M. Kasza, Á. Végh, R. Béládi, T. Gyimóthy: End User Programming in Smart Home. Patterns, Programming and Everything. DOI 10.1007/978-1-4471-2350-7_3. Springer London, pp. 19–29, 2012