

Rendszertechika

A rendszertechika az *informatikai berendezések és rendszerek* felépítésének architekturális alapelveire és jelenségeire terjed ki – beleértve a hálózatokkal összekapcsolt berendezések által alkotott rendszerek és általában a rendszerekből kialakított, *magasabb szintű rendszerek* felépítési elveit is.

Az ilyen módon a *rendszerek egymásba ágyazódása* a rendszerek egy sajátos hierarchiáját határozza meg az informatika területén, ahol a legalsó szinten 1) a *processzormagok*; e fölött maguk 2) a *processzorok*; a következő szinten a különböző – speciális vagy szabványos – buszokkal összekötött processzorokból szerveződő, már önálló feladatvégzésre alkalmas 3) *számítógépek* (például munkaállomás, szerver); az ezekből szoros összekapcsolással kialakított 4) *számítógéprendszerek* (például klaszter, elosztott rendszer, szuperszámítógép); vagy lazább, rugalmasabb összeköttetéssel szervezett, de szorosan menedzselt 5) *számítóközpontok* (például szerverfarm); végül mindezeknek az egységes internetprotokollra épülő hiperrendszere, azaz maga 6) a *világháló*. Így nyernek értelmet azok a divatos kifejezések, mint a „hálózat maga a számítógép” vagy „a hálózat világméretű számítógéppé alakul”: a világháló valóban és valójában egy rendkívül összetett, számítógépekből, számítógéprendszerekből és számítóközpontokból összeálló informatikai rendszer.

E rendszerhierarchia minden szintjén *hasonló elvek* érvényesülnek, de ezeknek merőben *különbözőek a megvalósulásaik*, és éppen ez adja az egyes szintek jellegzetességeit. Rendszertechikai szempontból a leginkább meghatározó elv a *párhuzamosság* elve, amely több, hasonló, de egymástól viszonylagosan független rendszer elem egyidejű, ugyanakkor valamilyen mértékben szinkronizált működését, illetve együttműködését fejezi ki. A párhuzamosság alkalmazása egy rendszerben a feladatok és a terhelés megosztását, a részfeladatok egyidejű végrehajtását, és így végső soron *teljesítménynövelést* tesz lehetővé.

Egy másik fontos elv, hogy a rendszerarchitektúrákban az elemek közötti koordinációt és szinkronizációt egyre inkább *szoftver* végzi, amivel rugalmasabbá, skálázhatóbbá és menedzselhetőbbé lehet tenni a rendszerek működését. Ez ma már gyakran eléri azt a mértéket, hogy a rendszer kifelé mutatott viselkedése, külső kapcsolatai és *funkciói függetlenné válnak* a rendszer belső architektúrájának fizikai (hardveres) felépítésétől, és a kettő közötti kapcsolatot valamilyen szoftver (*köztesszoftver*) képezi le. Ez a folyamat a rendszerek *virtualizálódásaként* nevezhető meg és írható le.

Lényeges tendencia, hogy a párhuzamosság az informatikai rendszerek tervezésekor alkalmazott vezérelvvé válik, amely azok minden szintjén megmutatkozik. Kihashnálásához azonban az szükséges, hogy az adott rendszer mindenkor feladatát könnyen és lehetőleg automatikusan le lehessen bontani részfeladatokra, illetve ezek végrehajtását koordinálni lehessen. Ez ismét csak a rendszer- vagy köztesszoftver feladata, azonban nincs rá általános algoritmus – csak egy-egy feladatosztályra létezik valamilyen megoldás. Ilyen megoldások keresésével foglalkozik a *párhuzamos programozás* tevékenységköre, amely – érthető módon – egyre kiemeltebb, egyes szinteken már-már kritikus szerepet kap a rendszertechikában.

Utolsóként kerül említésre egy másik fontos rendszertechikai tendencia, a *miniatürizálódás*, amelynek eredményeként a különböző hierarchiaszinteken egyre

kisebb fizikai helyen egyre több rendszerelem fér el, így egyre több funkció nyújtható, és egyre nagyobb teljesítmény érhető el. Azzal, hogy a szoftver mérete nincs közvetlen kapcsolatban az általa nyújtott funkciókkal, a miniatürizálásban a szoftvernek szintén fontos szerepe jut. Így több (hardver- és szoftver-) technológia együttes fejlődése odavezet, hogy egyre kivitelezhetőbb lesz különböző rendszerek egyetlen használati vagy munkatárgyba, berendezésbe történő integrációja.

Az említett rendszertechnikai elvek és jelenségek végső soron abba az irányba mutatnak, hogy a rendszerek funkcionalitása és teljesítménye, azaz *információfeldolgozó képessége* folyamatosan növekszik, és ez az, amit gyakran „intelligens készülék”-ként vagy „intelligens rendszer”-ként fogalmazzuk meg.

1. Processzorok

A processzortechnológia fejlődése egyértelműen a homogenizált, szabványos magprocesszorok párhuzamos működésű architektúrákba szervezésének irányában halad – méghozzá egyre tömegesebb méretekben.

Várható, hogy a következő évtized végére négy-nyolcezer magos processzorokat fognak készíteni, ami a mai kereskedelemben kapható négy-nyolc, illetve kutató laboratóriumokban előállított negyven-nyolcvan maghoz képest két nagyságrenddel többet jelent.

Egy évtizeden jóval túl – talán az évszázad második felében – várható csak, de valamikor valószínűleg előáll a digitális “agy”, ahol azonban már nem ezrei, hanem információfeldolgozó alapegységek (az akkori „magprocesszorok”) milliárdjai szerveződnek párhuzamos működésű funkcionális alrendszerekbe – kötegekbe, mint az emberi agyban az idegsejtek.

2. Számítógép-rendszerek

A számítógéprendszerek egyre könnyebben méretezhető, *párhuzamos architektúrákká* alakulnak át, és a párhuzamosság hatására előrelépés várható a processzorok közötti kommunikációban is. Mindezt a párhuzamos feldolgozás különböző előnyei, például gyorsaság, megbízhatóság, kompaktság, hibátűrő képesség, alacsony fogyasztás, indokolják. A fejlődés iránya ezen belül egyaránt mutat a *heterogén struktúrák* és a nagy mértékben *szabványosított elemekből építkező* (masszívan párhuzamos) architektúrák felé.

A rendszerek tervezése és üzemeltetése során a *dinamikus rekonfigurálhatóság* és az *automatikus terheléelosztás* lesznek a legfontosabb szempontok. A virtualizáció mindkét szempontból jól használható technika.

A rendszerek készítésekor egyre jobban figyelembe veszik az *élővilágban megfigyelt és bevált elveket*, megoldásokat. Az apró módosításokra és nagy számú egyeden elvégzett, azonnali tesztre építő, *evolúciós jellegű algoritmusok alkalmazásával*, például egy adott célra legmegfelelőbb megoldást, illetve rendszert lehet kialakítani. Az ilyen jellegű kutatások egyik célja az önmagukat *diagnosztizáló, javító* és végső soron *megtervező rendszerek* létrehozása.

Külön említendő a *szuperszámítógép*, amelyek hosszú időn keresztül a legtipikusabb példák voltak a párhuzamos architektúrákra és amelyek mindig is a *központosított informatikai rendszerek* világszintű csúcsteljesítményei lesznek.

Várható azonban, hogy a szuperszámítógépek architektúráis megoldásai, a *masszívan párhuzamos feldolgozási képességek* megjelennek a hétköznapi rendszerekben is. Sok ilyen megoldás található meg már a mai rendszerekben is (például klaszter). Használatuk további elterjedéséhez azonban a *párhuzamos programozás* terén jelentős fejlődésnek kell bekövetkezni, hogy kellően sok és kellően általános feladatra lehessen őket egyszerűen konfigurálni.

3. Elosztott rendszerek (grid)

Az *elosztott rendszerek* olyan párhuzamos architektúrát jelentenek, amely teljesen (CPU-val, memóriával, tápegységgel, hálózati csatlakozással stb.) felszerelt, hálózatba kapcsolt számítógépekből épül fel. Az elosztott rendszer fő előnye, hogy elemeit tömegcikként is meg lehet vásárolni, és ezeket összeintegrálva a szuperszámítógépekéhez hasonló teljesítményre képes – csak jóval alacsonyabb költség mellett. Ez éles ellentétben áll a hagyományos szuperszámítógépek szerveződésével (ld. Számítógéprendszerek).

A *grid* olyan elosztott rendszer, amely egyenrangú, többnyire különböző tulajdonban lévő elemekből áll kitüntetett vezérlőközpont nélküli hálózatban, és amely kívülről, szabványos felületeken keresztül könnyen igénybe vehető szolgáltatásokat nyújt (*utility computing*). A jövőben várhatóan még közelebb kerülnek egymáshoz az elsősorban akadémiai grid-rendszerek és az üzleti jellegű elosztott rendszerek.

Az elosztott (és így a grid-) rendszerek lényeges alkotóelemei a különböző, rendszervezérlő szerepet betöltő *közteszoftverek* (middleware), amelyek lehetővé teszik, hogy egy adott rendszer lehetőségeit a legoptimálisabban használják, vagy új rendszereket hozzanak létre. Segítségükkel a kapacitások rugalmasan vehetők igénybe, ezáltal a rendszerelemek kihasználtsága maximalizálható, és hosszabb távon csökkenthető a beruházás költsége.

Néhány éven belül várható olyan elosztott rendszerek megjelenése is, amelyek közönséges és szuperszámítógépekből álló *heterogén architektúrát* alkotnak, és az ezt kihasználni tudó közteszoftverrel rendelkeznek.

A p2p (peer-to-peer) *tartalomszolgáltató hálózati rendszerben* a csomópontok kliens- és szerverfunkciókat egyaránt ellátnak. Legújabb, harmadik generációs változataik (tisztá p2p) mutatják a legtöbb hasonlóságot a griddel, mert működésük teljesen elosztott, és az összes csomópontjuk egyenrangú.

A következő évtized közepétől a *legkülönbözőbb tárgyak* is informatikai rendszereket fognak tartalmazni, amelyek összekapcsolásával és -integrálásával a gridhez hasonló, lazán szervezett elosztott rendszerek fognak létrejönni.

4. Beágyazott rendszerek

A világméretű információs rendszerek megváltoztatják a hétköznapi tárgyakhoz fűződő viszonyunkat, a tárgyak „intelligensebb”-eknek (*smart objects*) tűnnek, mert egyre több és változatosabb információt egyre nagyobb teljesítménnyel tudnak feldolgozni és visszajelezni.

A beágyazott rendszerek a mindennapi élet különböző használati cikkeiben (például autó, lakóház, telefon) már eddig is megjelentek, de a folyamat a következő néhány évben még erőteljesebb lesz. A miniatürizáció a nagyobb komplexitás felé mutat: a rendkívül egyszerűtől a komplex, multifunkcionális, multimodális ugyanazon feladat megoldásához több alkalmazást futtató és különböző típusú információkat (beszéd, szöveg, kép, videó)

kezelő eszközökig a legheterogénebb beágyazott rendszerek várhatók. A jövő beágyazott rendszerei lényegesen rugalmasabbak lesznek, mint a maiak, mert tervezésük során figyelembe veszik, hogy különböző alkalmazási szituációkhoz, működési feltételekhez, rendszerkövetelményekhez és szabványokhoz kell igazodniuk. A miniatürizáció jegyében a rendszertechnikai megoldásoknál növekvő mértékben használnak fel nanoméretekhez közelítő MEMS (Micro Electro-Mechanical Systems) eszközöket.

5. Szoftverrendszerek

A szoftverrendszerek olyan, alapvetően *szoftverelemekből álló rendszerek*, amelyek lényegi működését nem a szoftvereket futtató fizikai (hardver-) elemek határozzák meg, azaz működésük többé-kevésbé független azoktól.

Egy virtualizált rendszer például teljesen mértékben elkülöníti a rajta működő szoftvert a rendszer részét alkotó hardverelemektől. Más esetekben a szoftverek kihasználhatják a fizikai hardver minden lehetőségét, így ekkor a függetlenség jóval kisebb mértékű.

A fejlődés egyértelműen abba az irányba mutat, hogy a szoftverrendszerek a lehető legteljesebb mértékben függetlenedjenek a működésüket végső soron lehetővé tevő, a szükséges számítási, tárolási és átviteli kapacitásokat biztosító hardverrendszerektől. A virtualizációs köztesszoftver mintegy ragasztóanyagként köti össze a két réteget, amely természetesen szükség esetén feloldható.

5.1 Operációs rendszerek

Az operációs rendszer egy számítógéprendszer azon szoftverkomponense, amely a számítógép hardvererőforrásainak közvetlen koordinálásáért és irányításáért felelős, és amely lehetővé teszi, hogy az alkalmazási rendszerek függetlenedhessenek a konkrét hardverarchitektúrától.

Az operációs rendszer szolgáltatásokat kínál az alkalmazási programok számára, amelyek ún. *rendszerhívások* formájában valamilyen alkalmazásprogramozási felületen (Application Programming Interface, API) keresztül érhetik el ezeket. E rendszerhívások legfontosabb területei: rendszerindítás, programvégrehajtás, megszakításkezelés, valóságos és virtuális memóriakezelés, multitasking, lemezkezelés, file-rendszer, hardvermeghajtás, hálózatkezelés.

A jelenleg leggyakrabban használt operációs rendszerek a Microsoft Windows, a Mac OS, a Linux és a Solaris. A Microsoft Windows-nak jelentős piaci fölénye van az asztali és noteszgépeken, míg a hálózati szerverek terén a Linux a meghatározó.

A vállalati infrastruktúrák üzleti alkalmazásokat futtató szervereinél a helyzet kiegyensúlyozottabb, de a Windows egyre nagyobb szerepet játszik. Hasonló a helyzet a beágyazott rendszereknél, azonban várható, hogy a Linux megőrzi a „beágyazott világ” leggyorsabban fejlődő operációs rendszerének pozícióját.

Bár műszaki értelemben az operációs rendszerek mindig is fontosak lesznek, azonban a szoftverszolgáltatások, illetve általában a közműszerű IT-szolgáltatás terjedésével együtt közvetlen piaci szerepük várhatóan jelentősen lecsökken.

5.2 Köztesszoftverek (middleware)

Az elosztott rendszerek koordinációja, a rendszerek komplexitásának növekedése, az alkalmazások változó körülmények közötti működőképességének biztosítása jelentős

mértékben növeli az operációs rendszer és az alkalmazás közötti szoftverréteg, az ún. köztesszoftver (middleware) szerepét.

Jelentőségüket jól példázza az IBM köztesszoftverre épülő régóta hangoztatott és követett stratégiája: nem fejleszt alkalmazási termékeket, hanem minden fejlesztési erőforrását köztesszoftverek létrehozására használja fel.

A köztesszoftverek közé tartoznak (a teljesség igénye nélkül): 1) a rendszerek *virtualizációját* megvalósító szoftverek; 2) a rendszerek és rendszerelemek koordinált működését és megfigyelését biztosító *rendszermenedzsment*-szoftverek; 3) a rendszerek és rendszerelemek egymásközi *kommunikációját vezérlő-irányító* szoftverek; 4) a különböző *alkalmazási rendszerek integrációját* megvalósító, azaz összetett alkalmazásokat létrehozó szoftverek; 5) az *adatok* – legkülönbözőbb tárolóegységeken és tárolási formátumokon keresztül – szabványos *elérését és integrációját biztosító* szoftverek.

Az eddigiekből is látszik, hogy a köztesszoftvereknek *kiemelt rendszertechnikai szerepük* van. A helyzetet tovább bonyolítja az is, hogy nem csak egy közbenső rétegről van szó, hanem a köztesszoftverek egymást átfedő, egymásra ráakódó rétegeiről, illetve ezek egyfajta rendszeréről.

Ennek következtében nagy valószínűséggel a köztesszoftverek a jövő évtized folyamán sokkal fontosabbak lesznek, mint a hardverelemek és a rendszerszoftverek, és meghatározó rendszertechnikai pozíciót szereznek.

A jövő köztesszoftver-rendszereinél ilyen módon a következő tulajdonságok minél teljesebb elérése lesz a sikeresség kritériuma: a teljesítmény, a szabványosság, a skálázhatóság, az illeszthetőség, a biztonság és az alrendszerek függetlenségének biztosítása.¹¹

5.3 Virtualizáció

Virtualizáción az informatikai erőforrások, a rendszerelemek (processzor, memória, diszk, szerver, operációs rendszer, hálózat, platform, alkalmazás stb.) absztrakt kezelésére lehetőséget nyújtó technológiákat kell érteni. Az „absztrakcióra” való utalás teljesen helyénvaló, hiszen arról van szó, hogy a fizikailag létező elemeket és működésüket valamilyen más módon, más platformon *logikailag* jelenítik meg – a számítógépek memóriájában futó, ember alkotta szoftverek segítségével.

A virtualizáció mélyen áthatja a mai rendszereket, és nemcsak hogy messzire nyúló gyökerei vannak a múltban, hanem a jövő rendszereiben is meghatározó szerepe lesz. Erre utalnak azok az egyéb virtualizációs technológiák is, amelyek például az adat- és folyamatintegráció felől közeledve nyújtanak – többek között – virtuális adatbázisokat és összetett (virtuális) alkalmazásokat képesek létrehozni.

A virtualizáció az informatikai rendszerek IT-szolgáltatássá való „átalakításának” is egyik technológiai alapja. A szolgáltatásminőség garantálásához ugyanis el kell tudni határolni egymástól a mindenkori igényeket a mindenkori kapacitásoktól. Az igények és kapacitások szoros összekapcsolódása ugyanis szűk keresztmetszeteket alakíthat ki. Míg a virtualizációval megteremthető laza kapcsolódás dinamikus kapacitáskiosztást, teljesítményszint-szabályozást és terhelésoptimalizációt tud biztosítani.

¹¹ A köztesszoftverek két alapvető típusa, a virtualizáció és az adatbáziskezelés – meghatározó szerepük miatt – külön is ismertetésre kerül.

5.4 Adatbáziskezelés

Az adatbáziskezelő szoftverek az elsőként megjelent köztesszoftverek közé tartoznak. Mivel az egész informatika valójában az adat- és információkezelésről szól, tulajdonképpen ez a végső célja, ezért az adatbáziskezelés külön is ismertetésre kerül. Változatlanul kulcsszerepet játszanak a relációs modellt támogató adatbáziskezelő rendszerek (Relational Database Management Systems, RDBMS). Nem várható, hogy a jövő évtized első felében e tekintetben alapvető változás következne be, mert az RDBMS-technológia teljesen kiérlelt, jól használható alapterminológia, amelybe nagy összegeket fektettek mind a megrendelői, mind a fejlesztői oldalon.

Inkább olyan területeken válik fontossá az adatbáziskezelés fejlesztése, mint 1) a strukturálatlan adatok kezelése; 2) mobil DBMS megjelenése a legkülönbözőbb végberendezéseken; 3) adatbázisadminisztráció komplexitásának csökkentése; 4) metaadatkezelés; 5) szemantikai információkeresés támogatása; 6) elkülönült adatbázisok együttes kezelése; 7) nyílt forráskódú adatbázismotorok; 8) skálázhatóság.

Az interneten megjelenő, külső adatforrások jelentőségének növekedése ugyanis egyértelművé tette azt a gyakorlati tapasztalatot, hogy szervezeti kereteken belül sem lehetséges egyetlen, egységes, központi adatbázist létrehozni. Ezt mutatja többek között az adattárházak, az adatpiacok és az ún. operatív adattárak elszaporodása is.

Ennek következtében ma már általánosan elfogadott, hogy az adatok szoros integráltsága (adatbázisba szervezettsége) mellett a lazábban integrált megoldásoknak is fontos – gyakran meghatározó – szerepe van.

Az elkövetkező évtized első felében elterjednek az ipari méretekben is használható, – technikailag és szemantikailag – különböző adatforrások egységes kezelését lehetővé tevő *adatintegrációs eszközök*. Az ilyen eszközök ma virtuális adatbázis, vállalati információintegráció, virtuális operatív adattár vagy egyesített (federatív) adatbázis, illetve lekérdező rendszer neven jelennek meg. Az évtized közepétől pedig a hálózaton szétszórt, egészen különböző típusú (strukturált, félig strukturált és strukturálatlan) adatok egységes kezelését biztosító adatkezelési *technológiák integrálódnak* a hagyományos adatbáziskezelő rendszerekkel. Ezt segítik elő az adatváltozások automatikus hálózati továbbításának és nyilvántartásának egyre integráltabb és kiforrottabb megoldásai, valamint a félig strukturált (XML, RDF) és a strukturálatlan adatok kivonatolásának – valamint az azokból strukturált adatok kinyerésének – egyre kifinomultabb technológiái.

Fontos szerepet kapnak az adatok kihasználtságát is figyelembe vevő, hierarchikus tároló rendszerek és a – mind technikai, mind üzleti értelemben vett – *metaadatok elosztott kezelését* biztosító technológiák.

5.5 Szolgáltatás-orientált architektúra (+ webszolgáltatások)

A szolgáltatás-orientált architektúra (Service-Oriented Architecture – SOA) az alkalmazási rendszerek architektúrájának fő elveit és elemeit fogalmazza meg.

A tapasztalat azt mutatja, hogy ezeknél a rendszereknél a nagyon szorosan kapcsolódó, azaz saját használhatóságukat korlátozó, komponensobjektumok helyett, az egymáshoz jóval *lazábban kapcsolódó*, ugyanakkor *kifelé nyitott komponensekre* (az ún. szolgáltatásokra) kell helyezni a hangsúlyt. Ilyen módon ezek a „szolgáltatások” válnak a szoftverek komponensekre való felosztásának elsődleges egységeivé. Az egymáshoz nyílt

és szabványos felületen kapcsolódó szolgáltatásokból a korábbinál jóval rugalmasabb architektúrával rendelkező szoftvereket lehet létrehozni.

Ez a nyílt *szolgáltatás-orientált architektúra* nagyobb mértékű elmozdulást jelent a ma uralkodó, zárt szoftverarchitektúrákhoz képest, mint amilyen a kliens-szerver vagy a böngésző alapú architektúra-modellekre való áttérés volt korábban a nagyszámítógép-terminál alapú architektúrákról. Az architektúra fő alkotó elemei: 1) maguk a webszolgáltatások (SOAP/REST szabvány szerint) belső vagy külső kapcsolattal; 2) vállalati szolgáltatásbusz (Enterprise Service Bus, ESB); 3) metaadattár.

A SOA-elveket használva ma a szoftvercégek szolgáltatás-orientált alkalmazásokat fejlesztenek ki, illetve meglévő szoftvereiket ennek megfelelően írják át. Ez a megközelítés az alkalmazási rendszereket lazán integrált (és jellemzően már kész) szoftvermodulokból (szolgáltatásokból) építi fel. Az ilyen fajta alkalmazásintegráció kulcsszerepet játszik majd mind a *szervezetten belüli* (hagyományos rendszerek újrafelhasználása), mind a *szervezetek közötti* (elektronikus kereskedelem, B2B) információkezelés összehangolásában.

A SOA elterjedése már a következő években éreztetni fogja hatását, ahogy a nagy szoftvergyártók termékeiket átalakítják SOA-ra, és a monolitikus szoftvercsaládok helyett az *összetett alkalmazások* – azaz több, különböző szolgáltatásból összeépített alkalmazások – válnak dominánssá az új rendszerek kialakításánál. A jövő évtized közepétől pedig a szoftvercsomagok licencvásárlása helyett a szolgáltatások *előfizetése* kezd el tipikussá válni.

5.6 Ágensrendszerek

Az ágensrendszereket ágenssek, mint komponensek alkotják. Ágensseken szoftverágenssek értendők, azaz olyan független szoftverkomponensek, amelyek magas szintű támogatást nyújtanak a rendszer felhasználójának (például postakezelő ágenssek). Egyik fő típusuk az *egyedi ágens*, amelyik saját ismeretekkel és következtetési – esetleg tanulási – képességekkel rendelkezik. A másik típus, amikor több kölcsönható ágens *multiágens rendszereket* alkot, és az egyes ágenssek képességeit meghaladó intelligens viselkedést mutató ágenscsoportok jönnek létre (*rajintelligencia*). Az elektronikus kereskedelem, filmgyártásban, online árverések során stb. gyakran helyettesíthetik az embert.

Az ágenseket különböző alkalmazásokba, például párhuzamos problémákat megoldó rendszerekbe „ágyazzák be”. Szimulációs alkalmazásaik még nagyobb figyelmet fognak kapni. A jövő ágenstechnológiája erőteljesebben igyekszik integrálni az objektumorientált és elosztott feldolgozói környezeteket, a komponensalapú szoftverfejlesztést és a mesterséges intelligenciát.

Az ágens- és a beszédtechnológiák együttes alkalmazást ötvöző szintetikus karakterek kevésbé lesznek téma-, illetve honlap-specifikusak. Az interakciós és a kommunikációs technológiák (gesztus-, hangulat-, beszédfelismerés, kontextus-megértés stb.) szinkronba hozásának eredményeként lényegesen interaktívabb viszonyba kerülnek környezetükkel.