

MŰSZAKI TUDOMÁNYOS FÜZETEK

1

JODÁL ENDRE

SZÁMÍTÁSTECHNIKA AZ EZREDFORDULÓ KÜSZÖBÉN



AZ ERDÉLYI MÚZEUM-EGYESÜLET KIADÁSA
KOLOZSVÁR 1992

Az Erdélyi Múzeum-Egyesület
Műszaki Tudományi Szakosztályának
tanulmányorozata

Szerkeszti
Jenei Dezső



Készült a Református Egyház Misztótfalusi Kis Miklós
Sajtóközpontjában nyomdájában
Felelős vezető Tonk István

<https://doi.org/10.36242/mtf-01>

Mi volt eddig?

A számlálás, számolás művelete szinte egy idősebb az emberi civilizációval. Az ember egyik legrégebbi törekvése volt, hogy bizonyos ismétlődő, monoton műveleteket — ilyen például az összeadás és a szorzás — valamilyen könnyen használható segédeszközzel végezzen el. Kis túlzással az első ilyen eszköznek az i.e. 6. században élt Püthagorasz által összeállított szorzótáblát tekinthetjük. A négy aritmetikai alapművelet elvégzésére használt, legrégebben ismert eszköz az abakusz (bár a Belső-Mongóliában talált, körülbelül háromezer éves elefántcsont pálcikákról bebizonyítottak tűnik, hogy a Csou-dinasztia idejéből származó, tízes számrendszerű matematikai eszközről van szó!). Fennmaradt dokumentumok tanúsága szerint az abakusz egymástól távol eső kultúrákban egyaránt használatos volt: a mediterrán övezetben az i.e. első évezredből maradtak fenn bizonyítékok; Hérodotosz szerint az egyiptomiak i.e. az 5. században már elterjedten használták; az Amerika partjaira szálló spanyolok megtalálták a mai Mexikó és Peru területén kialakult precolumbiánus civilizáció jeleként; egy jellegzetes formája már Konfuciusz idején használatos volt Kínában és Japánban; s bármilyen furcsának tűnne is mai technikai világunkban, még ma is elterjedten használják a Távol-Keleten.

Jelentős mérföldkövet a 17. században John Napier (1550—1617) által megalkotott mechanikai eszköz jelentett, amellyel minden aritmetikai műveletet végső fokon összeadásra és kivonásra lehetett visszavezetni. Napier másik nagy felfedezése a logaritmus és az annak alapján működő logarléc, amely mind a mai napig a mindennapi műszaki számítások egyik alapvető eszköze (bár az elektronikus zsebszámológépek nemsokára teljesen ki fogják szorítani a gyakorlatból).

A 17. század máskülönbén a mechanikai számológépek megjelenésének periódusa. E gépek alapvető feladata diszkrét objektumok számlálása és a tizedes pozíciók közötti átvitel megvalósítása volt. Ez utóbbira a legegyszerűbb megoldásnak a fogaskerék-átvitel bizonyult: a meghajtó kerék 2, a meghajtott 20 foggal rendelkezett, így a meghajtó kerék egy teljesen megtett fordulattal a meghajtott kereket a teljes fordulat $1/10$ -ével mozdította el. A következő lépés az elemi összeadómű volt, amelyet a fenti számológépekből lehetett felépíteni.

A legelső ilyen elgondolású számológépet, úgy tűnik, 1623-ban alkotta Wilhelm Schickard tübingeni matematikatanár. Szerkezetét "számoló órának" nevezte, s egy, a kor nagy asztronómusának, Johannes Keplernek (1571—1630) küldött levelében írta le. Bár valószínűleg nem működött hibátlanul és túl nagy sebességgel, kétségtelenül az első számológépnek tekinthető, amely, legalábbis elméletileg, képes volt mechanikai úton két szám összeszorzására. Schickard egész családjával az 1635-ös járvány áldozata lett, Kepler pedig feltételezhetően nem tulajdonított különösebb jelentőséget a leírt szerkezetnek. Mindenesetre ez a levél az egyetlen bizonyíték, hogy a számológép létezhetett. Johann Ciermans 1640-ben megjelent könyvében (*Disciplinae Mathematicae*) említést tesz egy bizonyos mechanikus szerkezet létezéséről, amely kerek mozgása révén szorzási és osztási műveletek elvégzésére képes, de további részleteket nem közöl.

Az első, bizonyíthatóan létezett mechanikai számológép megalkotása Blaise Pascal (1623—1662) nevéhez fűződik. 1642-ben — 19 éves korában — önműködő összeadó gépet szerkesztett apja munkájának megkönnyítésére. Az ötletet annyira érdekesnek találta, hogy további gépeket is kezdett összeállítani. Egyesek 6, mások 8 pozíciós gépek voltak; nagy részük tízes számrendszerű egészekkel dolgozott, de akadt olyan is, amely pénzösszegek kezelésére is alkalmas volt — tehát 12-es, illetve 20-as egységekkel kellett dolgoznia az összegnek sous-ban, illetve dinárban kifejezett formájáért. 1647-ben szabadalmaztatta ötletét, így hivatalosan is a Pascal-gépet tekintik az első mechanikai számológépnek. Egyik alapvető ötletét a mai modern elektronikus számítógépek is használják: a kivonást a kivonandó komplementumának a kisebbítendőhöz való összeadására

vezeti vissza. Pascal gépe alapvetően összeadó gép volt, a szorzást és osztást ismételt összeadással, illetve kivonással végezhetette el.

A következő lépést Wilhelm Gottfried Leibniz (1646—1716) tette: gépesítette a szorzás és osztás műveletét is, továbbfejlesztette Pascal gépét. Az újdonságot egy henger képezte, amely lehetővé tette, hogy az összeadó gépbe beírt szám annyszor megismétlődjön, ahányszor azt a szorzó mutatja. Első modelljét Leibniz 1671-ben mutatta be, s ennek köszönhető, hogy 1673-ban a londoni királyi akadémia tagjai közé választotta.

Bár nem kimondottan számológép, későbbi hatása miatt igen fontos jelentősége van Joseph-Maria Jacquard (1752—1834) szövőgépének, amellyel a mintaszövést sikerült gépesítenie. Jacquard gépéhez a szövés mintát kartonlapokra kellett lyukasztani, s egy tapintópálcás berendezés útján lehetett vezérelni a gépet. Ez az ötlet szolgált a későbbi számítógépek adatbeviteli eszközeinek, illetve külső memóriájának alapjául.

Ugyancsak ebben a periódusban születtek olyan elméleti felfedezések, amelyek jóval később a tulajdonképpeni számítógépek megalkotásának előfeltételeiként szolgáltak. Ezek közül a legfontosabbak: a formális logika megalapozása ugyancsak Leibniz által, amelyet később George Boole (1815—1864) fejlesztett tovább; a kettes számrendszer alaposabb tanulmányozása, a bináris aritmetika szabályainak meghatározása stb.

Az első javaslatot önműködő számológépre Charles Babbage (1782—1871) tette, aki a logaritmustáblák készítését akarta gépesíteni. Babbage 1823. július 13-án mutatta be a Királyi Csillagászati Társaság ülésén a gép első modelljét, a "differenciagépet". Első példányának működési elve egyszerű volt: a szerkezet egymás mellé helyezett tízfokú fogaskerekekből állt, s nagyon hasonlított egy mai gépkocsi kilométer-számlálójához. Ha az egyes helyértékű számok kerek egy teljes fordulatot tett, akkor egy hellyel elmozdította a tízes helyértékű fogaskereket, annak teljes fordulata a százat mozdította el és így tovább. Egy bonyolult számítás elvégzésére azonban rendkívül összetett szerkezetre volt szükség. (Babbage tervei alapján George Scheutz, a gazdag stockholmi nyomdász el is készítette a differenciagépet: 1843-ban fából, vasból és kartonból készült el az első kísérleti példány, majd 1850-ben megvalósult az első teljesen fémből készült gyakorlati példány. Továbbfejlesztett változata 1855-ben a párizsi kiállításon aranyérmert nyert, s a gyártók nem kis meglepetésére legnagyobb csodálójá maga Babbage volt.)

Babbage és Scheutz gépe új irányzatot hozott a számolás automatizálásában. Míg elődei egy-egy összeadás vagy kivonás (Pascal-gép), illetve szorzás vagy osztás műveletét tudták elvégezni (Leibniz-gép), az új szerkezet bevezette a műveletsorozat elvét.

1833-ban Babbage új tervet készített az "analitikus gép" megvalósítására. Korszakalkotó ötletnek tekinthető az a javaslata, hogy a mindig következő műveletet lyukkártya segítségével kellett közölni a géppel. A szerkezet szívét — vagy még inkább "agyát" — az a két kártyacsomag jelentette, amelyek közül az egyik perforációk formájában tartalmazta az előre meghatározott műveletsort, míg a másik az operandusok szerepét játszó számokat. A kor technikai fejlettsége nem tette lehetővé a terv valóra váltását, elvei alapján azonban joggal tekinthető általános célú, programvezérelt számítógépnek. Lady Ada Augusta, Byron leánya (akiről máskülönben az egyik legújabb programozási nyelvet elneveztek), megsejtve a gépben rejlő lehetőségeket, nagy érdeklődéssel tanulmányozta, és programokat készített számára, s így a számítástechnika első programozójának tekinthető.

Száz évnek kellett elteltie ahhoz, hogy Babbage előrelátó terve — a matematikai műveleteket az emberi agy alkalmazkodóképességével és a gép pontosságával elvégezni képes szerkezet — gyakorlatilag is megvalósulhasson.

Howard Aiken professzor 1939—1944 között építette meg Mark I nevű relés számítógépét, amely Babbage alapelvét megvalósította (s egyben elindította hódító útjára a világ első számú számítógépgyártó iparóriását, az IBM-et). Ugyancsak erre a periódusra esik a német Konrad Zuse tevékenysége, aki több relés kiépítettségű, bináris aritmetika alapján működő, programvezérlésű gépet szerkesztett. Ezekben az években készült el egy másik elektromechanikus számítógép is a Bell Telephone Laboratories-nél, amely a táv-

adatfeldolgozás csíráit is magában hordozta, amennyiben telexgépen küldött adatokat volt képes feldolgozni, majd az eredményeket szintén telexen továbbította.

Az elektromechanikus (relés) számítógépekkel körülbelül egy időben folytak kísérletek elektronikus számítógépek megvalósítására is. Ezek korszakát századunk ötödik évtizedétől számíthatjuk.

A harmincas évek második felében a háborús előkészületek elsőrendű feladatát léptették elő a bombázások irányítására szolgáló táblázatok gyors előállítását, s erre a célra kellett kidolgozni az első elektronikus digitális számítógépeket.

Az első elektronikus számítógép az ENIAC (Electronic Numerical Integrator And Computer) volt, amelyet hivatalosan 1946-ban mutattak be. Noha elsősorban ballisztikai számításokra készült, az ENIAC tulajdonképpen általános célú számítógép volt, s elektronikus, ugyanis az adatok belső tárolása és mozgatása elektronikus áramkörök segítségével zajlott le. A J. W. Mauchly, I. P. Echert és H. H. Goldstine vezetésével a pennsylvaniai egyetemen elkészült számítógép mai szemmel valóságos mamutnak tekinthető: 70 m² felületen (40 panelen) körülbelül 18 000 elektroncső foglalt helyet, és 150 kW (!) beépített teljesítményt testesített meg. Tárolási kapacitása 20 tízjegyű vagy 40 ötjegyű szám volt, s kézi dugaszolás kapcsolótáblán kellett előhívni az elvégzendő műveleteket. Sebessége viszont szinte százszorosa volt a Mark I-ének.

Neumann János (1903—1957), akinek a modern elektronikus számítógépek megteremtését tulajdonítják, csak 1943-ban került kapcsolatba az ENIAC-tervvel. A nagyszerű terv először elkápráztatta, de csakhamar nyilvánvaló hiányosságaira összpontosította figyelmét. 1944 második felétől Neumann már ténylegesen részt vett egy újabb gép tervezésén. 1945. június 30-án jelent meg H. H. Goldstine-nal együtt készített dolgozata, amely alapvetően meghatározta mind a mai napig az elektronikus digitális számítógépek építésének alapjait (ennek máig is érvényes, globális elnevezése a "Neumann-architektúra"). E dolgozat alapján minden számítógépnek 4 alapvető építőeleme kell hogy legyen: 1. memóriagység, amely a feldolgozandó adatokat és a feldolgozó utasításokat — ugyanabban a kettes számrendszerben kódolva — egyaránt tárolja; 2. aritmetikai egység, amely a bináris aritmetika szabályai alapján elvégzi a négy alapvető műveletet; 3. központi vezérlőegység, amely a memóriaegységben tárolt utasításokat leolvassa és értelmezi, majd emberi beavatkozás nélkül a kívánt műveletet az előírt adat(ok)al elvégzi; 4. be- és kimeneti egység, amelynek révén a számítógéprendszer és az ember (külvilág) közötti kapcsolat, kommunikáció megvalósul. (Itt érdemes megjegyezni, hogy Neumann világosan látta, hogy a kommunikációnak a grafikai változata az, amely legjobban alkalmazkodik az ember képességeihez, s ezért még 1945-ben javasolta, hogy az új gépet lássák el képkimenettel!)

E leírás legzseniálisabb tételének a végrehajtandó műveleteknek adatok módjára történő kódolása, a gép memóriájában való tárolása és kezelése bizonyult, hiszen ezáltal teljes egészében ki lehetett használni a nagy műveleti sebesség adta lehetőségeket. Még fontosabb viszont az, hogy lehetővé vált a tárolt programok adatként való kezelése (módosíthatósága!), amely forradalmasította magának a programozási tevékenységnek az értelmezését. Erről Neumann *A számológép és az agy* című könyvében így ír: "... a gép kellő utasításra számokat (vagy utasításokat jelentő számokat!) hívhat le a memóriájából, feldolgozhatja az így nyert számadatokat (akkor is, ha utasításokat jelentenek!), majd a feldolgozás eredményeit ismét betárolhatja memóriájába (ugyanarra vagy más helyre!). Magyarán: a gép megváltoztathatja memóriájának tartalmát, sőt éppen ez az ilyen gépek normális működési módja."

A fenti elveket megtestesítő első számítógép 1952-ben készült el Princetone-ban, az Institute for Advanced Study-nál, s a Neumann-architektúra mind a mai napig egyeduralkodó: minden számítógép tartalmazza a négy alapegységet, ám a technológiai-technikai megvalósításban óriási, szinte döbbenetes iramú változások mentek végbe.

Eddigi fejlődésük során az elektronikus számítógépeknek négy nemzedékét lehet körülhatárolni.

Az első generációhoz az elektroncsöves áramkörökből felépített számítógépek tartoznak. Igen nagy méretű, komoly energiafelvételű és hőleadású berendezések voltak, s szinte csak speciális (elsősorban katonai, kisebb mértékben kutatási) célokra használták őket. Programozásukra kizárólag a bináris jelekből álló gépi kódot lehetett használni.

A második generációt elsősorban a diszkrét félvezetős elektronikával megvalósított architektúra jellemzi. A tranzistoros kiépítés előnyei között elsősorban a nagyobb megbízhatóságot, a kis méreteket és a nagy műveleti sebességet említhetjük. Ennél a nemzedéknél jelenik meg először a ferritgyűrűs operatív memória, s ami talán még fontosabb, a megszakítási rendszer. Ez utóbbinak köszönhető, hogy az időigényes be-, illetve kimeneti műveleteket szét lehetett választani az aritmetikai-logikai műveletektől, s így a központi egység (aritmetikai és vezérlőegység) nagy műveleti sebességét sokkal jobban ki lehetett használni. A programozás megkönnyítésére megjelennek az első szimbolikus programozási nyelvek (autokód, asszembler, sőt az első magas szintű programozási nyelv, a FORTRAN). Ennek köszönhetően a számítógépet felhasználók köre bővülni kezdett, olyanok is gépközelbe kerültek, akik nem kimondottan számítástechnikai szakemberek. Megkezdődött a pénzügyi-gazdasági, államigazgatási, tudományos-technikai feladatok megoldásának számítógépes megközelítése. E területek szakembereinek segítségével valósultak meg az első operációs rendszerek, amelyeknek elsődleges feladata a számítógépek erőforrásainak gazdaságos kihasználása, a felhasználó és a konkrét gép közötti kapcsolat megteremtése.

A harmadik generációról jelennek meg az első integrált áramkörös megvalósítások, de talán még jellemzőbb a hatás a szervezésben, a rendszertechnikában megfigyelhető változás. Megjelennek a nagy felületű kapacitású mágneslemez és mágneszalagos háttérmemóriák, s az egyre bonyolultabb kiépítésű rendszerek gazdaságos kihasználása és vezérlése már nem valósítható meg hatékony operációs rendszer közreműködése nélkül. Az egyfelhasználós működtetést felváltja a többfelhasználós (multiprogramozásos) üzemmód, amely újabb feladatokat ró az operációs rendszerre: az operatív memóriában egy időben jelen levő különböző programokat meg kell védeni egymástól, az egész rendszer erőforrásait úgy kell ütemezni, hogy minden felhasználót — fontosságukat is figyelembe véve — optimálisan lehessen kiszolgálni. Erre a generációra jellemző más-különbben a magas szintű programozási nyelvek általános elterjedése is.

A hetvenes évek elejére tehető a negyedik generáció megjelenésének időpontja, bár az előző generációhoz viszonyítva itt már éles határ igen nehezen vonható. A mennyiségi változást talán a tömeges elterjedés jelenti, a minőségit viszont a mikroprocesszoron alapuló rendszerek megjelenése, a számítástechnika és távközlés egymásra találását jelképező számítógép-hálózatok kialakulása testesíti meg, s ennek rendszertechnikai vetülete, az elosztott adatfeldolgozás.

Manapság, amikor elektronikus számítógépekről beszélünk, a szupravezetési technikát felhasználó szuperszámítógépekről a hobbi kategóriás személyi számítógépekig terjedő, igen széles skálájú eszközhalmazra gondolhatunk. S a láthatáron máris feltűnt az ötödik generáció, amely a soros elvű gépi feldolgozás helyébe a párhuzamos, tudásbázison alapuló információfeldolgozást állítja. Ehhez a teljesen új számítástechnikai felfogáshoz is egy Neumann-munka adta a kiindulópontot: a sejtautomata-elmélet.

Programozás: művészet — tudomány — technológia

A "klasszikus" számítástechnika eddigi fejlődése során egyértelműen megállapítható, hogy a szoftver (= programtermékek) részaránya a számítógéprendszerek összköltségében egyre növekedett, s jóval meghaladta a hardver (= technikai eszközök) által képviselt összegeket. Ennek legfontosabb okozója egyrészt az egyre növekvő kód-mennyiség (a klasszikus, Neumann-elvre épülő számítógépek a szoftverre hárítják a feladatmegoldás oroszlánrészét), másrészt a programminőség garantálásához szükséges egyre nagyobb munkaráfordítás. A fejlődés másik meggondolkodtató jellegzetessége, hogy míg

a hardver példa nélküli gyors fejlődése, a mini- és mikroelektronika megjelenése és egyre olcsóbbá válása a potenciális felhasználók körét szinte napról napra tágította, a programozástechnika viszonylagos bizonytalanságot mutatott (s talán mutat még ma is). Érdekes tényként állapítható meg, hogy míg a termelés más területein az iparszerű előállítás (technológia) az alapvető forma, a programozás területén még ma sem lehet kiforrott technológiáról beszélni. Ennek egyenes következménye, hogy a potenciális felhasználók nagy része nem képes saját maga előállítani a problémáikat megoldó programokat, másrészt a számítástechnikai szakemberek által előállított programok igen gyakran nem elégítik ki az egyéni felhasználók tényleges igényeit. (Ennek az ellentmondásnak feloldására alakítanak ki sokszor számítástechnikai szakemberekből és az alkalmazási terület ismerőiből álló vegyes csoportokat egy-egy nagyobb feladat megoldására. Az ilyen megoldás mesterként, nem életképes mivoltát csak az ismeri igazán, aki dolgozott már ilyen vegyes csoportokban!)

Mit is értünk azonban a "programozás technológiája" megnevezés alatt?

Mint bármely más terület termékeinek előállításánál, a programtermékek vonatkozásában is igaz, hogy technológia = eszközök + módszerek. Természetesen az eszközök és a módszerek valamilyen értelemben mindig kölcsönhatásban vannak egymással, egymást kiegészítik, befolyásolják. Az eszközök nagymértékben meghatározzák a módszereket, s a módszerek fejlődése jelentős hatással van az egyre fejlettebb eszközök előállítására.

Kísérjük végig röviden az eszközök, majd a módszerek fejlődésének fontosabb állomásait.

Az avatatlanok gyakran még ma is olyan értelmességet tulajdonítanak a számítógépeknek, amely azonos lenne az ember tudatos tevékenységével. Tudásunk mai állása szerint azonban ez hibás elképzelés: a (mai) számítógép csak azt képes elvégezni, amire előzőleg "megtanították" — igaz, azt már az emberre jellemző sebesség többszörösével. Mindig az előre megalkotott program testesíti meg az "okosságot", a döntési képességet — soha nem maga a gép. Alapvető feladat tehát a minél jobb, gyorsabb, gazdaságosabb, biztonságosabb programok előállítása.

Kezdetben tehát létezett egy gép, amely néhány igen primitív műveletet volt képes felismerni és végrehajtani (s ez a megállapítás nagyjából még ma is érvényes!), és létezett egy kisszámú, magas képzettségű szakember — rendszerint maguk a számítógépek megalkotói —, aki képes volt egy-egy feladatot ezekre az egyszerű műveletekre lebontani s a gép által közvetlenül megérthető és végrehajtható formára hozni (programozni). Ez a tevékenység amellét, hogy jelentős szellemi munkert követelt, igen nagy mennyiségű monoton, magas hibalehetőségű, úgynevezett "fekete" munkát is igényelt.

Igen fontos lépést jelentett annak felismerése, hogy ezt a fekete munkát ugyancsak a gépre lehetne bízni: a műveleteket, operandusokat kézenfekvő szimbólumokkal kéne helyettesíteni, majd ezek "lefordítását" a gép által közvetlenül felismerhető formára maga a gép végezze el. Így jelentek meg az első "programgenerátorok": az asszemblerek programok (ezek végezték el a szimbólumok lefordítását), programbetöltők (ezek helyezték el a programokat a gép memóriájában, hogy közvetlenül végrehajthatók legyenek).

Ezekkel az eszközökkel a programozás igen sokat nyert mind a programok előállításának megkönnyítésében, mind a rájuk fordított idő lerövidítésében, s csakhamar egyre nagyobb feladatokat lehetett a számítógéppel elvégeztetni. A programozás mégis csak egy igen szűk körű szakembergárda kiváltsága maradt, ugyanis a szimbolikus programozás még mindig igen mélyreható ismereteket követelt a számítógép intim felépítéséről és működéséről. De még az ő számukra is szinte legyőzhetetlen akadályokat jelentettek az olyan jelenségek, mint a logikai és kódolási hibák igen nagy aránya és nehézkes kiszűrése, az egyre nagyobb programok áttekinthetatlensége, a végül elkészült programok karbantarthatósága, a felmerülő változtatásokkal szembeni rugalmatlansága, az egyik gépről a másikra való átvihetatlensége.

Nemsokára az is nyilvánvalóvá vált, hogy sok rutintevékenységet, amely nem tartozik szorosan a megoldandó feladathoz, szintén a számítógépre lehet bízni (ilyenek

például a be-, illetve kimenettel kapcsolatos tevékenységek ellenőrzése, az elég szűkösen rendelkezésre álló hardver erőforrások gazdaságos kihasználása stb.). Így jelentek meg az eleinte egyszerű, majd mind összetettebb, hatékonyabb operációs rendszerek, amelyek a számítógépből és a hozzákapcsolt periferikus egységekből álló, egyre komplikáltabb rendszer összehangolt működését, hatékony kihasználását biztosították.

Az eszközök fejlődésének új minőségi szintjét jelentette a magas szintű programozási nyelvek megjelenése, amely lehetőséget adott arra, hogy egy-egy szakterület bevált formalizmusával lehessen az illető terület feladatait definiálni, a megoldási algoritmusokat leírni. Így például a matematikus vagy mérnök szinte változtatás nélkül használhatta a megszokott képleteket, a könyvelő a bevált számítási és tervezési módszereket stb. Egyre-másra jelentek meg a különböző problémaorientált programozási nyelvek, amelyeknek legfontosabb előfutárai a tudományos-műszaki számításokra specializált FORTRAN, illetve ALGOL, vagy a gazdasági-pénzügyi adatfeldolgozásra kifejlesztett COBOL. Ezeknek köszönhetően a programozónak már nem volt szüksége, hogy a számítógép felépítésének és működésének minden apró részletét ismerje (ezeket a fordítóprogramok, illetve más rendszerprogramok építették be a szimbolikus formában kódolt programba), s így a különböző szakterületek ismerőinek is megnyílt az út a számítógépek közvetlen felhasználásához. A fordítóprogramok a szimbolikus programok lefordítása mellett nagy segítséget nyújtanak a programozási hibák felderítésében, a programok "belövésében", a minél nagyobb hatékonyságú végrehajtható programok előállításában, s ezáltal jelentősen megnövelik a programozás hatékonyságát.

Ezeknek az előnyöknek köszönhetően, a magas szintű programozási nyelvek ma is a programozás legfontosabb eszközei. S mivel a programozási nyelvek megvalósításához egyre hatékonyabb, sőt bizonyos fokig automatizálható eszközök jelentek meg, szinte megállíthatatlan folyamat kezdődött: jóformán tízesével jelentek meg az új programozási nyelvek (egyes feltételezések szerint számuk ma már meghaladja a beszélt természetes nyelvek számát!).

A programozási tevékenység rövid időn belül számottevő fejlődésen ment keresztül, s ezáltal egyre több új elem került az általános elméletbe: új típusú adatszerkezetek, vezérlési szerkezetek, új programozási kategóriák. Ezek beépítése a már meglevő nyelvekbe vagy nagyon nehézkes, vagy egyáltalán nem valósítható meg. Minél általánosabb egy programozási nyelv, annál nehézkesebb, gazdaságtalanabb fordítóprogramja. Sokkal kézenfekvőbb mindig egy-egy olyan programozási nyelv felhasználása, amely elsősorban egy adott szakterület jellegzetes adatstruktúráit ismeri fel s így a legjobban felel meg az éppen aktuális algoritmus előnyös programozásának. Az egyre fejlettebb operációs rendszerek ülnyomó többsége biztosította a különböző nyelveken írott programrészeknek egyetlen egységes programba történő egybeillesztését. Egy másik irányzat azt a célt tűzte maga elé, hogy megszünteti ezt a nyelvi Bábelt. Az időközben kikristályosodott programozási elmélet alapján olyan általános programnyelv előállítása került a figyelem középpontjába, amely a legkülönbözőbb szakterületek igényeit is kielégíti. Így született meg az Ada programozási nyelv, amely azonban a mögötte álló hatalmas gazdasági és politikai erők ellenére is, úgy tűnik, nem fogja beváltani a hozzá fűzött reményeket. Ez annál is valószínűbb, mivel 1995 körül várható az ötödik generációs számítógép megjelenése, amelyet nemcsak új, a hagyományos Neumann-elvtől eltérő típusú architektúra, hanem a programozásnak is egészen új szemlélete fog jellemezni.

A módszerek változásának tekintetében is jól körülhatárolható néhány fejlődési szakasz.

Elterjedésük első szakaszában a számítógépeket a kisméretű és drága memóriák és a műveleti sebesség viszonylagos alacsony szintje jellemezte, tehát a programozók legfontosabb szempontja az idővel való versenyfutás és a programok terjedelmének mindenáron való minimalizálása volt. Ebben az időben — a gyakorlati tapasztalat és az általános érvényű szabályok hiányában — a programozás inkább hasonlított művészetre, mint rendszerezett tervezésre. Ekkor még a jó programozó jellemzői elsősorban a takarékoság

(tárfoglalással és végrehajtási idővel) és a minél több "egyéni" fogás használata. Így a programtermékek is egyedi alkotások voltak, sokszor szinte teljesen érthetetlenek mások, de igen gyakran — bizonyos idő elteltével — maguk az alkotók számára is.

A hatvanas évek elejétől a számítástechnika széles körű elterjedése egyre sürgetőbbé tette a programozás jellegének megváltoztatását a nagyipari termelés mintájára, amelynek jellemzői a magas fokú szervezethez és a széles körű szabványosítás.

A magas szintű programozási nyelvek megjelenésével új módszerként jelenik meg az úgynevezett moduláris programozás, amely az építőkockák elvén alapul. A módszer lényege az, hogy a megoldandó feladatot egymástól független részfeladatokra bontják fel, amelyeket külön lehet tervezni és kódolni, esetleg átvenni már létező programokból. Mivel ezek az egységek — modulok — egyszerű, kis terjedelmű logikai egységek, a programozó könnyen alkalmazhatja a programozási szabályokat, megfelelően választhatja meg a legalkalmasabb programozási nyelvet, pontos dokumentációt készíthet számára, amely a későbbi karbantartás, illetve módosítás számára rendszerint elengedhetetlen. Ezeket a programmodulokat egyenként lehet tesztelni, amikor a fő hangsúlyt még az egyedi programfunkció helyes működésének elérésére lehet helyezni anélkül, hogy a végső programtermékek mint rendszernek működését figyelembe kéne venni. A megvalósítás következő szakaszában ezekből a hibátlanul működő építőkövekből lehet felépíteni a végső programot, amikor a figyelem teljes egészében a program általános logikai működésére összpontosulhat.

A moduláris programozási módszer jó ideig kielégítette a programozási tevékenységgel szemben támasztott követelményeket. Sokáig az a nézet uralkodott, hogy az egyszerűbb programoknál bevált módszer tovább alkalmazható az egyre komplexebb problémák megoldásánál is. A mennyiségi felhalmozódás azonban csakhamar komoly nehézségeket okozott, s minőségi átalakulást, új szemléletet tett szükségessé. A minőségi ugrás a hatvanas évek végén következett be a strukturált programozás körvonalazódásával. Az új módszer alapelve az, hogy a programot egy modellsorozattal kell megközelíteni, ahol a modell egy olyan program, amely a célprogram működését valamilyen "magasabb intelligenciájú" gépen szimulálja. Természetesen ez a gép nem szükségszerűen létezik, tehát a modellsorozat következő eleme az előzőnek egy konkrétabb lebontása kell hogy legyen, egészen addig, míg az utolsó változat egybeesik egy létező számítógép konkrét lehetőségeivel. Nagyon leegyszerűsítve a fogalmakat, a strukturált programozás nem egyéb, mint a feladatok szisztematikus megoldása olyan sorrendben, amely természetes logikai módon az ember szellemi tevékenysége során lezajlik.

A strukturált programozás megjelenése óta gyakran felülvizsgálták a programozási stílusokat, s ez együtt járt a programozási nyelvek felülvizsgálatával is. A hetvenes évek elején megjelent Pascal, majd ennek minőségileg is továbbfejlesztett változata, a Modul-2, már felépítésénél, adat- és vezérlési szerkezeténél fogva is szinte kényszeríti a programozót a rendszeres, strukturált programozásra.

A közelmúltban megjelent egy új, az előzőektől eltérő alapokon nyugvó irányzat, amelyet összefoglaló néven logikai programozásnak neveznek. Ennek az irányzatnak két alapvető ága a funkcionális, illetve a predikátumkalkulus programozás, s mindkettő a logikai formulákat használja. Ennek az irányzatnak jellegzetes képviselője a programozási nyelvek területén a PROLOG. Az új stílus szélesebb körű elterjedésének egyelőre még határt szabnak azok a nehézségek, amelyeket a Neumann-elvre épülő, szekvenciális architektúra támaszt. Jövője viszont biztosítottnak látszik, mivel az ötödik generációs számítógép kiinduló programozási nyelvének választották.

A programozási stílusok fejlődésével a programozási tevékenység visszafordíthatatlan lépést tett a művészeti jellegtől a technikai, az ipari jelleg felé.

Az eddig áttekintett eszközök és módszerek valóban megkönnyítették a programozási tevékenységet — de csak a számítástechnikai képzettségű szakemberek számára. Az átlagos felhasználó "álma" továbbra is egy olyan számítógép, amivel csak a feladatot kell közölni, a megoldás részletezése nélkül. Más szóval, csak azt kelljen közölni, hogy "mit" kell a számítógépnek megoldania, és nem azt, hogy "miként". Tulajdonképpen csak

egy ilyen lehetőség vezethet el ténylegesen oda, hogy valóban technológiáról lehessen beszélni a programtermékek terén. E cél érdekében már elég rég folynak a kutatások, gyakorlati próbálkozások, ha nem is a tényleges automatizálás felé, legalább a rutinszerű, tényleg a felhasználástól független részfunkciók automatikus megvalósítása érdekében.

Egyik ilyen irányzat, amely bizonyos eredményeket is fel tud mutatni, a "méretre szabható" programtermékek előállításával foglalkozik. E módszer lényege az, hogy a szűkebb-tágabb szakterület legáltalánosabb feladatait megoldó, erősen parametrizált programcsomagot állítanak elő. Az ezekből szelektált és csoportosított programnak ki kell elégítenie az átlagos felhasználó minden elképzelhető igényét. Gyakorlatilag a felhasználó a paraméterek beállításával fejezi ki igényeit, rendszerint egy párbeszédés környezetben, s ezek alapján a rendszer maga választja ki a megfelelő programegységet s generálja a végső programterméket. Bizonyított tény, hogy a méretre szabható programtermékek sokkal elfogadhatóbban válaszolnak az egyedi felhasználó igényeire, mint az előre programozott klasszikus programok, s ennek köszönhetően ez a módszer a gyakorlatban is tényleges felhasználást nyert. Különösen érvényes ez a személyi és a professzionális számítógépek operációs rendszereire, ahol az egyes funkciókat igénytáblázat (menü) alapján maga a felhasználó választja ki.

Sok előnye ellenére, a módszernek hátrányai is vannak. Ezek közül egyik legfontosabb az, hogy az egyedi felhasználónak saját feladatát mint az általános feladat sajátos esetét kell megfogalmaznia, ami sokszor nem is olyan egyszerű. Felmerül a hatékonyság kérdése is: a generált programtermékek elkerülhetetlenül magukban foglalnak sok olyan funkciót is, amelyekre az egyedi felhasználónak talán nincs is szüksége. A "dedikált" szakterület is állandó fejlődésben van, s az egyedi felhasználónak esetleg olyan funkcióra is szüksége lehet, amelyet nem lehetett előre figyelembe venni. Ez esetleg oda vezethet, hogy az egész rendszert felül kell vizsgálni, s újra csak a számítástechnikai szakembereknek kell közbelépni.

Folynak kutatások az evolutív programozás irányában is, ahol a programterméket egy — a számítógéprendszerben eleve meglévő — alapprogram lépésenkénti átalakítása útján nyernék, igen magas szintű interaktív környezetben.

A jelenlegi, negyedik generációs számítógépek, úgy tűnik, meg sem közelítik azt az ideális célkitűzést, hogy "felhasználó-orientált", s nem számítástechnikai képzettségű felhasználók által is könnyen kezelhetőek legyenek. Ma még bármely feladat megoldását előre kell programozni. Az eszményi lehetőség az volna, ha egy-egy feladat megfelelő definíciójának alapján hibátlan és hatékony programokat automatikusan lehetne előállítani. A feladatmegoldás körülbelül a következő módon kellene hogy megvalósuljon:

A felhasználó nem számítástechnikai szakember, de alapos ismerője a szakterületnek, amelyen az alkalmazás történik. A számítógéprendszerrel egy igen magas szintű terminálon keresztül van kapcsolatban, a dialógust valamilyen természetes nyelven folytatja. Ezen keresztül kéri a rendszer a megoldandó feladat pontos, koherens leírását, esetleg annak sajátos, várható megoldásait (a már létező tapasztalat alapján). A leírás, valamint a rendszerbe már eleve beépített tudásbázis alapján a számítógép maga generálja a feladatra szabott, elfogadhatóan hatékony megoldást, miután annak helyességéről saját maga előzőleg már megbizonyosodott.

Programozási nyelvek

Anthony Ralston, neves számítástechnikai szakember a következőket írja egyik munkájának előszavában. Az egyetemi karokra jelentkezőktől rendszerint megkérdezik, milyen idegen nyelveket ismernek. Egyre gyakoribb az ilyen típusú válasz: francia, német, FORTRAN, ALGOL. Szerző szerint ez egyáltalán nem komolytalankodás. Modern életünk szinte minden területén megjelentek a számítógépek, s ezekkel csak egy-egy ilyen speciális "idegen" nyelvvél teremthetünk kapcsolatot (legalábbis még egy jó ideig!).

A mai értelemben vett számítógépek előtt a kalkulátorokat, de még a legelső számítógépeket is, minden egyes művelet végrehajtása előtt vezérlőgombokkal vagy külső adathordozóval (lyukkártya, lyukszalag stb.) kellett irányítani. Ez a módszer azonban éppen az ilyenfajta gépekben rejlő legnagyobb lehetőséget, a nagy műveleti sebességet hagyta kihasználatlanul. Neumann János korszakalkotó ötlete volt, hogy a számítógépek által elvégzendő műveleteket kódolt formában már előre elhelyezzék a gép tárában (memóriájában), s azokat éppen úgy lehessen hívni onnan, mint a közönséges adatokat. A számítógép áramkörei viszont kizárólag az elektronikus jelek formájában megjelenő, kettes számrendszerben írott számokat ismerik fel. A végrehajtandó programot tehát végső soron ilyen számok sorozatából kell felépíteni. Ezt a formát nevezik gépi kódnak.

Programot lehet közvetlenül gépi kódban is írni, ugyanis a számítógépek utasítás-készletének ismeretében gyakorlatilag bármely algoritmus programja elkészíthető és tárolható. Ez a módszer azonban igen nehézkes — még egy rövid program elkészítésének is csekély a valószínűsége —, s szinte használhatatlan azok számára, akik a számítógépet csak mint eszközt használják. (A gépi kódok felismerése és megkülönböztetése a számítógépnek nem okoz nehézséget, az emberi agy és szemek képességei azonban nem jól érvényesülnek az ilyen feladatoknál.)

Az 1940-es évek végén és az 1950-es évek elején a gépi kódú programozás nehézségének csökkentésére a programozók kitalálták az asszemblér kódnak nevezett jelölésrendszert. Ebben a gépi utasítások számkódja helyett szavakat vagy rövidítéseket használtak, mint ADD (képezd összegét), SUB (képezd különbségét), MOVE (helyezd át) stb. A memória elemeire címük helyett szimbolikus nevekkal hivatkoztak. Az asszemblér kódról a gépi kódra való fordítást eleinte kézzel végezték, táblázatok, diagramok segítségével, majd e folyamat gépesíthetőségének felismerésével előállították a fordítás elvégzésére szolgáló, asszemblérnek nevezett programokat. A programozás még így is alapos ismereteket követelt a számítógép konkrét felépítéséről és működéséről, és szó sem lehetett ugyanannak a programnak különböző gépeken történő végrehajtásáról.

A legtöbb asszemblér nyelv vezérlőszerkezete kezdetleges, és ami a legfontosabb, jóval közelebb áll a gép nyelvéhez, mint a felhasználóhoz. Sokkal kézenfekvőbb lenne, ha a felhasználók saját tevékenységi területükre jellemző fogalmakkal, formalizmusokkal írhatnák le programjaikat, majd azoknak gépi kódra való átalakítását szintén a számítógépre bízhatnák. Így alakultak ki a (magas szintű) programozási nyelvek mint az ember és a számítógép közötti kommunikáció alapvető eszközei. Ezeknek több előnye is van az alacsonyabb szintű ábrázolásokkal szemben. Mivel egy utasítás számos gépi kódú utasításnak felel meg, a programok rövidülnek, csökken a hibák lehetősége, a programok olvashatóbbak, áttekinthetőbbek lesznek; a különböző nyelveket szabványosították, így a programtermékek közvetlenül vagy minimális módosításokkal átvihetők lettek egyik gépről a másikra, megteremtődött a "gépfüggetlenség" lehetősége. A gépi nyelvre fordítást elvégző fordítóprogramok (vagy értelmezőprogramok) nagy segítséget nyújtanak a programozási hibák felderítésében, a programok "belövésében", s ezáltal jelentősen megnövelik a programozás hatékonyságát.

Az első jelentősebb programozási nyelvek az ötvenes években jelentek meg. A legelső próbálkozás Konrad Zuse német matematikus nevéhez fűződik, aki feltalált egy jelölésrendszert, amelyet Plan-Kalkülnek nevezett el. Zuse papírra írta programjait — köztük olyat is, amely sakklépéseket szimulált! —, sajnos azonban nem alkalmazta gépre a nyelvet. Elgondolásai közül azonban sokat átvettek a modern nyelvek.

Az első tényleges programozási nyelvek közül a FORTRAN említhető, amelyet John Backus és kollégái fejlesztettek ki az IBM-nél 1945—1957 között (FORTRAN = FORmula TRANslation) és tudományos-technikai számítási műveletek elvégzésére szántak. Nagyjából ugyanebben az időben az Univac-cégnél Hopper és munkatársai kifejlesztették az üzleti tevékenység adatainak kezelésére a Flow-Matic nyelvet, amely az 1960-ban megjelent COBOL (Common Business-Oriented Language) előfutárának tekinthető. A fejlődés legfontosabb időszakának 1958—1959-et tartják, mert ekkor adták közre az IAL- (International Algebraic Language) jelentést, amely ALGOL-58 néven

maradt közismert, s ekkor került ismertetésre a Backus-Naur formalizmus, amely meghatározó jelentőséggel bír a fordítóprogramok elméletének és gyakorlatának kialakulásában. Az 1960—1970 közötti periódus jelenti a programozási nyelvek általános elterjedését, s ugyancsak ebben a periódusban kezdődnek meg azok a törekvések, hogy a programozási nyelvek szabványosításával egységes keretet biztosítsanak a különböző gépeken történő megvalósításainak.

Általános értelmezésben programozási nyelvnek tekinthető bármely olyan kommunikációs eszköz, amely alkalmas adatstruktúrák és az azokkal dolgozó algoritmusok egyértelmű leírására. Fontos különbséget tenni azonban a programozási nyelv és annak megvalósítása között. Maga a programozási nyelv tulajdonképpen egy jelölésrendszer, szabályok halmaza, amely meghatározza egy programnak szabványos elemekből történő felépítését, formai kötöttségeit. A nyelv megvalósítása viszont az a program, amely a formai szabályok megsértéseit felfedezi és kiszűri, a helyes magas szintű jelöléseket pedig gépi utasítások sorozatává alakítja. A programozási nyelvek megvalósításának két fő fajtája létezik: a fordítóprogramok és az értelmezőprogramok. A fordítóprogram a magas szintű program teljes szövegét lefordítja egyetlen folyamatos eljárás keretében, így egy teljes gépi kódú programot hoz létre, amely azután a fordítóprogramtól függetlenül (többször is) végrehajtható. Egy fordítóprogrammal megvalósított programozási nyelvben a munka általában három (esetleg több) lépésből áll: először előállítják a program szövegét (= forrásprogram) egy szövegszerkesztő vagy szövegfeldolgozó program segítségével, azután ezt a szöveget lefordítatják a fordítóprogrammal, majd az eredményül kapott célprogramot végrehajtatják a számítógéppel. Ezzel szemben az értelmezőprogram felhasználása e fő lépések kevert végrehajtását jelenti. Az értelmezőprogram a programozási nyelvre jellemző egy-egy utasítást dolgoz fel egyszerre, kezdve az utasítás előállításával, folytatva annak elemzésével, lefordításával és a célutasítás közvetlen végrehajtásával. A különbség érzékeltetésére talán egy irodalmi mű fordítója és egy társalgási tolmács tevékenysége közötti különbségre lehetne hivatkozni. Elvileg bármely programozási nyelv megvalósítható mind fordítással, mind értelmezéssel, mindkét módszernek megvannak a maga előnyei és hátrányai. A fordítás fő előnye a sebesség és hatékonyság: egyszeri fordítás után akárhány végrehajtás következhet (a végleges programokat tulajdonképpen felesleges és költséges újra és újra lefordítani). Az értelmezés viszont sokkal közelebb áll a párbeszédhez, a felhasználó számára kényelmesebb formához: egy-egy program részeit meg lehet írni, majd rögtön ki is próbálni, az esetleges hibákat helyben kijavítani, majd újra végrehajtani. Mint az élet megannyi más területén, itt is az értelmes kompromisszum jelenti az arany középutat. A programok írásának, belövésének kezdeti fázisában előnyben lehet részesíteni az értelmezést, majd a végleges, hibátlan formájú forrásprogramból egyszeri fordítás útján kész célprogramot előállítani.

A magas szintű programozási nyelveken megírt programok még nagyon távol vannak attól a formától, amelyen ténylegesen végrehajthatók egy adott számítógépen. Függetlenül attól, hogy fordítást vagy értelmezést alkalmaznak, a forrásprogramnak egy egész sor átalakításon kell átesnie. Ennek a komplex átalakítási folyamatnak legfontosabb alapelve a "jelentésmegmaradás" tétele, amely azt jelenti, hogy egy program jelentése nem változhat meg a rajta végzett átalakítások során. Más szóval, a forrásprogram jelentése teljes egészében és változatlanul kell hogy visszatükröződjön a tárgyprogram (célprogram) jelentésében.

A természetes, illetve programozási nyelvek összehasonlítása igen érdekes közös vonásokat, illetve különbségeket tár az érdeklődők elé.

Míg a természetes nyelvekre a közösségi fejlesztés és felhasználás jellemző, a programozási nyelveket szűk körű tervező-fejlesztő csoport állítja elő a felhasználók számára. Az előbbieket ezért dinamikusan fejlődhetnek, rögtön alkalmazkodhatnak az emberi tevékenység egy-egy újabb területéhez új szavak létrejöttével és elterjedésével. Ezzel szemben a programozási nyelvek jóval statikusabbak, nem változnak meg egy-egy újabb programozási kategória bevezetésével.

Alapvető különbséget mutat az összehasonlítás a kétértelműség szempontjából is. Vegyük például a következő magyar mondatot: "Látta az embert a parkban a távcsővel." Ez a mondat, környezetéből kiragadva, semmi pontosat nem közöl, minden olvasó saját fantáziája szerint értelmezheti: ki volt a parkban, a megfigyelő vagy a megfigyelt? Kinél volt a távcső, a megfigyelőnél vagy a megfigyeltnél? A természetes nyelvekben ilyen kétértelműségek gyakran előfordulnak, de beszédhelyzetben könnyen feloldódnak — sőt színesebbé, gazdagabbá teszik az illető nyelvet. A programozási nyelvek tervezői viszont mindig egyértelműségekre törekednek: a számítógép mindig ugyanazt az értelmezést kell hogy válassza egy-egy adott forrásnyelvi formának.

Másik lényeges különbség a szövegekörnyezettől való függőség, illetve függetlenség. Nézzük a következő két mondatot:

A kislány gyöngyöt fűz.

Szomorúan bólogat a fűz.

E mondatokban a fűz — a két azonos szóalak — mást és mást jelent, s a helyes értelmezés csupán a teljes mondatból szűrhető ki. A természetes nyelvek tehát többségükben környezetfüggőek. Ezzel szemben a programozási nyelvek környezetfüggetlenek, bár ez nem szükségszerű tulajdonságuk.

Az összehasonlítás számos közös vonást, hasonlóságot is kimutat. A bevezetőben említett párhuzam helyességének alapvető bizonyítéka az, hogy a programozási és a természetes nyelvek egymáshoz hasonló fogalmakkal írhatók le: ábécé, szókincs, szintaxis és szemantika. Régi törekvése volt a nyelvészeknek, hogy ezeket a fogalmakat matematikai modellek alapján formalizálják. Ez a törekvés nem volt egészen sikeres a természetes nyelveknél, annál hasznosabbnak bizonyult viszont a programozási nyelvek leírásában és gyakorlati megvalósításában, ugyanis ezek — pontos szerkezetük és egyértelműségük révén — egyenesen kínálkoznak a formalizálásra.

Akárcsak a természetes nyelvek esetében a hangkészlet — és a megfelelő betűkészlet —, a programozási nyelveknél egy ábécé a kiindulópont: mindazon jelek összessége, amelyekből a nyelv más egységei felépíthetők. A nyelv szavait, mondatait az ábécéhez tartozó jelekből építik fel. Azok a szabályok, amelyek segítségével a nyelv mondatai egyértelműen levezethetők (generálhatók), az illető nyelv szintaxisát alkotják. Szókincsét tekintve a programozási nyelvek túlnyomó többsége tulajdonképpen az angol nyelv egy igen leszűkített változata, szintaxisuk viszont már egyedi, alapvetően jellemző. A szintaxis formális leírására egész tudományág alakult ki, amelynek eredményeként a fordítóprogramokat ma már egyre hatékonyabb és biztonságosabb formában, sokszor automatizált formában lehet előállítani.

A számítógépes nyelvészet szempontjából a nyelvtan az adott programozási nyelv mondatainak generálására vonatkozik, a szintaxis viszont a nyelvhez tartozó mondatok felismerését segíti elő. Ily módon kiegészítő viszony áll fenn a nyelvtan törvényei és a szintaxis szabályai között.

A programozási nyelvek definíciójának szintaxisbeli leírása alapvető követelmény, de ez önmagában még nem elégséges: ezzel csak a nyelvhez tartozó mondatok szerkezetét lehet leírni, de nem kapcsolható hozzájuk jelentéstartalom. A megfelelő jelentésnek a szerkezethez való hozzárendelésével a szemantika foglalkozik, akárcsak a természetes nyelveknél — azzal a különbséggel, hogy ez utóbbiaknál a szemantika nagyrészt intuitív jellegű. Nézzük a következő mondatot: "A szorgalmas könyv kiolvasta az érdekes tanulót." Ez a magyar mondat szintaktikailag helyes, szemantikailag viszont teljesen értelmetlen. Még ha az adott természetes nyelv értelmező szótára meg is adja a nyelvhez tartozó jelsorozatok (szavak) értelmét, egy-egy mondat szemantikailag teljesen intuitív jellegű. Természetesen a programozási nyelvek vonatkozásában szó sem lehet ilyen intuitív értelmezésről, minden egyes szintaktikailag helyes szerkezethez pontos értelemkapcsolás szükséges.

A szemantika leírására — elég bonyolult probléma lévén — még folynak az elméleti és gyakorlati kutatások. A legegyszerűbb mód valamilyen természetes nyelven megadni a szabályokat az összes előfordulható szerkezeti formához. Például az ADD A

TO B GIVING C szerkezet (utasítás) hozzátartozik a COBOL programozási nyelvhez (kielégíti annak összes idevágó szintaktikai szabályát), szemantikája pedig körülbelül a következő módon írható le: az A változó tartalma (értéke) hozzáadódik a B változó tartalmához, az eredmény pedig a C változóban raktározódik el; az A és B változók tartalma változatlan marad. Ezeket a jelentésleírásokat mind be kell építeni a fordítóprogramba, illetve értelmezőprogramba, mert csak így biztosítható a már említett jelentésmegmaradás elvének érvényesülése. Innen következik, hogy a jelentés megmaradásának alapján a szemantika megadásának módját maga az illető nyelv fordítóprogramja biztosítja. Ha ismert a generált célnyelv szemantikája, akkor adottnak tekinthető a forrásnyelvi szerkezet szemantikája is. Igen érdekes kutatási terület napjainkban is a szemantika leírásának formalizálása.

A programozási nyelveket tanulóknak talán a szemantikai leírás jelenti a legnagyobb nehézséget, ugyanis csak ezek alapos ismerete révén lehet ténylegesen működő programokat előállítani.

Az összehasonlítás egyik érdekes szempontja éppen a nyelvek megtanulásának és gyakorlati felhasználásának módjára vonatkozik. Amikor például angolul tanulunk, a nyelv elsajátításához és alkalmazásához nem szükséges (bár nem is árt!) az angol embert, társadalmat külön tanulmányozni és megismerni: a nyelv megtanulása után könnyen kapcsolatot teremthetünk bárkivel, aki angolul (is) beszél. A programozási nyelvek elsajátítása nem ennyire független a gyakorlati felhasználástól. Egy programozási nyelvet csak úgy alkalmazhatunk ténylegesen, ha az azt megvalósító számítógéprendszerre vonatkozó bizonyos ismereteknek is birtokában vagyunk.

Törpe — óriás lehetőségekkel: a mikroprocesszor

A hatvanas évek közepe táján a félvezető-technológia minőségi változásokon ment keresztül. Sikerült ugyanis olyan új eljárásokat találni, amelyekkel egy szilíciumlapkán már nemcsak egyetlen tranzisztort vagy diódát, hanem teljes áramkört lehetett kialakítani (egy áramkör általában tranzisztorokból, diódákból, ellenállásokból és kondenzátorokból áll). Az egyetlen tokban elhelyezkedő áramkört "integrált" áramkörnek (ismert és elterjedt angol rövidítéssel IC-nek) nevezték el. Az első ilyen IC-k néhányszor tíz áramköri elemet tartalmaztak, de nagyon rövid időközönként egyre komplexebb és finomabb felépítésű, bonyolultabb funkciójú integrált áramkörök jelentek meg. Az elmúlt két és fél évtizedben az egy tokban integrált elemek száma átlagosan ötvenként megtízszereződött. Ezzel együtt az egy áramköri elemre jutó költség is rohamosan csökkent, mégpedig szinte ugyanolyan arányban, ahogy nőtt a elemek sűrűsége. Az első nagymértékben integrált áramkörök a hatvanas évek végén, a hetvenes évek elején jelentek meg, s számítástechnikai vonatkozásban először a memóriagyártásban kaptak szerepet. Egyre-másra jelentek meg az egyre nagyobb kapacitású memóriagységek, hogy napjainkra elérjék a 64 kB, sőt 256 kB (kB = kilobájt = 1024 bájt) kapacitást. Önként kínálkozott tehát a gondolat, hogy egy ilyen roppant parányi méretű áramkör akár egy számítógép központi vezérlőegységének feladatait is elláthatná. Ez a gondolat, majd valóság szinte beláthatatlan jelentőségű eseménynek bizonyult a számítástechnika további fejlődésében. Manapság, amikor a munkahelyeken, az iskolákban, sőt a háztartásokban is egyre megszokottabb jelenség a személyi számítógép, szinte hihetetlennek tűnik, hogy az első mikroprocesszor csak 1971-ben született! Ebben az évben jelentette az Intel-cég a 4004-es típusjelzésű első mikroprocesszorát, s még ebben az évben kihozta az első 8 bites mikroprocesszort is, a 8008-as típusjelzésűt.

Az első mikroprocesszorok speciális felhasználói rendelésekre készültek, olyan berendezések vezérlésére, amelyeknél jelentős méretcsökkenést kellett megvalósítani. Ilyen speciális terület volt például az adatforgalmat lebonyolító berendezések (terminálok) vezérlése. Csakhamar a mikroprocesszoroknak félvezetős memóriával való kiegészítése révén valódi mikroszámítógépek alakultak ki, amelyek az adatforgalom lebonyolítása

mellett már önmaguk is képesek voltak egyes, eddig csak a "valódi" számítógépek által végzett feladatok megoldására is. Megjelentek tehát az "intelligens" terminálok.

A nagy integráltságú processzorok — egyre nagyobb megbízhatóságuk és viszonylagos olcsóságuk miatt — csakhamar a folyamatirányításban is fontos szerepet kezdtek játszani. Ezek a vezérlőberendezések eleinte alapvetően hagyományos gépvezérlői (indító, mozgató, kapcsoló, időzítő stb.) funkciókat láttak el, biztonságtechnikai, balesetvédelmi feladatokat hajtottak végre, s ugyanakkor megteremtették a kezelővel való kapcsolatot. A korszerű mikroprocesszorok, tárolóelemek, logikai és aritmetikai, sőt berendezésorientált áramkörök lehetővé teszik, hogy az új vezérlőberendezések bonyolultabb feladatokat is gyorsabban, pontosabban és biztonságosabban lássanak el, mint a klasszikus vezérlőberendezések. A mikroprocesszoros technika nagy előnye, hogy a vele épített vezérlők nemcsak fizikailag, hanem funkcionálisan is modulárisak lehetnek. Ez egyszerűbbé teszi a karbantartást, továbbá a felhasználói igények szerint bővíthető vagy módosítható kiépítettségű, általánosabban használható vezérléseket eredményez egy adott területen. A mikroprocesszoros vezérlés másik előnyét jelenti, hogy a hibás működések korai felismerhetőségével igen nagy károk kerülhetők el. Így például a forgácsolás során rezgési, hang-, erőnyomaték- stb. mérések adataiból következtetni lehet a szerszám kopottságára, a megmunkálendő anyag pillanatnyi jellemzőire. Megjósolható és jelezhető a szerszám törésének időpontja, így annak bekövetkezése előtt az egész berendezés leállítható.

Kezdetben, az első generációs mikroprocesszorok megjelenésekor még nem fűttek sok reményt azok önálló számítástechnikai alkalmazásához, a cél inkább csak az volt, hogy fellendítse a félvezetős memóriák, illetve az intelligens terminálok piacát. Amikor azonban 1973-ban az Intel bejelentette a 8080-as mikroprocesszort, már széles körökben felismerték a benne rejlő hatalmas lehetőségeket, s megkezdődött a tömeggyártás. Az 1970-es évek derekán az alacsony árak révén már magánszemélyek részére is lehetővé tette az ilyen alkatrészek beszerzését, s így új fogalom indult el a széles körű elterjedés, a köznapivá válás útján: a személyi számítógép.

A mikroprocesszorok piacra kerülésével igen alacsony áron lehetővé vált "számítógépek" házilagos építése és üzemeltetése, bár a hobbiból saját számítógépre vágyók még egyedi tervezésre és kivitelezésre kényszerültek. Az első, kereskedelmi forgalmazott, személyi használatra szánt mikrogépet, az ALTAIR-t 400 dolláros áron 1975-ben jelentették be. Az alkatrészként vagy szerelt kártyák formájában kínált személyi számítógépeket 1977-től fokozatosan felváltották a dobozolt eszközök, amelyek a hobbi célú alkalmazási igényeket kielégítő, billentyűzettel, megjelenítővel (rendszerint közönséges TV-készülék), kazettás háttérmemóriával (közönséges kazettafon) ellátott, programozható mikrogépek voltak. A legelső személyi számítógépeket készítők elsősorban amolyan családi játékszernek szánták. Úgy is lehetne mondani, hogy a személyi számítógép a tévéjáték és a programozható kézi számológép házasságából született: az előbbtől a képernyőt, az utóbbitól az adatfeldolgozási képességet örökölte.

Amikor a mai nagy teljesítményű mikrogépek "ősei" megjelentek a számítógépiacra, a programfejlesztéshez rendelkezésre álló eszközök még igen szegényesek voltak, majdnem csak gépi kódú programozást tettek lehetővé (Dijkstra, a strukturált programozás módszertanának kidolgozója szerint, a hardver technológiai csodája mellett a szoftver az ötvenes évek szintjére süllyedt!). Ennek nyilvánvaló okai is voltak. Az egyik ok a korai gépek hardver lehetőségeiben rejtett. Az első mikroprocesszoros rendszerek, szinte kivétel nélkül, közvetlenül csak a mikroprocesszor gépi kódjában voltak programozhatók, mivel az akkori memóriaméretetek nem engedhették meg, hogy hatékony fordítóprogramok, esetleg operációs rendszerekhez hasonló nagy méretű programok elhelyezhetők legyenek. A korai nagygépes nyelvekkel szemben viszont egy már használható BASIC értelmezőprogramot csakhamar meg lehetett valósítani. Ezek az értelmezőprogramok hamarosan el is készültek, s ennek eredményeképpen valóságos "szoftverrobbanás" következett be. A korai fejlesztők elsősorban játékprogramokat állítottak elő, de csakhamar megjelentek az első üzleti célú szoftvertermékek is. A BASIC mint a mikrogépek

"anyanyelve" minden kortársát kiszorítva (legalábbis az átlagos felhasználó számára), igen hamar egyeduralgódóvá vált.

A pillanatnyi megtorpanást gyors fejlődés követte. A memóriaméretetek gyors növekedése a professzionális célú gépek megjelenését is jelentette, amelyek a számítógépes játékokon túl, "komoly" területen való felhasználást is biztosítottak, elsősorban kisebb volumenű pénzügyi-gazdasági feladatok megoldására, hagyományos műszaki-tudományos alkalmazásokra, oktatási célokra stb. A mikroelektronika nyújtotta lehetőségeket kihasználva, e kis gépek csakhamar olyan kapacitású memóriával rendelkezettek, amelyet nemrég még az úgynevezett "nagy" számítógépek is megirigyelhettek. A személyi számítógépekhez illeszthető eszközök köre szinte napról napra bővült. Háttérmemóriaként már nemcsak kazettákat, hanem különféle mágneslemezeket is használhattak, kijelzőként színes televíziót, miniatűr grafikus nyomtatókat is kapcsolhattak. A megváltozott körülmények megkívánták az operációs rendszerek megjelenését is, hogy figyelemmel kísérjék a programok indítását, futtatását, a különféle kapcsolt eszközök működését stb. Szövegszerkesztő programok jelentek meg a forrásnyelvű programok előállítására, karbantartására és archiválására, a programbelövést pedig hatékony nyomkövető programok segítették. Az 1980-as évek elején megjelentek a 16 bites, majd 32 bites mikrogépek, amelyek kiépítettségük, teljesítményük, széles körű programozhatóságuk révén (BASIC, FORTRAN, Pascal, COBOL stb. fordítóprogramokkal) a hagyományosan miniszámítógépeknek nevezett kategóriát is elérték, sőt meghaladták, s a tíz-tizenöt évvel ezelőtti osztályokba sorolást (kis, közepes és nagy számítógépek) teljesen érvényen kívül helyezték. A legfejlettebb mikrogépeket úgynevezett adatátviteli csatolóegységgel is kiegészítették. Ez lehetővé tette, hogy a nagy számítógépek végberendezéseként — termináljaként — használhassák őket: telefonvonalon keresztül csatlakoztathatók a nagy számítógéprendszerekhez, s általuk akár dolgozószobából is hozzáférhetővé teszik a nagy információs adatbázisokat.

Az igen hatékony mikro-operációs rendszereknek köszönhetően több mikrogép helyi hálózatokba is szervezhető. Egy-egy ilyen hálózat egy vállalat, intézmény információs kiszolgálását sokszor sokkal hatékonyabban valósítja meg, mint a hagyományos nagyszámítógép.

Kedvező tulajdonságai révén a mikroszámítógép olyan területekre is behatolt, ahol néhány évvel ezelőtt elképzelhetetlen lett volna számítógép jelenléte (mint ismeretes, a hagyományos számítógépek túlnyomó többsége csak klimatizált környezetben "érezte jól" magát). Egyik ilyen legfontosabb terület az ipari és űrhajózási robottechnika. Ezekkel a robotokkal olyan munkák is elvégezethetők, amelyeknek környezete vagy káros, vagy egyenesen elképzelhetetlen — egyelőre — az ember számára. Ma már az ember nélküli ipari szerelőcsarnokok vagy a távoli bolygók felé repülő űrhajók nem is jelentenek különösebb szenzációt.

Napjaink 32 bites mikroprocesszora teljesítőképességét tekintve beérte az 1970-es évek második felének nagyszámítógépét. Tulajdonképpen a mikroprocesszor megtette mindazokat a fejlődési lépéseket, amelyek a klasszikus számítógépeknél is megfigyelhetők voltak, csak hogy még annál is elképesztőbb sebességgel: a 20 éves mikroprocesszor története nagyjából megismételte klasszikus elődje több mint 40 éves történetét. Technológiáját tekintve pedig eljutott arra a szintre, amellyel már sokkal többre képes, mint a "rég" architektúrák egyszerű utánzása. A mikroprocesszor "felnőtté" vált, amikor már saját maga alakíthatja jövőjét!

Operációs rendszerek, programozási környezetek

Az első programozható számítógépek megjelenésével és munkába állításával az elvégezhető hadászati, illetve tudományos-műszaki számítások bonyolultsága és terjedelme hirtelen megnőtt. Ebben az időszakban csak az látszott lényegesnek, hogy azokat a feladatokat, amelyeket addig gyakorlatilag megoldhatatlannak, kivihetetlennek tartottak,

most bármelyik szakember elvégezhetette, ha a megfelelő berendezéshez hozzáférhetett. Senkit sem érdekelt valójában az, hogy az eredményeket mennyi idő alatt és milyen költségen lehetett előállítani. Maga a számítógéprendszer is igen szerény kiépítettségű volt: a központi feldolgozó- és vezérlőegységen, illetve memórián kívül az adatforgalom számára egy-egy egyszerű adatbeviteli, illetve adatkiviteli egységet tartalmazott. Erre a periódusra a számítógépek központiegység-centrikus szervezése volt jellemző. Az ilyen szervezés alapvető jellegzetessége az, hogy a közvetlen kapcsolat a rendszeren belül csak a központi egység és a memória, illetve a központi egység és periférikus egységek között létezett. Az első és részben a második generációs számítógépek idején a felhasználást kényelmesebbé és gazdaságossá tevő eszközök csak a tulajdonképpeni működést biztosító eszközökre korlátozódtak: az üres gépbe való kezdeti programbevitelre, a memória tartalmának módosítására, a be-, illetve kiviteli egységek vezérlésére. Ezeknél a számítógépeknél minden komponens a központi vezérlőegység vezérelt. Ebből egyenesen következett, hogy az egységes rendszer sebessége mindig az éppen működő leglassabb berendezés sebességére csökkent, pedig egyetlen adatátviteli művelet időperiódusa alatt a központi egység több ezer aritmetikai vagy logikai műveletet is elvégezhetett volna. E hátrányt úgy próbálták kiküszöbölni, hogy megpróbálták a központi egységet az adatátviteli periódusra felszabadítani.

A számítógép-tervezésben minőségi ugrást jelentett a "megszakítás" fogalmának bevezetése az ötvenes évek végén. Ennek a módszernek az a központi elgondolása, hogy a rendszer mindegyik berendezését a központi vezérlőegység indítja be, de ezután már önállóan dolgoznak tovább, felszabadítva a vezérlőegységet más feladatok számára. Ezek a berendezések, mikor munkájukat befejezték, vagy valamilyen előre nem várt esemény következett be (például az adathordozók hibája) és beavatkozásra volt szükség, "megszakítás" jelzést küldenek, amely leállítja a központi vezérlőegység pillanatnyi munkáját, hogy kizárólag a megszakítást jelzett eseményt dolgozza fel. Az időben eltol, párhuzamos működések koordinálása már komplikáltabb vezérlést igényelt. Bár az operációs rendszer fogalma még nem létezett ugyan, csírái viszont már jelentkeztek.

Az emberek azonban csakhamar kezdtek ráébredni a számítógépekben rejlő nagy-szerű lehetőségekre. Használatuk egyre terjedt, lényeges változásokat idézve elő a potenciális felhasználók elvárásaiban. Alapvető igényként jelentkezett egy olyan felhasználási mód, amelyben az algoritmusok tervezését, kódolását és végrehajtását a legegyszerűbbé, leggazdaságosabbá lehet tenni. Az egyre kényelmesebb felhasználást a programozási nyelvek szolgáltatták, a számukra készült fordítóprogramok, illetve értelmezőprogramok, a forrásprogramok előállítását, szerkesztését, módosítását, illetve nyilvántartását végző szerkesztőprogramok, programbetöltő programok, adatállomány-kezelő programok stb. A számítógéprendszerekben számos olyan funkció jelent meg, amelynek vezérlését már nem volt célszerű a felhasználóra, illetve annak programjára bízni. Sőt olyan feladatok is jelentkeztek, amelyeket nem is lehetett a felhasználói programok keretében megoldani. Mindezen feladatok megoldására a számítógépet egy mindig jelenlevő programmal, programrendszerrel szerelték fel. Ezt a programrendszert nevezik operációs rendszernek, amely azt a feladatot is magára vállalja, hogy a különböző erőforrások működését úgy hangolja egybe, hogy a rendszer mint egész a leggazdaságosabban működjék.

A gazdaságosság irányában végzett kutatások és megvalósítások tulajdonképpen az operációs rendszerek fejlődésének irányát és minőségi szakaszait is meghatározták.

Az operációs rendszerrel ellátott számítógépet egy, a hágyamához hasonló réteges rendszerként foghatjuk fel. E rendszer középpontjában található az összes elektronikus, mágneses, gépi berendezések, amelyek együttesen alkotják a hardvert (ez a tulajdonképpeni "üres" számítógép). Erre a magra épülnek azok a legelemibb programozott funkciók, amelyek a gépi berendezések működését biztosítják. A mag és ez az első programozott réteg tulajdonképpen már egy "új" gépet képvisel. Újabb réteget alkotnak azok a programozott eszközök, amelyek az első két szinten helyet foglaló (gépi, illetve programozott) erőforrásokat valamilyen szervezési elvnek megfelelően irányítják. Hasonló módon építhetők erre a szerkezetre újabb és újabb rétegek, s így a felhasználó

mindig csak a külső rétegen keresztül jut kapcsolatba a virtuális gépet képező számítógéprendszerrel.

Az első tényleges operációs rendszerekben alkalmazott stratégia azt tűzte ki célul, hogy a rendszer működése közben a hozzá tartozó gépi erőforrások közül minél kevesebb legyen kihasználatlan, azaz lehetőleg mindegyik állandóan dolgozzék. Ez a stratégia az alapja az úgynevezett kötegelt üzemmódban dolgozó rendszereknek, ahol az elvégzendő munka előre rögzített feladatcsomagok formájában érkezett a számítógéphez, s az eredmények szolgáltatása sem volt időhöz kötve. Mivel a feladatok végrehajtására nem egy konkrét időpont, hanem napos nagyságrendű időintervallum állt rendelkezésre, elvileg lehetőség volt a feladatok olyan csoportosítására, amely biztosította az erőforrások optimális szervezését és kihasználását. Ilyen üzemmódban dolgoztak a hatvanas évek elején-közepén megalakult számítóközpontok, ahova a külső felhasználók leadták munkáikat, majd másnap vagy esetleg napok múlva jelentkeztek az eredménylisták után. A centralizált feldolgozás egyik nagy hátránya a nagy rendszerek merevsége volt. Mivel a központosított számítógéprendszer egy nagy szervezet valamennyi adatfeldolgozási feladatát meg kell hogy oldja, természetesen nem alkalmazkodhat minden egyes felhasználó igényeihez. A kötegelt feldolgozás természetéből következik a viszonylag kis feldolgozási sebesség, amit az egyes feldolgozási szakaszok közötti kényszerű várakozások okoznak. A feladatok jelentkezésétől az eredmények kézhezvételéig napok, rosszabb esetben hetek telhettek el.

Az operatív memóriák fizikai növekedésével ez a fontos erőforrás egyre kevésbé volt kihasználva egyetlen felhasználói program futtatásával. Egy-egy rövidebb, de időigényesebb program hosszú ideig a memória nagy részét tétlenségre kárhoztatta. Mivel a számítási műveleteknek rendszerint be kellett várniuk a szükséges adatokat, a központi vezérlőegység is többet állt, mint dolgozott. Csakhamar felmerült az az igény, hogy egyszerre több, csak a memória konkrét mérete által korlátozott számú felhasználói program állhasson feldolgozás alatt. Így a központi egység mindig azzal a programmal tudna foglalkozni, amely éppen nem várakozik adatbevitelre vagy adatkivitelre. Az ilyen lehetőségeket biztosító rendszereket hívták multiprogramozásos kötegelt feldolgozású operációs rendszereknek, s a hatvanas évek közepére voltak jellemzőek.

Bár ez a stratégia a legfontosabb gazdaságossági igényeket elég magas fokon kielégítette, észrevétlenül bevezetett egy olyan hátrányos helyzetet, amely csakhamar igen problematikusá vált: megszüntette a program végrehajtásának idejére a számítógép és a felhasználó közvetlen kapcsolatát — bár ez a kapcsolat a számítógépek kezdeti szakaszában létezett és igen hasznosnak bizonyult. Így az olyan típusú felhasználók (elsősorban kutatók, tervezőmérnökök), akik programjaikat gyakran módosítják, módosítás után az új változatot futtatják, a futtatás eredményeit elemzik, összevetik kísérleti tapasztalataikkal, majd újra módosítanak stb., hamar kiszorultak a gépteremből. Ezeknek a kielégítésére egy olyan új stratégia kialakítása vált szükségessé, amely, ha csak látszólagosan is, de visszaadja a számítógépet felhasználójának. Az új módszer az időosztásos (time-sharing) üzemmód néven vált ismeretessé, alapelve pedig az, hogy a rendszerhez kapcsolt bármelyik felhasználó bármelyik időpillanatban a teljes számítógépes erőforrás-választék felett rendelkezzen. Egyetlen processzort (feldolgozóegységet) tartalmazó rendszerekben ez az állandóan rendelkezésre állás természetesen csak látszólagos, hiszen az egyetlen feldolgozóegység egyszerre csak egy felhasználót tud kiszolgálni. Ha azonban figyelembe vesszük azt, hogy a számítógép feldolgozási sebessége, illetve a felhasználó műveleti és reakcióideje között több nagyságrendű eltérés létezik, bármely felhasználóban azt az érzetet lehet kelteni, hogy kizárólagos üzemeltetője a rendszernek.

Kezdetben ezt a módszert úgy valósították meg, hogy a központi feldolgozóegységet egyenlő időintervallumokra egyszerűen kiosztották a soron következő felhasználó számára. Később a módszert finomították, és a felhasználókat fontosságuk (prioritásuk), valamint a pillanatnyi konkrét helyzet függvényében valamilyen speciális kiszolgálási stratégia szerint szolgálták ki.

Az újra emberközelivé vált számítógép-felhasználásban megint előtérbe került a minél kényelmesebb, emberre szabottabb kommunikáció megteremtése. Az új, párbeszéd (interaktív) üzemmód néven közismert időosztásos operációs rendszer igyekszik lehetővé tenni, hogy a számítógépes munka szempontjából kevésbé képzett vagy valamikor képzett, de már sok mindent elfelejtett felhasználók is könnyen hozzáférjenek a számítógép szolgáltatásaihoz. Ennek érdekében kérdés-válasz rendszerű párbeszéd útján vesznek tudomást a felhasználói igényekről, s kérésre kimerítő információkat szolgáltatnak saját lehetőségeikről. Termináljától függő módon a felhasználó a rendszerrel állandó kapcsolatot tud fenntartani, sokféle rendszerfunkciót tud használni: programot írni és javítani (szövegszerkesztő segítségével), futtatni, a programfutást megszakítani vagy valamilyen esemény bekövetkeztekor a program állapotába beavatkozni, majd a program futását folytatni vagy újraindítani stb.

Rendkívül széles körű elterjedése folytán a számítógép eljutott olyan helyre is, ahol a potenciális felhasználók elsősorban humán műveltségűek (pl. orvosok, nyelvészek, művészek). E területek tanulmányozása során kiderült, hogy az ilyen típusú felhasználók sok esetben szívesebben használnak kevesebbet "tudó", de kényelmesebb kommunikációt biztosító eszközt, mint egy rendkívül hatékony, de számukra túl bonyolult felhasználói kapcsolattal bíró. Az ilyen természetű igények kielégítésére alakultak ki az úgynevezett programozási környezetek. Ezek a rendszerek alapjában véve csupán egyetlen magas szintű programozási nyelv ismeretét kívánják meg a felhasználótól. Ez a nyelv egyes személyi, illetve professzionális számítógépeknél csupán egy sor szuggesztív rajzos ábra felismerésére és helyes kiválasztására redukálódik. A számítógéprendszerrel várt összes szolgáltatást ezen a nyelven keresztül érhetik el a felhasználók anélkül, hogy a gépre vonatkozó konkrét ismeretekre lenne szükségük.

A számítógép felhasználása során kiderült, hogy ezek a fontos gépek közvetlenül, helyileg csak korlátozott mértékben használhatók, mert az általuk feldolgozott adatok forrása a legritkább esetben van a gépteremben és az eredményeket sem használják fel mindig helyben. Ennek következtében a kezdeti időszakban a következő típusú adatforgalom volt nagyjából érvényben: a forráshelyen összegyűjtötték és külső adathordozóra (lyukkártyára, lyukszalagra, mágnesszalagra stb.) rögzítették az adatokat, s ilyen formában szállították a számítógéprendszer géptermebe; kötegelt üzemmódban dolgozták fel az adatokat, majd az eredményeket (legtöbbször papírra nyomtatva) újra a számítógéptől függetlenül szállították rendelkezési helyükre. Ez a módszer igen sok esetben elég nehézkesnek bizonyult, s a számítógéprendszer és a felhasználó fizikai különválasztása nagyon sok — különben elkerülhető — hibás feldolgozásnak, gazdaságtalan erőforrás-kihasználásnak volt forrása. Ezt az elkülönítést akkor lehetett valamennyire kiküszöbölni, amikor először használtak távnyomatot számítógépes perifériaként (az összeköttetést telexvonalak vagy modemek és telefonvonalak szolgáltatták).

Az első lökést az elosztott feldolgozás felé a nagy központi egységre épült számítógéprendszerek decentralizálása adta. A hagyományos rendszerekben az adatforgalmat speciális elektronikus egységekkel bonyolították le. A mikroprocesszorok megjelenésével lehetővé vált olyan egységek előállítása, amelyeket mindig az éppen felmerülő igényeknek megfelelően lehetett programozni. Csakhamar saját memóriával és processzorral rendelkező mikroszámítépek vették át az adatforgalom lebonyolítását, amelyek már önmaguk is képesek voltak egyes, addig kizárólag a számítógép által végzett feladat megoldására (forrásprogram előállítására, adatállomány-nyilvántartásra stb.): megjelentek az úgynevezett "intelligens terminálok". Ezeknek köszönhetően sok másodlagos tevékenységet helyben el lehetett végezni, felszabadítva ezáltal a központi vezérlőegységet, amelyet most már sokkal nagyobb mértékben lehetett kimondottan feldolgozási műveletekre használni. Nagyjából ennek az alapelvnek a kibontakozása vezetett oda, hogy az egyetlen számítógéppel rendelkező rendszert felváltsa egy több kisebb egység összekapcsolása révén nyert hálózat. Az a kézenfekvő gondolat, hogy a számítástechnikai erőforrásokat összekössék egymással s mindig a legalkalmasabb

időpontban és mennyiségben bocsássák a felhasználók rendelkezésére, tulajdonképpen nem is volt olyan forradalmi elgondolás. Hasonló módszert már ismert a technikatörténet: a múlt század végén az elektrotechnikai gépeket és erőforrásokat kapcsolták össze, s az így létrejövő, különböző rendeltetésű és kiterjedésű hálózatokat a kis és nagy fogyasztók saját berendezéseik segítségével használhatták az erőforrásoktól távol eső helyeken is.

A több számítógépből álló rendszerek úgy működnek, hogy az egyes funkciókat ellátó egységeket földrajzilag különböző helyeken, szétszórtan tartalmazzák. A rendszerek működhetnek terhelésmegosztásos vagy erőforrás-megosztásos alapon, sőt a két módszer kombinációja is lehetséges. A terhelésmegosztás feltételezi, hogy a rendszer több hasonló funkciójú egységből áll, és mindegyikük bizonyos adott mennyiségű munka elvégzésére képes. Ha az egyes számítógépek pillanatnyilag foglaltak is, amikor új feladat érkezik a rendszerbe, ezt az új munkát az éppen szabad egységhez lehet irányítani. Erőforrás-megosztásos alapon működő rendszerekben az összetevő egységek nem hasonlóak, mindegyiknek meghatározott szerepe van, a hozzájuk tartozó erőforrások viszont elérhetők a rendszer más részei számára is. Ilyen módon a helyi feldolgozó egységek az adatok természetes előfordulási helyén helyezhetők el, s az egész rendszeren belül csökkenthetik az átviteli igényeket.

Az osztott vezérlés jóvoltából az egyes felhasználók felügyeletet gyakorolhatnak saját adataik és ezek feldolgozási folyamatai felett, a szükséges javítások és beavatkozások helyben megoldhatók. A helyi feldolgozóegységek egyben rugalmasan alkalmazkodhatnak az egyes felhasználók igényeihez. A teljes hálózat moduláris felépítésű, megkönnyíti a változtathatóságot, illetve a bővítést. Egyes helyi berendezések meghibásodásakor az adatfeldolgozás nem szenved kárt, ugyanis a hálózat más, hasonló berendezései minden megszorítás nélkül magukra vállalhatják az aktuális feladat elvégzését.

Számítógépes műszaki tervezés és gyártás (CAD/CAM)

Az elektronikus számítógépek felhasználása egyik legtöbbet ígérő területének már a kezdeti periódusban a műszaki tervezés, illetve az ipari folyamatok automatizálása bizonyult, noha ebben az időszakban a számítógép inkább mint "gyorsabb és okosabb lógarléc" jöhetett szóba a hagyományos tervezési feladatokkal járó nagy mennyiségű számítási munka elvégzésében.

Az olyan típusú felhasználások, amelyekben a számítógép munkaeszközzé, majd egyre jobban munkatársá lépett elő, a fejlett ipari államokban körülbelül a hatvanas évek elején jelentek meg. Az elmúlt három évtized során a hardvereszközök, a rendszerprogramozási és -fejlesztési eszközök teljesítőképessége ugrásszerűen megnőtt. A grafikus terminál (adatállomás) a számítástechnika eddigi eredményei közül talán az egyik leglátványosabb. Szinte esztétikai élményt nyújt (gyakorlati jelentősége mellett), amint a képernyőn nagy sebességgel kirajzolódik egy-egy bonyolult matematikai függvény vagy élethűen megjelenik egy épület vagy gépkocsi perspektivikus vagy vonalhálós képe, amely bármely látószög alatt megforgatható. Lehetőséget nyújt arra, hogy mozgás közben lehessen elemezni olyan objektumokat, amelyek eleget tesznek jól ismert természeti törvényeknek, de megjeleníthetők olyan elvont objektumok is, amelyek viselkedését egyedül a program matematikai modellje szabja meg. A számítógéppel segített műszaki tervezés (CAD = Computer Aided Design) kialakulása szorosan összefügg a számítógépes grafika és a különböző numerikus eljárások kifejlődésével. A számítógéppel segített gyártás (CAM = Computer Aided Manufacturing) alapja viszont a szerszámgépek számítógépes vezérlése és a gyártási adatbázisok elveinek kidolgozása. Tulajdonképpen már a számítógépes vezérlés igazolta, hogy a számítógépeknek a termelési folyamatba való bevonásával érzékenyen emelhető a hatékonyság és megbízhatóság szintje, növelhető a működési sebesség, csökkenthető az emberi élőmunka-ráfordítás. Napjainkra már jellemzőek a szerszámgépgyártásban, a gépkocsiiparban és sok más területen a számítógépes technológiák, amelyek nemcsak kimondottan a gyártást, hanem a kapcsolódó-kiszolgáló

tevékenységeket is többé-kevésbé automatizálják. A modern CAM rendszerek legfontosabb részterületei a számítógépes vagy közvetlen számjegyes vezérlés, a termelés-irányítás, a robotizált anyagmozgatás, az anyagszükséglet-tervezés és -biztosítás, a minőség-ellenőrzés és -biztosítás. A legfejlettebb ipari országokban megvalósulóban van a készlet nélküli termelés, amely elképzelhetetlen számítógépes segédlet nélkül, viszont számottevően növeli a gazdaságosságot. Az ipari robotok megjelenésével már ma is léteznek olyan festő-, hegesztő- vagy szerelőcsarnokok, amelyek teljesen nélkülözhetik az emberi munkaerőt.

Az évek során e két fontos felhasználási terület egyre jobban közeledett egymáshoz, hogy napjainkra egységes technológiává olvadjon össze. Ezt a megállapítást mi sem támasztja jobban alá, mint az, hogy ma már léteznek olyan CAD/CAM rendszerek, amelyekben a CAD rendszer eredményei közvetlenül felhasználhatók gépek vezérlésére, illetve a gyártás megszervezésére, lebonyolítására.

A legelterjedtebb felfogás szerint a CAD/CAM a számítógépes hardver és szoftver erőforrások olyan összehangolt felhasználását jelenti, amely a termékek tervezésében és megvalósításában egyre rövidebb gyártási ciklusokat, jobb termékminőséget biztosít. Valójában a tényleges CAD/CAM rendszerek magukban foglalnak — a számítógépes eszközökön kívül — egy egész sor gazdasági erőforrást, mérnöki-tervezői szakismeretet és kreativitást, módszertani-technológiai ismereteket stb. A számítógépes rendszer és a felhasználó közös tevékenységére alapozva olyan fejlesztő rendszer hozható létre, amely a felek egymást kiegészítő képességeinek hasznosítása mellett — az erőforrások adott szintjén — az optimális technológiát képes megvalósítani.

A számítógépes erőforrások tervezésbe való beépülésének folyamatát a fokozatosság jellemezte mind a felhasználás mértéke, mind annak célja tekintetében.

Kezdetben a tervezési tevékenységet a tervező nagyrészt hagyományos eszközökkel végezte, a számítógép részvételét olyan célprogramok jelentették, amelyek a tervezési folyamat meghatározott részfeladatainak megoldásában nyújtottak segítséget. Ezek önmagukban zártak és egymástól függetlenek voltak, vagyis nem kapcsolódtak más részfeladatokhoz. A programok sorrendjéről a felhasználó döntött, így a szervezésnek ez a legalsó szintje nem tette lehetővé az integrált programrendszerek kialakítását. A tervezési munka strukturálatlansága viszonylag kevés lehetőséget adott a számítógépes támogatásnak.

Az intuitív tervezés eredendő fogyatékoságait különböző módszertanok kidolgozásával igyekeztek kiküszöbölni: a tervezési folyamatot részlépések sorozatára bontották, a tevékenységelemekhez pedig különféle részlegesen algoritmizálható eljárásokat és modelleket kapcsoltak. A tervezési feladatmegoldásban már programcsomagok vettek részt, ahol az egyik program kimenete közvetlenül a következő bemenetére került. A megoldáskeresés tulajdonképpen megoldásvariációk előállítását jelentette. Ezek összehasonlítása, kiértékelése olyan viszonylag optimális megoldások kiválasztására adott lehetőséget, amelyeket a hagyományos intuitív tervezéssel csak esetlegesen lehetne előállítani. A munka hatékonyságának ugrásszerű növekedését idézte elő a tervezési katalógusok és eljárások számítógépes feldolgozása és kezelése.

A számítógépes erőforrások technikai és technológiai fejlődése, a párbeszédes feldolgozás, a grafikus szemléltetés általános elterjedése lényeges változásokat tett lehetővé a számítógépes erőforrások felhasználásának minőségében. Megjelentek a geometriai modellezési szoftverek, amelyek képesek voltak különböző objektumok két- vagy háromdimenziós, rajzolás vagy szerkesztéses megjelenítésére (vonalvázás megjelenítések, matematikai felületleírások stb.). Minőségi előrelépést jelentettek a grafikus szemléltető szoftverek, amelyekben a takartvonal-ábrázolás, a felületeltávolítás, az árnyékszerkesztés, a függvénygrafika s a mérnöki tervezés kelléktárának megannyi más eleme is megtalálható. Kialakultak a kisebb-nagyobb adatbázisok, a tervezett objektumhoz kapcsolódó geometriai, szemléltetési, elemzési, technológiai és dokumentálási információk rendezett, könnyen és hatékonyan lekérdezhető rendszerei. Ezek magukban foglalták a korábban gyártott termékekre vagy részegységekre vonatkozó leíró infor-

mációkat, szabványokat, vállalati előírásokat, felhasználói követelményeket egyaránt. Köztudott, hogy a jó minőségű, részletes gyártási dokumentációk kidolgozása igen munka- és időigényes feladat. Az információk, a megvalósított tipizált, illetve szabványos elemek, megoldások gyors lekérdezése egyes hagyományos munkalépések kihagyását, az egykor soros jellegű tevékenységek párhuzamosítását is lehetővé tette. A számítógépes adatbázisok, illetve megjelenítő eszközök gyorsan kivitelezhető, pontos és redundanciamentes tervek tesznek lehetővé. Ennek fontosságát csak növeli az a tény, hogy a gyakorlati megvalósítások sikertelenségét sokszor nem is a hibás koncepció, hanem éppen a kivitelezési tervek hibája, pontatlansága okozza.

Az új technológia így eljutott egy olyan szintre, amikor számítógéppel segített tervezésről és gyártásszervezésről lehet beszélni. A módszer alapvető jellegzetessége, hogy az eredményes alkalmazás megköveteli az adott terület emberi szakértőjét, mert csak az képes hatékonyan és értelmesen kihasználni a számítógépes erőforrások nyújtotta lehetőségeket. A mikroprocesszorok megjelenése és elterjedése ezen a területen is mélyreható változásokat eredményezett. Az elmúlt évtizedben megjelentek és egyre hatékonyabbá váltak a helyi hálózatokba szervezhető, professzionális mikroszámítógépekre alapozott, szűkített szolgáltatású CAD/CAM rendszerek, amelyeket feladatorientált tervezői munkahelyeként lehet használni.

A napjainkra jellemző tervezői-szervezői munka mennyiségi és minőségi elvárásai oda vezettek, hogy a teljes értékű szakértői tudással rendelkező szakemberekben egyre nagyobb a hiány. Egyre jobban érezhető az az igény, hogy nem szakértő tervezők is meg tudjanak oldani szakértői ismereteket követelő feladatokat. A mesterségesintelligencia-kutatások, a szakértői rendszerek megjelenése olyan rendszerek gyakorlati megvalósítását ígéri, amelyek megteremthetik a most már számítógépes tervezésnek és gyártásnak nevezhető módszert. Ebben az új technológiában már az ismeretfeldolgozás is megjelenik. A számítógéppel segített, illetve számítógépes megnevezés nem egyszerűen szójáték, minőségi különbséget hivatott kifejezni. A számítógéppel segített rendszerről azt lehet mondani, hogy csak "inasként" és nem értő társként működik közre az emberi szakértő munkájában. Nyilvánvaló, hogy a tervezés lényegesen hatékonyabbá válna, ha a rendszer már eleve kiindulási megoldásvariációkat adna a tervező által leírt problémákra, "szakértői társi" szintre emelkedne. A mesterségesintelligencia-kutatások eddig elért eredményei az intelligens tervezőrendszerek létrehozásában is biztató távlatokat nyitnak, bár még nem beszélhetünk kikristályosodott elgondolásokról sem. A közeli-távolabbi jövő feladata lesz, hogy a teljesen automatizált technológiát megvalósítsa.

A CAD/CAM rendszerek fejlődési szakaszainak e rövid áttekintéséből valószínűleg kitűnik, hogy napjainkban körülbelül a számítógéppel segített és a számítógépes fázis határánál tartunk. Már készültek el számítógépes rendszerek egy-egy szűkebb szakterületen, de még korántsem nevezhetők általánosoknak. A számítógéppel segített rendszerek viszonylatában már kialakult egy sor olyan alkalmazási terület, amely ma szinte klasszikusnak nevezhető. A teljesség igénye nélkül tekintsük át ezek közül a legfontosabbakat.

Természeténél fogva az egyik legfontosabb terület a gépipari tervezés. Bizonyos tervezési fázisokban a gépipari mérnök szinte kizárólag rajzban gondolkodik, másokban pedig a számítások eredményeit grafikonokkal, eloszlásdiagramokkal, elmozdulásfüggvényekkel, átviteli függvényekkel vagy a szerkezeteknek a terhelés hatására bekövetkező deformációjával szemlélteti. A numerikus vezérlésű szerszámgépek szerszámpályáit, a szerszám-munkadarab-készülék hármas kölcsönös egymásra hatását korszerű interaktív (párbeszédes üzem módú) módszerekkel vizsgálhatják, a paraméterek dinamikus változtatásával rögtön elemezhető a különböző megoldásvariációk. A gépipari tervezésnél és gyártásnál használt nagy mennyiségű rajzos dokumentáció elkészítése viszonylag könnyen automatizálható, így a megvalósításban érintett szakemberek mentesülnek a rutinjellegű feladatoktól, képességeik, szakismeretük és kreativitásuk jobban hasznosítható. A legszélesebb körben alkalmazott, szinte önálló területnek a véges elemek módszere tekinthető. Ezt elsődlegesen a szerkezeti elemek feszültségi és alakváltozási állapotának a meghatározására használják, de jól alkalmazható hőtani, áramlástechnikai és képlékenyalakítási

feladatok megoldásában is. A módszer lényege az, hogy a szerkezetet nagy számú, véges méretű, idealizált viselkedésű elemi részre osztják, és ezek elemzése útján képet nyernek a teljes szerkezet viselkedéséről. Az új módszer szinte teljesen kiszorítja a hagyományos szilárdsági méretezési eljárásokat, mert sokkal pontosabb náluk. Számos iparágban, mint például a repülőgépgyártásban, a hajóépítésben, a reaktorteknikai berendezések és az űrhajózás területén tapasztalt fejlődés nem valószínű, hogy megvalósulhatott volna meg a tervezésbe és a technológiai megvalósításba szervesen beépült számítógépes segítség nélkül.

A gépipari feladatok megoldásához hasonlóan a számítógépes támogatás az építészeti tervezés egészére is kiterjeszthető, ahol igen nagy mennyiségű grafikus információra van szükség: alaprajzok, elrendezési rajzok, homlokzati, alapozási, szerkezetkiosztási, fűdémkiosztási tervek, épületgépészeti tervek (csatorna-, víz-, gáz-, fűtés-, elektromos vezetékek stb.), berendezési tervek stb. A területfelhasználási tervek kidolgozásához létrehozható a helyszín háromdimenziós felületmodellje, amelynek alapján szinte élethű módon elemezhető egy-egy épület vagy épületegyüttes külső megjelenése, elrendezése, a környezetbe való illeszkedése. A számítógépes grafika mai fejlettségi szintjén már tónusos, árnyékos képek is előállíthatók, így megtakarítható a modellek, makettek költsége és időigényes elkészítése. A szakipari — víz-, gáz- és villamoshálózati — tervezéshez és szilárdsági számításokhoz háromdimenziós huzalvázás modellezés áll rendelkezésre.

A mély- és magasépítés, illetve az út-, vasút- és vízépítés műtárgytervezése is az építészethez hasonló igényeket támaszt a számítástechnikával szemben. Kiemelt szerepet játszik a számítógéppel segített terepfelvétel, amelynek során a mért terepadatok felhasználásával modellezik, majd háromdimenziós grafikával jelenítik meg a területet. A statikus vizsgálatok mellett igen fontos az építmények dinamikus elemzése is. Ez magában foglalja a sajátfrekvenciák és lengésképek meghatározását, az idő- és a frekvenciatartományú dinamikus és sztochasztikus gerjesztésekre adott szerkezeti válaszok vizsgálatát, a kihajlás- és a horpadásvizsgálatokat, a földrengésállóság elemzését is. Előnyösen használható a CAD-technológia a közúti- és vasútipálya-tervezésben. Grafikus feldolgozással domborzati-szintezési térképek szerkeszthetők, nyomvonaltervek és pályaszelvényrajzok dolgozhatók ki.

A villamosipari tervezés kapcsolódik össze a legjobban a számítástechnikával, a CAD rendszerek mind az elektronikában, mind az erőáramú tervezésben jelentős szerepet játszanak. A számítógépes grafika egyik első alkalmazása az elektromos áramkörök kapcsolási és huzalozási rajzainak interaktív tervezése volt. Ennek során a tervező előre megadott elemtípusokból állíthat elő áramköri vázlatokat. Alkalmazható a CAD-technológia a készülékek funkcionális és logikai tervezésében, a nyomtatott áramköri kártyák — térkihasználás-optimalizálást is elősegítő — geometriai tervezésében, működéselemzésében és gyártási előkészítésében. Komplikáltabb esetekben a kapcsolási rajzokkal egy időben létrehozható a tervezett készülék matematikai modellje, így mód nyílik annak számítógépes szimulációjára, a vizsgálat eredményeinek megjelenítésére, s elmarad az a veszteség, amit a működésképtelen, kivitelezhetetlen megoldások létrehozása és kipróbálása jelent.

Igen összetett feladat, s emiatt igényli a számítógép közreműködését a villamoshálózat-tervezés. Így meghatározhatók a csomópontok teljesítményei, a hálózati veszteségek, az összegezett teljesítményértékek és a vonalak feszültségtartományai, elemezhető a transzformátorok kihasználása és a vonalak leterhelhetősége. Szimulálhatók a különféle hibák, ellenőrizhető a földelés.

Új lehetőségeket nyitnak a CAD/CAM rendszerek az ipari formatervezésben is. A hagyományos grafikai eszközökkel csak nehezen szemléltethető formák a képernyőn természetes hatást keltve jelennek meg. Bevált és sikeres eszköz a számítógépes grafika a ruházati alapanyagok gyártásában, ahol a textíliák mintájának tervezésében segít.

Egyik legújabb alkalmazási területként a szakemberképzés említendő. Az elmúlt években jelentették be egy teljes CAD/CAM-oktatórendszer elkészültét, beépített robotokkal, oktató- és tervezőszoftverrel. Az egész összeállítás egy megvalósított integrált gyártórendszer. Segítségével egy hónap (!) alatt képezhetők ki NC-technikusok. Az első

héten a kétdimenziós forgácsolást, a másodikon a háromdimenzióst mutatják be. Az elvi alapok elsajátítása mellett elkészítenek 10—15 munkadarabot. A harmadik héten már ipari vezérlés következik, a rutinfeladatok gyakorlása, sok bemutató programmal. Végül a negyedik héten már tervezhetnek a technikusjelöltek, s a tervezett alkatrészek azonnal le is gyárthatók!

Számítógépes oktatás (CAI)

A számítógéppel segített oktatás (CAI = Computer Aided Instruction) alig több, mint két évtizedes múltta tekinthet vissza. Fejlődésének ütemét tekintve azonban nyilvánvaló, hogy ez az új oktatási forma rövid időn belül ki fogja szorítani a hagyományos formákat és eszközöket. Az elsajátítandó ismeretanyag mennyiségi és minőségi színvonala egyre nehezebb helyzetbe hozza a klasszikus iskolarendszert, s csak a módszerek és az eszközök minőségi változása oldhatja meg a felmerülő problémákat.

A számítógépeket tulajdonképpen megjelenésük óta használják az oktatás területén is (elsősorban a számítástechnikai oktatás segédeszközeként, illetve az oktatással kapcsolatos műszaki-adminisztratív feladatok megoldásában), de kimondottan számítógéppel segített oktatásról csak az időosztásos működésű, párbeszédű üzemű számítógéprendszerek megjelenése óta, tehát a hatvanas évek második felétől fogva beszélhetünk. Az azóta eltelt idő alatt az oktatási rendszerek igen gazdag választéka alakult ki, amelyek sok szempontból különböznek egymástól. Ennek ellenére néhány általános jellemvonás szinte minden ilyen rendszernél felfedezhető.

A legkülönbözőbb területeken és a legváltozatosabb témák oktatására igen hatékony oktatórendszerek jelentek meg az óvodától az egyetemekig, a szakmunkásképzéstől az űrhajósjelöltek felkészítéséig. Ezek a rendszerek általában két szinten valósulnak meg. Külön alrendszer áll a tanárok rendelkezésére, akik az elsajátítandó ismeretanyagot rendszerezik és programozzák, és külön alrendszer a tanulók számára. A számítógép oktatóprogramokat tárol, amelyeket szelektív módon, rövid idő alatt tud a tanulók rendelkezésére bocsátani. A legegyszerűbb rendszerek tankönyvszerűen használhatók: előre-hátra lehet lapozni az anyagban. A tananyag rendszerezése, bemutatási formája rendszerint sokban különbözik a hagyományos formától, ugyanis az új eszköz lehetővé teszi a vonatkoztatások, kapcsolatok, összefüggések sokkal szuggesztívebb bemutatását. Lehetőség van a tanuló és az oktatóprogram dinamikus kölcsönhatásának megteremtésére. Egyrészt a tanuló kezdeményezheti az aktuális tananyagrészt megértéséhez szükséges megelőző információk gyors visszakeresését, felelevenítését, saját maga határozhatja meg az elsajátítás ütemét stb. Másrészt a tanuló válaszait kiértékelve az oktatóprogram kényszerítheti a tanulót egyes tananyagrészekre való visszatérésre, a tanuló konkrét képességeinek, szorgalmának legjobban megfelelő oktatási stratégia megválasztására, követésére.

Az oktatási rendszerek általában tárolják a tanulók teljesítményére vonatkozó adatokat az oktatási folyamat alatt, kiértékelő és statisztikai nyilvántartásokat végezhetnek, s ezáltal sokkal realisabb értékeléseket adhatnak egy-egy tanfolyam elvégzése után a tananyag elsajátításának mértékéről, mint a hagyományos iskolarendszer.

Az oktatási rendszer sokkal magasabb fokon érvényesítheti az audiovizuális eszközök bevonását a tanítási folyamatba. Minőségi előrelépést jelentett a számítógépes oktatás fejlődésében a számítógépes grafika egyre többet tudó eszközeinek felhasználása. A sokszor száraz nyomtatott szöveg helyett kifejező, a folyamatokat lefolyásukban ábrázoló, mozgó képsorok, animációs megvalósítások révén a lényeg alaposabb, nem mechanikus elsajátítását teszik lehetővé. A vizuális információ rögzítése gyorsabb, alaposabb és pontosabb minden más formánál (ezt már a számítógépek megjelenése előtt is tudták!). Ezzel az új eszközzel nemcsak új minőségi szintre emelhetik a vizuális tanítás és tanulás színvonalát, hanem lehetővé válnak olyan szimulációk és kísérletek is, amelyek valós körülmények között igen költségesek vagy egyenesen kivitelezhetetlenek lennének. Szinte esztétikai élményt nyújt s a tanulást is érdekesebbé, vonzóbbá teszi, amint a képernyőn

nagy sebességgel kirajzolódik egy-egy bonyolult matematikai függvény, élő szervezetekben lezajló folyamatok ábrázolása, vagy megjelenik egy-egy híres épület vagy képzőművészeti alkotás perspektivikus képe, amelyet bármely látószög alatt megforgathatunk. Lehetőség nyílik arra, hogy mozgás közben elemezhetők legyenek olyan objektumok, amelyek eleget tesznek jól ismert természeti törvényeknek (fizikai vagy kémiai kísérletek). Megjeleníthetők olyan elvont objektumok is, amelyeknek viselkedését egyedül a program matematikai modellje szabja meg, mint például a csillagászati kísérletek szimulálásánál.

A számítógéppel segített oktatás rendszerint valós idejű, párbeszédéses üzemmódu alkalmazás. A mikroprocesszoros technika elterjedése lehetővé tette, hogy minden tanuló saját képernyős terminálja előtt ülve párbeszédéses módban használja az oktatóprogramot, másoktól függetlenül, a számára legjobban megfelelő ritmusban.

Az iparilag fejlett országokban a számítógépes oktatásnak köszönhetően új fogalomként jelent meg a "nyílt" egyetem. A nyitottság több szempontból is igaz: a hallgatók nem tesznek felvételi vizsgát, a tanmenet nincs rögzítve, a moduláris oktatási szerkezet sokféle elágazást, pályamódosítást tesz lehetővé. Nem csupán a felsőfokú, hanem az alacsonyabb szintű képzést is biztosítja. Az oktatás döntő része nem az egyetem falain belül, hanem a hallgatók otthonában vagy munkahelyén történik. Az oktatási anyaghoz a hallgatók számítógép-hálózaton vagy adathordozók segítségével juthatnak hozzá. A hallgatók a tantárgyak hierarchikusan felépített, moduláris rendszeréből önmaguk választhatják ki azokat a tantárgyakat, amelyeket el akarnak sajátítani. A tanulás bármelyik szinten megkezdhető, esetleg később folytatható. Ez a rendszer eleve számol azzal, hogy a hallgatók egy jelentős része már rendelkezik bizonyos ismeretekkel. Az egyetem valójában nem oktatja a hallgatókat a szó szoros értelmében, hanem az önálló tanuláshoz biztosítja a feltételeket.

A nyílt egyetemeknek fontos társadalmi vetülete is van. Közismert, hogy vannak "későre érő" emberek. Ezek is esélyt kapnak arra, hogy felsőfokú ismereteket szerezzenek felnőttkorukban. A távtanulás gyakorlatilag az egyetlen esély a kisgyermeküket otthon nevelő anyáknak, a katonai szolgálatot teljesítőknek, egészségkárosultaknak vagy a távol eső kistelepülések lakóinak.

Számítógépes szövegfeldolgozás, automatikus fordítás

A szöveges információk számítógépes feldolgozása szinte egy időben az elektronikus számítógépekkel. Az információhordozóknak ez a fajtája tűnt a legmegfelelőbbnek a számítógépek belső ábrázolására és a külvilággal tartott kapcsolat megteremtésére egyaránt. A szöveges információk hatalmas mennyiségi növekedése szinte kötelezővé tette a gépi feldolgozást, ugyanis a szerkesztéssel, rögzítéssel, továbbítással, visszakereséssel járó tetemes mennyiségű munka immár lehetetlen volna csupán emberi munkaerő és klasszikus eszközök bevetésével. Ezeknek az adminisztratív munkáknak a gépesítése terén a számítástechnikának kézzel fogható, komoly eredményei vannak: ma már a hobbi kategóriájú személyi számítógépeknek is olyan szövegszerkesztő és -karbantartó rendszereik vannak, amelyek meghaladják az ideális titkárnőktől, könyvszerkesztőktől és nyomdai szakemberektől elvárható legmagasabb követelményeket együttvéve.

A szövegszerkesztő programok, rendszerek fő feladatai a szöveges információk előkészítése, megfelelő formára való kialakítása, nyomtatása, illetve nyomtatásra való előkészítése, s nem foglalkoznak az információk nyelvi vonatkozásaival, csupán karakterláncokat kezelnek és jelenítenek meg. A legegyszerűbb szövegszerkesztők végterméke csupán szövegsorok tömege, amelyen esetleg többé-kevésbé komplikált módosító műveletek végezhetők. A komolyabb rendszerek már tipográfiai elrendezést, közbeiktatott rajzokat is lehetővé tesznek, más esetekben az eredmény könyvszerűen tagolt szöveg, fejezetcímekkel, számozott bekezdésekkel, számozott lapokon való elhelyezéssel, megszerkesztett tartalomjegyzékkel, illetve tárgymutatóval. Mint érdekesség említhető,

hogy már léteznek olyan szövegszerkesztők, amelyek az olyan esztétikai hibák elkerülésére is vigyáznak, hogy egy széles szóközzel tele sor után ne következzen sűrűn nyomtatott sor. Az egyre nagyobb felbontású képernyők felhasználásával a számítógép olyan formában jelenítheti meg a későbbiekben kinyomtatandó oldalt, ahogyan az a nyomtatás után ténylegesen mutatni fog (beleértve a betűk formáját, nagyságát, típusát, az egyes szövegrészek síkbeli elrendezését, az esetleg felhasználható grafikai díszítőelemeket is stb.). A felhasználó speciális billentyűzet és egy "egérnek" nevezett mutatóbeviteli eszköz segítségével közvetlenül alakíthatja ki az oldal kívánt elrendezését (ez az úgynevezett "azt kapod, amit látsz" kapcsolat). Kezdenek megjelenni az olyan programrendszerek is, amelyek a klasszikus irodalmi szerkesztési munkákat is elvégzik. Ezek nem a szöveg vizuális képével foglalkoznak, s nem is a szöveg tartalmával, hanem a helyesírással, a nyelvtannal és a stílussal. Ide tartoznak a gépesített háttérszótárak és szöveg helyesség-vizsgáló programok. Ezek lehetővé teszik, hogy amikor a felhasználó megjelöl egy-egy szót a képernyőn, speciális ablakeljárással megjelenik az illető szó helyes nyelvtani alakja, esetleg szinonimáinak listája. Ezután a szerkesztő-felhasználó kérheti a helyes nyelvtani alak beépülését, kijelölhet egy megfelelőbb szinonimát és a kicserélendő szó helyét, amely azután automatikusan a szövegben eredetileg szereplő szó helyére kerül. A legmodernebb ilyen jellegű programrendszerek arra is képesek már, hogy az átlagos mondathosszúság, a szavak átlagos hossza és más ilyen típusú jellemző alapján meghatározzák az "olvashatósági" mutatót, s felhívják a szöveg írójának figyelmét a rossz minősítésű szövegrészekre.

Az általános feladatnak csak egyik, legegyszerűbb vonatkozásaként tekinthetők a szöveges információk feldolgozásának eddig ismertett módok, amelyek tulajdonképpen csak formai feldolgozást, illetve rögzítést jelentenek. Sokkal bonyolultabb és érdekesebb feladat a szöveges információkat mint a kommunikáció egyik megnyilvánulási módját tekinteni, tehát azoknak szemantikai, jelentéstartalmi vonatkozásait is figyelembe venni.

Az első programozási nyelvek megjelenésével és a számukra elengedhetetlen fordítóprogramok előállításával sokak számára úgy tűnt, hogy a természetes nyelvek megértése, egymás közötti fordítása, majd a természetes nyelvű ember-gép kapcsolat megvalósítása csupán idő kérdése. Csak a hatvanas évek közepén lett világos, hogy az ember túl nagy fába vágta fejszéjét, és még nagyon sok kutatásra, munkára van szükség ahhoz, hogy az egykor közelinek tűnő célt valaha is megvalósíthassa.

A természetes nyelvű információk feldolgozása tulajdonképpen a gépi fordítás megvalósítására irányuló törekvésekkel indult, amelyek kezdetben azon a hibás elképzelésen alapultak, hogy a számítógép mechanikusan le tud fordítani valamilyen mondatot, ha a mondat egyes szavainak megfelelőjét a célnyelvben megkeresi. Az egyes természetes nyelvekben használatos szórend kialakítása sem volna megoldhatatlan feladat. A gépi fordítással foglalkozó tanulmányok jó része ebben az időben olyan technikai kérdésekkel foglalkozott, miként táplálható be a számítógép memóriájába egy hatalmas méretű szótár, vagy hogyan képes a gép a szavakat hatékonyan visszakeresni.

Ezek a programok érthetetlenül rossz fordításokat készítettek, ugyanis a szavak a különböző természetes nyelvekben nem kölcsönösen felelnek meg egymásnak, továbbá nagyon sok szónak több, eltérő értelmezése, jelentése is lehet, s a helyes fordítási változat meghatározása csak az egész mondat értelmezéséből következhet.

Az ötvenes évek első próbálkozásai szinte kizárólag csak a szintaxisra, a nyelvtani formára támaszkodtak. Érthető volt tehát a nyelvészeknek az a törekvése, hogy a nyelvészeti fogalmakat matematikai fogalmak alapján formalizálják. (Ez a természetes nyelveknél nem volt egészen sikeres, annál hasznosabbnak bizonyult a programozási nyelvek leírásában, ugyanis ezek — pontos szerkezetük és egyértelműségük révén — egyenesen kínáltak a formalizálásra.)

A hatvanas évek közepe táján lett világos — sorozatos sikertelenségek után —, hogy a természetes nyelvek definíciójának szintaxisbeli leírása alapvető követelmény, de önmagában még távolról sem elégséges: ezzel csak a nyelvhez tartozó mondatok

szerkezetét írhatjuk le, de nem kapcsolhatunk hozzájuk jelentést. Ez a felismerés oda vezetett, hogy a teljes pesszimizmus vett erőt az addig lelkes kutatókon, s sokan abba is hagyták a munkát, hiszen a természetes nyelvek szemantikája igen nagy mértékben intuitív jellegű.

A problémák közül talán legegyszerűbb a lexikális többértelműség, amikor egy-egy adott szónak több jelentése is van. Például ez a mondat: "Óvakodj a szélről!" — lehet orvosi tanács, de ugyanakkor lehet munkavédelmi felhívás az építőiparban. Az első értelmezésben a helyes román nyelvű fordítás: "Ferește-te de vînt!", a második: "Ferește-te de prag!" lenne. Sem a mondat formája, sem szerkezete nem ad semmi támpontot a helyes változat megtalálására! A bizonytalanság feloldásában esetleg segítséget nyújthat a teljes szöveg statisztikai jellemzése: ha sok orvosi műszó fordul elő, tehát orvosi szövegről van szó, az első változatnak van nagyobb esélye, ipari szövegkörnyezetben viszont a másodiknak. De mi a megoldás, ha az inkriminált mondat egy irodalmi mű fordításánál kerül a felszínre?

Komolyabb problémák forrása lehet az a kétértelműség, amely csak a teljes mondat értelmezésével oldható fel (vagy még így sem!). Például: "Az asztalra dobta az edényt, és eltörte" mondat román fordításánál feltétlenül el kell döntenie a fordítónak, hogy végül is mi tört el, az edény ("vas" — semleges nemű főnév), vagy az asztal ("masă" — nőnemű főnév), hogy a megfelelő nemű formát használhassa ("l-a spart", illetve "a spart-o").

Szinte megoldhatatlan feladatot jelentenek viszont a szemantikai értelmetlenségek! Nézzük a következő mondatot: "A szorgalmas könyv kiolvasta az érdekes tanulót"!!! Ez a magyar nyelvű mondat szintaktikailag, nyelvtanilag teljesen helyes, szemantikailag azonban teljesen értelmetlen. Még ha a magyar nyelv értelmező szótára meg is adja a nyelvhez tartozó összes jelsorozatot (szó) formáját és a hozzája tartozó jelentés(ek)e(t), a fenti mondat még így is csak intuitív, megérzéses, esetleg tapasztalati alapon értelmezhető. Az ember számára ez csak ritkán okoz problémát (főleg anyanyelvi szinten). Ennek a magyarázata az, hogy az értelmezés során az ember a mondat igéjét egyéb ismeretei, szemantikai elvárása alapján a megfelelő tárggyal kapcsolja össze, vagyis figyelembe veszi a kijelentésből fakadó következtetést is. Ezek szerint tehát tudásnak, tapasztalatnak kell közrejátszania ahhoz, hogy a mondatból implicit információkat vonhassunk ki. A természetes sorrend szerint az ember először nyelvtani szempontból elemzi a mondatot, és annak jelentése alapján következtetésanalízist végez.

Szinte áthághatatlan akadályokba ütközik a számítógépes fordítás akkor is, ha humoros vagy erősen átvitt értelmű mondatokat kell átültetni más nyelvre (pedig ezek nélkül szálanalmasan sőtlan, egyszerű volna az emberi kommunikáció!).

A teljesség igénye nélkül felsorolt megoldatlan problémákból kifolyólag teljesen automatizált, jó minőségű gépi fordítás egyenlőre még nem valósítható meg, viszont a számítógép máris megkönnyítheti a fordítást. Előzetes kézi feldolgozással elő lehet állítani a lefordítandó szöveg egy olyan leegyszerűsített változatát, amelyet a számítógép már elég biztonsággal le tud fordítani. Így például ki lehet küszöbölni a többjelentésű szavakat és a szintaktikai elemzést nehezítő nyelvtani szerkezeteket. A nem egyértelmű kapcsolatokat meg lehet szüntetni vagy speciális jelekkel felhívni rájuk a fordítórendszer figyelmét, megfelelő útmutatásokkal segíteni a kétértelműségek feloldását. A gépi fordítás végén egy emberi utószerkesztő ellenőrizheti a melléfogásokat, finomíthatja a lefordított szöveget. Ez a módszer, főleg szakirodalmi szinten, elfogadható eredményeket hozott. Újabb reménysugarat villantottak fel a gépi fordítás problémájának megoldására az elmúlt 10—15 év során a mesterségesintelligencia-kutatások eredményei. A szemantikának tudásbázis formájában történő rögzítése révén sikerült egypár olyan automatikus fordítórendszert előállítani, amelyek — elsősorban egy-egy szűkebb szakterület viszonylag egyértelműbb szövegeinél — 70—80%-os helyességet is elértek. Napjainkban már az is komoly eredménynek számít, mert az emberi beavatkozást csak a majdnem kész fordítások ellenőrzésére és kijavítására kell felhasználni. Érdekes és reménykeltő eredmények születtek az eszperantó nyelvnek átmeneti nyelvként való felhasználásában. Ennek a nyelvnek a kétértelműségi szintje nagyságrenddel kisebb, mint a nemzeti

nyelveké, teljesen fonetikus, nyelvtana egyszerű és kivételektől mentes, és nem utolsósorban elég elterjedt ahhoz, hogy a különböző nyelvek közötti híd szerepét sikeresen betölthesse.

A természetes nyelv teljes, emberi jellegű megértését megvalósító programrendszer mai tudásunk alapján még megvalósíthatatlan. Az írott szöveg fordításánál említett nehézségeket még súlyosbítja a beszélt nyelv speciális problematikája. Gondoljunk csak azokra a gondokra, amelyeket a szinte folytonos akusztikai jel összetevő fonémáira bontása, majd ezekből a morfológiai egységek felépítése okoz. Így a diktálás után író "fonetikus írógép" reménye éppen olyan halvány, mint a jó minőségű gépi fordításé vagy a gépi nyelvmegértésé. Vannak viszont már olyan eszközök, amelyeknél nem feltétlenül szükséges a beszéd kimerítő elemzése, s néhány száz szóból álló speciális parancsokat nagy biztonsággal felismernek.

Mesterséges intelligencia — szakértőrendszerek

A szakemberek rosszallása, ellenkezése dacára az elektronikus számítógépekről általában sokáig úgy beszéltek, mint "elektronikus agyakról", értelmes gépekről, amelyek olyan szellemi teljesítményekre képesek, mint a legmagasabban képzett szakemberek, sőt még ennél is többre.

1981-ben azonban Japánban létrehoztak egy tudományos intézetet azzal a kifejezett céllal, hogy megvalósítsanak egy új, a "mesterséges intelligenciát" megtestesítő számítógépet (az ötödik generáció megnevezéssel). Magától tevődik fel a kérdés: miért, az eddig ismeretes számítógépek mégsem voltak ténylegesen intelligens gépek? Végül is mi az, hogy mesterséges intelligencia? Mitől ez az éles elhatárolás (a klasszikus számítógépes programok és a mesterséges intelligenciát megvalósítók között), amely egy minőségileg magasabb szinten álló számítógép-generáció megszületését követeli?

A mesterséges intelligencia néven emlegetett új, interdiszciplináris kutatási terület pontos meghatározása mind a mai napig nem talált általános elismerésre. A. Barr és E. A. Feigenbaum meghatározása alapján a mesterséges intelligencia a számítástechnikának az az új ága, amely intelligens rendszerek előállítását tűzi maga elé célként, vagyis olyan rendszerek előállítását, amelyek az emberi intelligencia megnyilvánulásait képesek reprodukálni: nyelvek megértését, tanulást, problémák megoldását, absztrahálást, elvonatkoztatást, általánosítást stb. Úgy tűnik, a mesterséges intelligencia meghatározásához a legjárhatóbb út a természetes intelligencia meghatározásából indulhatna ki. De mi akkor a természetes intelligencia? Egyesek szerint az embernek az a tulajdonsága, hogy képes megérteni a környező világ jelenségeit, hogy ismereteket képes felhalmozni és alkotóan felhasználni. Mások szerint az a képessége, hogy előzetes ismereteinek szelekciója, szintézise, illetve absztrahálása révén képes új, merőben ismeretlen szituációkhoz alkalmazkodni, képes az új körülmények információtartalmát feldolgozni, elsajátítani. Megint mások szerint az embernek az a képessége, hogy a konkrét világ összefüggéseinek ismeretében képes elvonatkoztatni, absztrakt fogalmakban gondolkodni, majd gondolatmenetének eredményeit felhasználva a konkrét világ eseményeit bizonyos valószínűséggel előre látni. Mindezek a nézetek közös nevezőre jutnak viszont abban a megállapításban, hogy az intelligencia semmiképpen sem magát a tudást, a képzettséget, a pusztá ismeretanyagot jelenti, sokkal inkább a tanulásnak, a tudás megszerzésének, feldolgozásának és célszerű alkalmazásának a képességét.

Az intelligencia mértékének meghatározása állandóan változott az idők folyamán. Nem is olyan régen intelligensnek tartották azt az embert is, aki könnyedén elvégezte az elemi aritmetikai műveleteket. E nézet hamisságát mi sem bizonyítja jobban, mint az a tény, hogy a tudomány, a művészet jelentős képviselői csak igen kivételes esetekben voltak jó fejszámolók is. Csupán az elmúlt pár évtizedben körvonalazódott ki igazán az intelligencia és a megismerés, a tudás közötti összefüggés, s a hatvanas évek második felében alakult ki egy olyan intelligenciamodell, amelyet már bizonyos konkrét adatok is

alátámasztanak. E modell szerint az intelligencia három fő szempont szerint tanulmányozható, ezek: a gondolati tartalom, a műveletek természete és a gondolkodás végeredményei. Az ember természetes tevékenysége során ezek az intelligenciátényezők nem válnak el egymástól. Munka, beszéd, tanulás vagy akár játék közben is valamennyi funkció közrejátsszik.

A hagyományos számítógépes programok jellegzetessége, hogy numerikus vagy alfanumerikus adatokon dolgoznak, és a kívánt végeredményhez algoritmikus úton jutnak el. Ez azt jelenti, hogy a programot író szakember a megoldást előállító algoritmust már eleve ismeri (tehát számítógépen kívüli használhatóságának esetleg csak a műveletvégzési sebesség szabhatna határt), s ennek alapján írja elő a számítógép által a későbbiekben elvégzendő műveleteket. Csakhogy az emberi tevékenység számos területén a problémákat jellemző információk nem numerikus jellegűek, és kiértékelésükhöz sincsenek mindig szigorú, pontos eljárások. Gondoljunk csak az orvosi tevékenységek zömére, ahol a szakember a beteg által körülírt panaszok, az ok és okozat közötti összefüggések mérlegelése alapján diagnosztizál, megállapítja a pontos kórismét, esetleg behatárolja a lehetséges betegségek körét, és a pontosabb felismerés végett további vizsgálatokat rendel el. Ha munkamódszere után érdeklődünk, tanult ismeretein kívül alighanem tapasztalatára, megérzésére hivatkozik, egyszerűen csupa olyan dologra, amit reménytelen volna algoritmikusan megközelíteni. Vannak azután olyan feladatok, mint például a mérnöki tervezések, amelyekhez vannak ugyan jól behatárolt és bejáratott algoritmusok, ezek azonban a különböző körülmények között eltérő hatékonyságúak; hogy mikor melyiket érdemes és kell használni, a tervező csak saját ismereteire, tapasztalatára támaszkodva döntheti el. Nagyon sok területen az információk zöme szimbolikus jellegű. Például a gyógyszertervezésben a legtöbb fizikai-kémiai jellemző (halmazállapot, oldhatóság, gyúlékonyság, összeférhetetlenség stb.) szimbolikus jellegű, a feladatok nagysága és bonyolultsága viszont egyre jobban megköveteli a számítógép felhasználását.

A mesterségesintelligencia-kutatások kezdeti eredményei alapján ezeknek a speciális feladatoknak a megoldására a számítástechnika egész új ágazata alakult ki, amely egyelőre még csak fejlődésének kezdeti szakaszában található, de máris vannak figyelemre méltó eredményei. A terület szakértőrendszerek néven vált ismeretessé, s mint már említettük, szoros kapcsolatban van a mesterségesintelligencia-kutatásokkal.

Mi is tulajdonképpen a szakértőrendszer? Alapjában véve egy olyan számítógépes alkalmazási rendszer, amely egy jól körülhatárolt szakterületen (vagy feladattípusban) szaktanácsadást képes nyújtani, esetleg komplex feladatokat képes megoldani felhasználójával szoros együttműködésben. Ami alapvetően megkülönbözteti a szakértő rendszereket a hagyományos dedikált információs rendszerektől, az az, hogy olyan tényekre és eljárásokra vonatkozó ismereteket feltételeznek, amelyeket az emberek szaktevékenységük közben vagy a köznapi életben használnak — sokszor öntudatlanul.

A szakértőrendszereket két fontos tulajdonság jellemzi. A tartalmazott és feldolgozott információ jobbára szimbolikus jellegű, és az alkalmazott módszerek elsősorban heurisztikusak (tapasztalati jellegűek). Más megközelítésben: általános problémakezelés helyett egy-egy szűkebb szakterületre összpontosít (ebből a szempontból tehát dedikált rendszer, mint számos más hagyományos számítógépes rendszer), továbbá szakértői tudást, ismeretanyagot és kiértékelő mechanizmust tartalmaz. A szakértői tudásba beletartoznak a tárgykör szakirodalmában megtalálható tények, fogalmak, összefüggések mellett a szakember tapasztalata, intuitív szellemi tevékenységének megnyilvánulásai is. Elvileg tehát olyan képességekkel is rendelkezhet, mint az észlelés (vizuális, illetve fonikus információk fogadása és felismerése), tanulás, mérlegelés, döntés, cselekvés, szimbólumok, modellek, ábrázolások szerkesztése, illetve kezelése, a tulajdonképpeni gondolkodás, vagyis az a képesség, hogy meglévő ismeretek segítségével a tudás kiterjeszthető új ismeretekkel.

Természetesen a felsorolt képességek számítógépes megvalósítása nagyjából még csak a kutatás fázisában található, és a mesterségesintelligencia-kutatások tárgykörébe tartozik. Hibásnak bizonyult az a kezdeti elképzelés, hogy egypár igen általános

következtetési törvény és nagy teljesítményű számítógépek összekapcsolásából megszülethetnek az emberi teljesítményt megközelítő, esetleg meg is haladó rendszerek. Sorozatos kudarcok után a hetvenes évek második felére kristályosodott ki az a nézet, hogy egy szakértőrendszer hatékonysága elsősorban a tudásbázis, az ismeretanyag függvénye, s kevésbé függ a felhasznált formalizmusoktól vagy következtetési sémáktól. Világossá vált, hogy sokszor egy-egy komplikáltabb feladat megoldása nem is annyira az ismeretek mennyiségének köszönhető, hanem inkább annak a képességnek, hogy mindig a pillanatnyi körülményeknek és feltételeknek legjobban megfelelő ismeretek kerüljenek felhasználásra. Ezeknek a felismeréseknek köszönhetően azután a szakértőrendszerek megvalósítását egészen más megközelítésben látták a kutatók: a fő hangsúlyt az emberi szakértő módszereire, ismereteire helyezték, s csak másodszorban foglalkoztak ezeknek tényleges ábrázolásával is, számítógépre való implementálásával.

A napjainkig megvalósított szakértőrendszerek magja a tudásbázisból és a kiértékelő mechanizmusból áll. A tudásbázis ábrázolására eddig elsősorban a szabály alakú megoldás jellemző: a szakértői tudást "ha (feltételek) ... akkor (következmények)" alakú szabályok képviselik. (Íme egy triviális példa: ha Ágnes nőnemű személy, és ha Éva nőnemű személy, és ha mindkettőnek azonos személyek a szülei, akkor Ágnes és Éva nővérek.) Egy szabály akkor alkalmazható, ha a feltételrészben szereplő összes tény aktuálisan érvényes, jelen van a tudásbázis tényhalmazában (tehát a fenti példában biztosan tudni lehet, hogy Ágnes és Éva egyaránt nőnemű személy és valóban mindkettőnek ugyanazok a személyek a szülei).

Az alkalmazható szabályokat a kiértékelő mechanizmus választja ki az összes szabály közül, s ugyancsak ez a mechanizmus határozza meg, hogy a potenciálisan érvényesíthető szabályok közül — bizonyos elsőbbségi elveket is betartva — melyik kerüljön ténylegesen felhasználásra. A szabály alkalmazásával a következményrészben szereplő tényekkel kibővül a tudásbázis (tehát az, hogy Ágnes és Éva nővérek, tényként rögzíthető), és a kiértékelő mechanizmus kiválaszthatja az újabb aktuálisan alkalmazható szabályokat. A végső következtetés a lehetőségek előre meghatározott körére vonatkozó szabályok összekapcsolásával valósul meg, ugyanis a ciklus addig ismétlődik, amíg olyan szabály kerül sorra, amelynek következménye megoldásnak tekinthető.

A következmények (szabályok) megítélése a feladatkör szakértőinek a feladata, a tudás szerkezetének és a szabályoknak gépi megvalósítását a számítástechnikai szakemberek végzik. Az eddig megvalósított rendszerek körülbelül a következő osztályba sorolhatók: értelmező rendszerek (a külvilágban végbemenő események jellemzőit kiemelik, és ezek alapján szimbolikus osztályokba sorolják az illető eseményeket — ide tartoznak a beszédfelismerő, képelemző, kémiaiszerkezet-leíró, jelértelmező stb. rendszerek); következtető rendszerek (adott szituációk alapján helyzetanalíziseket végeznek, s több-kevesebb eséllyel következtetéseket vonnak le a bekövetkező események esélyeiről — ide tartoznak a meteorológiai előrejelző, demográfiai, közlekedési következtető rendszerek); diagnosztizáló rendszerek (rendszerelemzéses úton behatárolják egyes komponenseknek a normálistól eltérő működését — ide tartoznak az orvosi, folyamatirányítási felügyelő és diagnosztizáló rendszerek, nagy programtermékek működésére felügyelő rendszerek stb.); tervező rendszerek (lehetséges építőelemekből rendszert építenek úgy, hogy közben felügyelnek a feladat speciális megkötéseinek kielégítésére, az esetleges összeegyeztethetlenségek kikerülésére, optimalizálási számításokat végeznek stb.).

A mai napig előállított, első generációsnak tekinthető szakértőrendszerek általában erősen szakosított, az előre nem látható körülményekhez nem vagy nehezen alkalmazkodó, nem rugalmas rendszerek. Az emberre jellemző, hogy ha érdeke úgy kívánja, képes tapasztalataiból tanulni — akár úgy, hogy csak bővíti, akár úgy is, hogy átszervezi a világról alkotott belső modelljét. Ennek a képességnek hiányában a szakértőrendszer nem marad, nem maradhat értelmes, "intelligens" partner. Az elkövetkező generációk megvalósításában valószínűen a tanulási mechanizmusok tanulmányozására, modellezésére kell majd a fő hangsúlyt fektetni.

A mesterségesintelligencia-kutatás alapvetően abban különbözik a számítástechnika többi ágától, hogy figyelmét elsősorban az emberi magatartás, tevékenység modellezésére összpontosítja. A kutatásoknak két fontos irányzata figyelhető meg. Az egyik olyan rendszerek előállítására irányul, amelyek intelligens műveletek elvégzésére használhatók. Ezek elsőrendű célja nem az emberre jellemző módszerek, eljárások hű másolása, hanem bizonyos emberi tevékenységek egyszerű, gyors, olcsó és nem utolsósorban biztonságos elvégzése (tehát intelligenciamodellünknek csupán egyik dimenzióját célozza meg). Ide tartozik az egyszerű robotok előállítása (forrasztóautomaták, festőrobotok, szállítóautomaták stb.), a természetes nyelvek számítógépes fordítása stb. Ugyancsak ehhez az irányzathoz tartoznak azok a törekvések, amelyek a logika modellezése révén közelítik meg a klasszikus számítógép-programozást. Tulajdonképpen ide tartoznak az első generációs szakértőrendszerek is.

A második irányzat legfontosabb célkitűzése a természetes intelligencia teljes modellezése, vagyis az intelligenciamodell háromdimenziós megközelítése, amelyben az ismeret játssza a főszerepet. Egyrészt olyan szerkezetek előállítása, amelyekben a tudás, az ismerethalmaz kifejezhető és rögzíthető, másrészt olyan következtető rendszer megvalósítása, amely az előbbi szerkezetek felhasználásával újabb problémák megoldására használható.

A mesterséges intelligencia területén kifejlesztett formalizmusok még nem kecsegtetnek nagy sikerekkel az alkalmazások szempontjából. Hogy megértsük az említett formalizmus lényegét, meg kell értenünk az ismeretábrázolás problémáit. Ezek elsősorban abból erednek, hogy nem egyszerűen valami idealizált tudást kell ábrázolni, amely a logika törvényeinek engedelmeskedik, hanem a valós világ körülményeinek megfelelő, az emberi agyban tárolt tudásnak a kezelését.

A szellemi feladatok (érzékelés, problémamegoldás, tanulás stb.) gyakran kombinatorikus robbanással járó keresési eljárásokat vonnak maguk után. Hogy a keresést kézben lehessen tartani, megfelelő vezérlési információkra van szükség, amelyek természetüknél fogva nagyrészt heurisztikus, tapasztalati jellegűek. (Amikor például felismerünk egy ismerős arcot egy társaságban, e folyamat lehetetlen volna az illető arcnak minden egyes ismert arccal történő összehasonlítása alapján. Egészen más jellegű eljárásokra, így például valószínűleg képzettársítási, különleges szelektálási stb. folyamatokra van szükség.)

A valós világra vonatkozó ismeretek szükségszerűen bizonytalanok, nem teljesek és belső ellentmondásokat is tartalmazhatnak. A tudásábrázolásra irányuló kutatások feladata olyan struktúrák és következtetési rendszerek létrehozása, amelyek képesek megbirkózni ezekkel a problémákkal. A valós világra vonatkozó ismeretek egyrészt a fizikai világra (okási kapcsolatokra, térre és időre) vonatkoznak, de ugyanakkor olyan elvont dolgokra is kiterjednek, mint hitek, meggyőződések, emberi elgondolások, motivációk, morális és pszichológiai tényezők stb.

Mit ígér a jövő?

A látás, a beszéd és a hallás az emberek közötti kommunikáció természetes módozatai. Ezek a közlési formák azonban a legutóbbi időig nem jelentek meg az ember-számítógép kapcsolatban is. A klasszikus számítástechnika az említett természetes kommunikáció helyett az ember számára elég idegen, mesterséges adathordozókat (lyukszalag, lyukkártya, mágnesszalag, mágneslemez stb.) iktatott az ember és a számítógép közé. Ily módon a felhasználónak speciális ismereteket kellett szerezni a gép felhasználásához, meg kellett tanulni a programozási nyelveket, elő kellett állítani az adathordozókat stb., s ezek a műveletek az emberi erőforrások igen nagy hányadát tulajdonképpen elvonták a tényleges feladatok megoldásától. Nem mellékes újra megemlíteni, Neumann János még 1945-ben, a legelső elektronikus számítógép tervezésének idősa-

kában felismerte a természetes hírközlés fontosságát, s javasolta, hogy az új gépet lássák el képkimenettel!

Az első lépéseket egy természetesebb kommunikáció felé az első rajzgépek, illetve képernyős terminálok (adatvégállomások) megjelenése jelentette. Ezekkel együtt alakult ki s indult fejlődésnek a számítástechnika egyik legdinamikusabban fejlődő alkalmazási területe, a számítógépes grafika.

Az oly gyakran emlegetett szédületes fejlődés oda vezetett, hogy a számítástechnika egyre jobban közüzemi szolgáltatássá kezd válni. Ez azt jelentené, hogy a gép felhasználásához egyre kevesebb speciális ismeretre volna szükség. Első olvasásra komoly ellentmondás vélhető ebben a kijelentésben: egyrészt az egyre fejlettebb, komplikáltabb számítógép, másrészt pedig az egyre kevesebb speciális felhasználói ismeret? Oldódik azonban az ellentmondás, ha arra gondolunk, milyen természetességgel veszi fel a telefonkagylót, kapcsolja be a rádiót vagy TV-készüléket az akár óvodás korú gyermek! Ilyen értelemben kíváncsi tehát, hogy a számítógépeket is speciális ismeretek nélkül, közüzemszerűen használhassa a modern ember. E fontos cél érdekében a modern számítástechnika már tett lépéseket, tényleges megoldása azonban a közeli-távolabbi jövő feladata.

A szintetizált beszéd előállításában már születtek kézzelfogható eredmények, s ezáltal a számítógépek felhasználásának újabb lehetőségei tárulnak fel. (Itt érdemes megemlékezni arról, hogy a mesterségesen előállított beszéd már régebben is izgatta az emberek fantáziáját. Fennmaradt dokumentumok szerint Kempelen Farkas [1734—1804] volt az első, aki ezzel a feladattal érdemben foglalkozott, s bizonyos mértékben sikereket is mondhat magáénak: 1788-ban szerkesztett egy beszélő gépet, amely 3—4 éves gyermek hangját utánozta. Valódi megoldást azonban csak az elektronika kora ígérhetett!)

Az orvosi kutatások számos ágában felhasználható a szintetizált beszéd. A beszéd-audiometria új ága alakulhat ki azzal, hogy a hallásterveztek anyagát szintetizált szavakból állítják össze. A szintetizált szavak egyes akusztikai paramétereinek kiemelésével, időtartamviszonyainak megváltoztatásával vagy egyes frekvenciakomponenseinek áthelyezésével például olyan "betegségorientált" tesztek lehet előállítani, amelyekkel behatárolható a pontos diagnózis. A hallássérültek rehabilitációjára olyan "beszélő" készülékeket lehet szerkeszteni, amelyek segítségével a beteg maga is gyakorolhat. A némák vagy beszéd-sérültek is olyan hasznos segédeszközöket kaphatnak, amelyek billentyűzet segítségével szavakat, mondatokat képesek előállítani és megszólaltatni s azokat például telefonálásnál felhasználni. A vakok életének és munkájának megkönnyítésében is fontos szerepet kaphat a beszélő számítógép, amely a nyomtatott szöveget képes hallhatóvá tenni. Az iparban is számos lehetőség adódik a szintetizált beszéd felhasználására. A postai gyakorlatban számítógéppel előállított beszéddel működtethetők a pontos idő, tudakozó, számváltozást bejelentő szolgáltatások. A szárazföldi, légi és vízi közlekedésben is számos probléma csak a szintetizált beszéd felhasználásával oldható meg. Igen fontos felhasználási területe lehet a beszélő számítógépeknek az idegen nyelvek oktatásában, a helyesírás, a helyes kiejtés gyakoroltatásában.

A természetes nyelvű beszéd feldolgozása még nehezebb feladat elé állította a kutatókat. A hetvenes évek elején tett optimista kijelentések, amelyek szerint a fonetikus írógép problémáit még 2000 előtt megoldják a mesterségesintelligencia-kutatások, nem váltak be teljesen. Nem lehet azt állítani, hogy ezen a területen nem születtek gyakorlati eredmények is. Ami az egyes izoláltan kiejtett szavak felismerését illeti, ma már vannak nemcsak laboratóriumi, hanem kereskedelmi szintű berendezések is. A természetes nyelvű, folyamatos emberi beszéd gépi felismerésében még számos megoldatlan feladat vár megoldásra. Ezek közül a legfontosabbakként említhetők: az akusztikai-fonetikus felismerés, a morfológiai egységek (szavak) körülhatárolása a majdnem folytonos akusztikai jel fonémákra való felbontásának alapján, a morfológiai elemzés és nem utolsósorban — a számítógéppel segített fordításnál már bővebben bemutatott — szemantikai elemzés, azaz jelentésértelmezés.

Még csak kutatási fázisban található, de az ötödik generációs számítógépek konkrét célja az ember-gép kapcsolat vizuális formája. Közismert tény, hogy az ember és környezete közötti kommunikációban a vizuális információ a legfontosabb, minden más felülmúló tömörsége miatt. Minőségi jellemzői az alak, a szín, a mélység és a mozgás.

A látás automatizálása — ide tartozik a vizuális információ felvétele, tárolása, továbbítása, feldolgozása és nem utolsósorban értelmezése — nagyon bonyolult, napjainkban még számos megoldatlan problémát magában rejtő feladat.

A vizuális információ kezelésére az első próbálkozásokat a számítógépes grafika tette, bár egyetlen feladata képek, ábrák előállítása más, nem vizuális természetű információk alapján. Így például a számítógép képernyőjén kirajzolódó függvénykép, bármennyire lebilincselő is, nem jelent más, mint függvényértékek kiszámítását és valamelyik koordináta-rendszerben való ábrázolását. A generált kép vonalas kép, vagyis egyenes és görbe vonalakkból és ilyenekkel határolt idomokból áll (ezek lehetnek természetesen színesek is). A képgenerálást "szintaktikus" leírásnak is lehet nevezni: ez tulajdonképpen egy grafikus nyelven írt program a kép felrajzolására, és lényegében különbözik a látórendszerekben kezelt képleírástól.

A grafikus adatok feldolgozása során az egyik legnagyobb problémát egy meglévő kép grafikus információtartalmának bevitelle jelenti. Mivel a hagyományos, Neumann-elvre alapuló elektronikus számítógép digitális (numerikus) adatokkal dolgozik, a grafikus információk ilyen alakúra kell átalakítani. Rendkívül bonyolítja a helyzetet az a tény, hogy az egyes információelemek halmaza önmagában még nem fejezi ki az illető kép szemantikáját is (nem lehet tudni például, hogy a rögzített pontthalmaz emberi arcot, matematikai függvényt vagy esetleg egy absztrakt festményt képvisel). Igen bonyolult programok szükségesek ahhoz, hogy a képi információ szemantikáját, jelentését is valahogy kódolni lehessen. E feladat tulajdonképpen még ma sincs teljesen megoldva, részleges eredmények azonban már ezen a téren is vannak. Így léteznek már a grafikus információknak numerikus adatokká alakítására szolgáló, úgynevezett digitáló berendezések: ezek egy rajz vagy kép egészét elemi pontokra bontják, majd a pontok pozícióját, tónusát, illetve színét alakítják át numerikus értékekké.

Az alakfelismeréssel, illetve a vizuális információ tényleges feldolgozásával a mesterségesintelligencia-kutatások foglalkoznak, elsősorban a természetes képekben foglalt vizuális információ megértését tűzve ki célul. E nagyon bonyolult folyamatban három minőségi fázis különböztethető meg. A képfeldolgozás során a természetes képek jellemző tulajdonságait kell kiemelni, s ezek alapján kell a nyert információhordozó adatokat a feldolgozás, illetve továbbítás számára megfelelőbb formára átalakítani. A következő fázis az alakfelismerés: ennek során a képet jellemző legkülönbözőbb sajátosságok kiszűrése alapján leírják, majd osztályozzák a képet, illetve annak egyes részleteit. Igen bátorító eredmények ezen a téren is születtek, hiszen olyan ipari robotok is léteznek már, amelyek képesek szerszám- vagy alkatrészhalmazokból alakjuk alapján megtalálni előírt darabokat. A harmadik fázis a képfelismerés, amelynek célja a valós világ leírása. Itt azonban már szükség van a valós világra vonatkozó bizonyos mennyiségű vizuális ismeretanyagra, tehát újra találkozik a mesterségesintelligencia-kutatások általános célkitűzéseivel, módszereivel.

Az ember és a számítógép közötti kommunikációnak ez az új megközelítése, "emberre szabottabb" megvalósítása is eléggé világosan mutatja, hogy a feladatok algoritmikus feltevése és megoldása egyre jobban rákényszerül, hogy átadja helyét az ismeret- vagy tudásbázisú logikai feldolgozásnak. Valóban, az emberek elé tornyosuló megoldandó feladatok igen nagy hányada nem matematikai jellegű, megoldásuk elsősorban logikai következtetések és nem számítások útján érhető el. Ez a tendencia még olyan területeken is egyre jobban előtérbe kerül, amelyeket eddig algoritmikusnak tekintettek. Példának vegyük csak az ipari folyamatirányítás területét.

A nyolcvanas évekig általában az volt az általános törekvés, hogy az egyes irányítási folyamatokat explicit algoritmusokra, zárt modellekre alapozott irányítási stratégiákkal oldják meg. (Az ilyen megoldásokhoz a kezdeti 8 bites, majd 16 bites

mikroprocesszorok kitűnő eszközbázist teremtettek.) A folyamatirányítás elméleti megoldása abból a feltételezésből indul ki, hogy a valóságos folyamatokról alkotott ismereteink, az irányítóberendezések által nyert adatok, valamint a matematikai modellek és algoritmusok pontosak, teljesek és véglegesek. Csakhogy már a nyolcvanas évek elején világossá vált, hogy a fenti feltételezések csak bizonyos tűrések mellett igazak. Az alkalmazott matematikai modellek általában közelítő lineáris modellek, az adatok legtöbbször véges számú kísérlettel meghatározott tapasztalati értékek csupán. És a valós folyamatok igen nagy százalékánál ez utóbbiak a tényleges feltételek. A közelítő megoldások az irányított berendezések működésének csak bizonyos határain belül érvényesek. Ha a működés e határokon kívülre csúszik, az irányításhoz esetleg heurisztikus (tapasztalati) tényezőkre is szükség lehet. Ezeket algoritmizálni viszont reménytelen feladat.

A mesterségesintelligencia-kutatások, illetve a számítógépes nyelvészet már napjainkig is igen bátorító eredményeket mondhatnak magukénak. Ezek közé tartozik a logikai programozás alapjainak lerakása is. Ennek fő jelszava a "mit" és nem "hogyan", ami azt az igényt fejezi ki, hogy a felhasználónak elég legyen csak azt leírni, hogy mit kíván megoldani, és ne kelljen azzal foglalkoznia, hogy ezt a gép hogyan oldja meg. Ezt a követelményt a matematikai logika természetéből fakadóan teljesíti. A logikai programozás alapelve az, hogy maguk a logikai állítások is felfoghatók programnak, ha rendelkezésre áll egy megfelelő végrehajtási mechanizmus, amely az állítások összességét programként kezeli. Ennek megfelelően egy feladat megoldásához az állításokból "logikai programot" (a "mit" részt) kell csak megadni, a megoldás módját (a "hogyan" részt) a végrehajtási mechanizmus szolgáltatja. Az egyik legfontosabb — és legtöbbet ígérő — logikai programozási nyelv a PROLOG, amely már a mai hardver-szoftver adottságok mellett is elég hatékony (természetesen nem túl nagy méretű feladatoknál). Értékeinek tényleges hasznosításához viszont új elvű számítógép-architektúrára van szükség.

A napjainkig ismert négy számítógépes generáció közös jellemzője a Neumann-architektúra, amely a tárolt programban rögzített lépésenkénti, sorozatos aritmetikai-logikai adatfeldolgozáson alapszik. A számítástechnika elé tornyosuló feladatok megoldására ez az architektúra már nem tűnik megfelelőnek, annál is inkább, mivel az elért technológiai-technikai ismeretek sokkal hatékonyabb architektúrák kialakítását is lehetővé teszik. Ezek megjelenése "ötödik generációs" vagy "Neumann utáni" architektúra megnevezéssel évezredünk utolsó éveire várható.

Tartalom

Mi volt eddig?	3
Programozás: művészet — tudomány — technológia	6
Programozási nyelvek	10
Törpe — óriás lehetőségekkel: a mikroprocesszor	14
Operációs rendszerek, programozási környezetek	16
Számítógépes műszaki tervezés és gyártás (CAD/CAM)	20
Számítógépes oktatás (CAI)	24
Számítógépes szövegfeldolgozás, automatikus fordítás	25
Mesterséges intelligencia — szakértőrendszerek	28
Mit ígér a jövő?	31

