

# Infomációmenedzselés

## V. rész: Rendszerszemlélet

*Az előző cikk részlet az integráció bonyodalmaival foglalkozott. Most arra teszünk kísérletet, hogy feltárjuk a valódi integráció titkát. A kulcs egyetlen szó: rendszerszemlélet. Aki az ismereteket menedzselni óhajtja, annak legelőször azzal kell tisztában lennie, hogy az adat és annak feldolgozása absztrakció. Ezt nálunk többnyire sikertelenül hajtják végre, a szakértelem hiánya és/vagy a tervezésre szánt idő szűkössége miatt. A sikeres rendszerhez két dolog szükséges. Egyrészt időt igénylő alapos elemzés. Másrészt az információs rendszerek és azokon belül az adatbázisok általános elvi felépítésének, architektúrájának az alapos ismerete. Cikkünk ennek a modellnek egy részletét mutatja be, megvilágítva az adat, az eljárás és a környezet összefüggéseit, valamint az ismeret három absztrakciós szintjét.*

### Adatbázisrendszer és -architektúra

Az információmenedzselés nélkülözhetetlen tényezője az adatbázis. A róla alkotott nézetek nagyon eltérők. Van aki az 'adatbázis' szó hallatán csak a valamilyen korlátos szempontból összetartozó ismeretek együttesére gondol. Más viszont – éppen ellenkezőleg – kizárólag egy adatbáziskezelő szoftvert lát a szemelőtt. Az ilyen szemléletek korlátosak, és nem ritkán ez az oka annak, hogy az információs rendszer nem jár a várt sikerekkel.

Az adatbázis egyetlen helyes látásmódja szerint az adatbázist magát is rendszerként kell felfogni. Az *adatbázisrendszer* a teljes információs rendszer részeként

- az ismereteknek
- a rájuk vonatkozó eseményeknek és tevékenységeknek (együtt: eljárásoknak)
- az erőforrásoknak
- a felhasználóknak és szervezeteiknek, valamint
- a szabványoknak

a szervezett együttese. Ez a definíció szinte szó szerint megegyezik a másutt a teljes információs rendszerre (a továbbiakban: IR) adott meghatározással. Ez nem véletlen és nem is ellentmondás. Éppen ellenkezőleg. Az adatbázis is az IR része. Ebből következően magát az adatbázist is a felsorolt tényezők szervezett együtteseként célszerű elképzelni.

Az adatbázisrendszer alkotóelemei természetüket tekintve két nagy csoportba sorolhatók. (A szabványokról egyelőre feledkezzünk el.) Az erőforrások, a felhasználók és szervezeteik *konkrétan* létező dolgok. Ezen jelenleg azt kell érteni, hogy az IR fejlesztőjének ezeket a tényezőket figyelembe kell vennie, de nem kell úgymond kitalálnia. Velük szemben az adatok és az eljárások eltérő módokon foghatók fel, fogalmazhatók meg és valósíthatók meg. Ezért elemzés és tervezés tárgyai illetve eredményei. Következésképpen absztrakcióval születnek, tehát *absztrakt* dolgok. Akkor is azok, ha konkrétan tároljuk és kezeljük az adatokat illetve konkrétan hajtjuk végre az eljárásokat.

Mármost az információs rendszerek általában két ok miatt nem eléggé sikeresek. Az egyik ok az, hogy a fejlesztők rossz absztrakciókat alkalmaznak. Például nem korrekten fogják fel a valóság lényegét és helytelenül tükrözik azt az adatokban illetve pontatlanul vezetnek végig a valós eljárásokat az azokat megtestesítő számítógépes programokban. Ez a kisebbik, bár nem jelentéktelen baj. A másik ok és probléma az, hogy – nagy megdöbbenésemre – a legtöbb számítástechnikus egyáltalán nincs tisztában azzal, hogy milyen az

információs rendszerek általános, a konkrét gyakorlati alkalmazásoktól független elvi absztrakt felépítése.

Ennek pedig kétségtelenül az a magyarázata, hogy a számítástechnikusok jó része képtelen elvonatkoztatni az általa éppen alkalmazott eszközöktől, elsősorban a konkrét adatbáziskezelő rendszertől. Mindebből már logikusan következik,

”

**A legtöbb számítástechnikus egyáltalán nincs tisztában azzal, hogy milyen az információs rendszerek általános, a konkrét gyakorlati alkalmazásoktól független elvi absztrakt felépítése.**

”

hogy az eszközben bekövetkező legkisebb változás esetén hozzá kell nyúlni az IR egyéb tényezőihez is és azokat át kell alakítani. Ilyen szemléletben pedig képtelenség tartósan stabil, robusztus (azaz változástűrő), végső soron gazdaságos rendszereket készíteni.

Itt emlékeztetnem kell az olvasót két lényeges momentumra. Egyrészt a változtatás fajlagos költségei az első fejlesztés kiadásainak akár a tízszeresét (!) is elérhetik. Például ha egyetlen adatnak az első megtervezését egy egységnek vesszük, akkor annak a megváltoztatása már tíz egységbe kerül. Másrészt hazánkban a fejlesztők munkaidejének ma már a 70-80 százalékát a módosítások foglalják le. Tessék ezt a két ténytet összevetni és levonni a következtetéseket! Mivel a változtatás iszonyatosan drága, az ismeretekkel csakis akkor gazdálkodunk jól, ha a legelső tervezést alaposan végezzük el és arra több erőforrást áldozunk annak érdekében, hogy stabil rendszereket építsünk.

A rendszerekkel kapcsolatos közismert fogalom az ún. fejlesztési *életciklus*, amely szerint az IR kialakítását problémafeltárás/helyzetfelmérés, elemzés/tervezés, specifikálás, kivitelezés és bevezetés szakaszokra osztjuk. A ciklust az erőforrásigény szempontjából egy görbével, egy diagrammal szokták szemléltetni. A hetvenes években ez a görbe a kivitelezés fázisánál 'púpodosott ki', vagyis akkor azt feltételezték, hogy az erőforrásoknak 50-60 százalékát kell erre a lépésre fenntartani. Már a nyolcvanas években a diagramot átrajzolták úgy, hogy az elemzés/tervezés tevékenységeihez rendelték az erőforrások 40-50 százalékát. Ehhez képest hazánkban a tipikus kép szerint a tervezésre átlagosan 10 százaléknyi energiát fordítanak, miközben a kivitelezésre 60-70 százalékot. Ez magyarul azt jelenti, hogy a kivitelezés során – mi több, olykor a bevezetés közben! – találják ki ad hoc módon, hogy mit is kellene tenni...

Megismétlem a lényegét: a vezetők, a felhasználók, de még a fejlesztők sem ismerik az IR és azon belül az adatbázisrendszer általános elvi képét, vagyis *modelljét*. Azt a mintát, képet, képletet, sablont, prototípust, etalont, amit minden IR tervének követnie kellene. Ezért nem lehet csodálkozni azon, hogy a modelltől fakadó következményekkel és követelményekkel még kevésbé vannak tisztában. Ezen a hiányosságon próbálok meg segíteni sorozatunk részében. Mielőtt ezt

”  
A változtatás fajlagos költségei az első fejlesztés kiadásainak akár a tízszeresét (!) is elérhetik.

◆  
Hazánkban a fejlesztők munkaidejének ma már a 70-80 százalékát a módosítások foglalják le.

◆  
Az ismeretekkel csakis akkor gazdálkodunk jól, ha a legelső tervezést alaposan végezzük el és arra több erőforrást áldozunk annak érdekében, hogy stabil rendszereket építsünk.

”

megtenném, el kell mondanom egy tragikomikus történetet.

Egyik szervező barátomra ráfordmedt a tisztelt felhasználó: „Miért szöszmötöl maga annyit azon a terven! Miért nem látunk már működő programokat?” Információk a mérlegeléshez: a vonatkozó cég addigra már a harmadik számítógépes 'rendszerét' és 'adatbázisát' dobta ki, mert azokat nem tervezték meg rendesen... Barátom azonnal rá is bökött egy hibára: „Jé, itt a közüzemi fogyasztót és a fogyasztási helyet egy kalap alá vették.” Magyarul: a fogyasztási helyhez kapcsolták a fogyasztó ismereteit is. Például az én esetemben az áramfogyasztási hely '1024 Forint u. 3. 5. sz.', a fogyasztó pedig 'Halassy Béla'.

Kollégámnak ezen és a hasonló kitételein a felhasználó elálmélkodott: „És az baj?” Megkérem az olvasót, hogy mélézzen el a felvetésen. A cikk végén majd elárulom a választ. Mindenesetre nagyon elszomorító, hogy a felhasználó azért mérges, mert a fejlesztő az ő érdekében elgondolkodik a jobb megoldáson.

Az adatbázisrendszer általános képét, azaz modelljét más szóval *adatbázisarchitektúrának* nevezzük. Ez a felépítés meglehetősen összetett és megértése némi szakértelmet feltételez. Mivel pedig ez az írás nem csak szakembereknek készül, az architektúrának csak az alapvető vázát fogom ismertetni, remélhetőleg kö-zérthető módon.

#### Az adatbázisrendszer vetületei

A mondanivaló szemléltetésére egy közös példát fogok használni. Számlákról lesz szó, előbb boltiokról, majd általánosokról. Mivel mindenki vásárolt már, feltételezem, hogy a példaválasztás nem szerencsétlen.

Most képzeljük el, hogy vásárolunk egy pár cipőt, amely akciónk azzal zárul, hogy kapunk egy számlát. Azonban ne csak arra gondoljunk, hogy mit fogunk mi tenni azzal a kis cetlivel, hanem arra is, hogy mi minden történik a számlának a boltban maradó másával. Mindezek után a *vetületek* megértése már nem okozhat nehézséget.

A számla adatokat tartalmaz, amelyek a papíron ilyen-olyan szerkezetben, tartalommal és formában látszanak a számunkra. Az adatok szerkezete, azok jelentése illetve alakja együtt jelenti az úgynevezett *adatvetületet*. Azt csak előzetesen jegyzem meg, hogy az

egyik boltban kisebb, a másikban nagyobb cetlit kapunk. Az egyik ilyen, a másik olyan beosztású. Az egyik kézzel írt, a másik géppel kiállított. Mindezekről függetlenül a számla az ugyebár számla, vagyis egy lényeg. És mindaddig, ameddig a bolt ezt a funkciót tölti be, lesznek ott számlák, amikkel foglalkozni kell. Emiatt mondhatjuk azt, hogy az adatvetület viszonylag stabil.

Amikor a bolti eladó a számlát elkészíti, akkor egy tevékenységsort hajt végre, ami lehet egyszerű vagy – a vevő számára sokszor nem is látható módon – összetettebb. Az okosabb számlázógépek nemcsak a papírt adják ki, hanem a forgalmi adót is vezetik, gyűjtik a bevételt, sőt, már arra is akad példa, hogy automatikusan karbantartják a készletet. Tehát maga a számla az csak egy dolog, az pedig már más kérdés, hogy mi minden történik annak adataival. Az azokon elvégzett tevékenységek együttese adja az úgynevezett *eljárásvetületet*. Mivel pedig a számlázási előírások mint speciális szabványok időnként megváltoznak, az alapadatokon végzett műveleteket hol így, hol úgy kell végrehajtani. Ezért mondhatjuk azt, hogy az eljárásvetület viszonylag instabil.

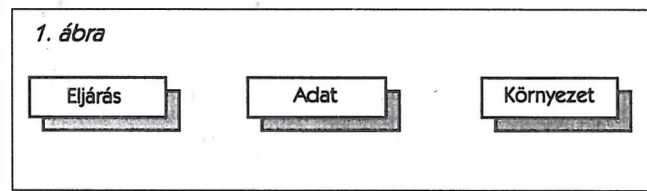
Mármost vannak egyszerű 'kézi' számlázógépek, de akadnak olyanok is, amelyek sajátos számítógépeknek tekinthetők. Nyilván a gép képességeitől (is) függ, hogy a számlaadási tevékenység mi mindent ölel fel. Ebben a tevékenységben eltérő módokon működhet közre maga az ember is. Az egyik boltban a pénztárnál kapjuk meg a számlát, a másiknál viszont a csomagolóban, vagyis az utóbbi esetben a számla több kézen megy keresztül. A tevékenység végrehajtását segítő eszközök és az alkalmazás módja együttesen jelenti a *környezetvetületet*, amit eszköz- illetve alkalmazási részvetületekre szoktunk bontani. A környezetvetület is változékony, instabil. Viszont az eljárásvetülettől abban különbözik – amint erre már utaltam a fentiekben is –, hogy tényezői nem absztrakciók, hanem konkrétan létező dolgok.

Miért van szükség e három vetület szerinti látásmódra, a vetületek megkülönböztetésére és sajátos természetük megértésére? Azért, mert itt a cél, a mód és a tág értelemben vett eszköz hármasáról van szó. Az adatok jelentik a *célt*, mert számlát kell adni, forgalmi adót kell kimutatni, készletet kell vezetni stb. Az eljárás jelenti a *módot*, mert az előző célokat számos eltérő úton érhetjük el. Úgy is, hogy a bolt nappal csak a számlát vezeti, este osztályozza a bevételt és

csak időszakonként készíti készletleltárt. Úgy is, hogy minderre egyszerre kerít sort. Ezt akkor teheti meg, ha van rá megfelelő *eszköz*, azaz jó számlázógépe. Az eszköz mégsem határozza meg sem az adatot, sem a feldolgozást, mert az automatikus készletvezetés a jó eszköz esetében is csak lehetőség, nem előírás.

A fentiekből két fontos tanulságot kell levonni. Az egyik az, hogy az adat nem kötődik szorosan se az eljáráshoz, se a környezethez. Ez az ún. horizontális függetlenség elve. Ezen azt kell érteni, hogy a vetületek egymás mellé rendelve (horizontális) ill. az adat különböző módokon és eszközökkel kezelhető anélkül, hogy attól a lényege megváltozna (függetlenség). A másik az, hogy mivel az adat a viszonylag stabil, a többiekétől független tényező, arra kell törekedni, hogy annak lényegét pontosan ragadjuk meg. Ez az integráció titka. Az IR sikere nem a változékony eljárások aprólékos kidolgozásán és nem a gyakran cserélt eszközökön vagy a szintén változó személyeken, netán a szervezeti felépítésen múlik.

Első, igen egyszerű ábránk a három vetület egymás mellé rendeltségét mutatja. Talán nem véletlen, hogy éppen az adat került középre...



#### Az adatbázisrendszer szintjel

A környezetvetület tényezői a valóság részei, konkrétumok, nem kell és lehet őket kitalálni. Velük szemben az adatok és az eljárások absztrakciók. Ha csúnya szóval akarnék élni, akkor azt is mondhatnám, hogy 'kitalációk'. Például a számlaszám vagy a cikkszám nem

eleve létezik a valós világban, hanem ezeket a dolgokat mi találjuk ki. Ennél a szervezetenél ilyen, annál a cégnél olyan módon. (Talán még emlékeznek az előző cikk részlet egyik kitételére, amely szerint az ismeret – szemben az egyéb erőforrásokkal – mindig többszörös.)

Ezzel az absztrakcióval függ össze a *szint* fogalma. Sajnálatos módon ennek lényegét a vezetők és a felhasználók egyáltalán nem ismerik, mi több, azzal olykor még a szakemberek sincsenek kellőképpen tisztában. A legtöbben úgy gondolkodnak, hogy az adat az adat, ami meg-

Az adat nem kötődik szorosan se az eljáráshoz, se a környezethez.

Ez az ún. horizontális függetlenség elve.

Mivel az adat a viszonylag stabil, a többiekétől független tényező, arra kell törekedni, hogy annak lényegét pontosan ragadjuk meg.

Ez az integráció titka.

jelenik bizonyos papírokon és/vagy tárolják a számítógépen, azután punktum. Ez az egyszintű nézet számos alapvető hiba forrása. Gondoljuk csak át, hogy miért az!

Abból kell kiindulni, hogy az adat a *valóság* tükörképe. A számla azt tükrözi, hogy valaki vett egy pár cipőt, és a cipő éppenúgy valóság, mint az, hogy egy boltban megvásárolták. A cipővel szemben az azt reprezentáló ismeretnek, a cikkszámnak, mindig van sajátos *tartalma* (a cikkszám azt jelenti, hogy...) és reprezentációja, vagyis megjelenési *formája* (a cikkszám úgy néz ki, hogy...). A szintek lényege a valóság, a tartalom és a forma ezen hármából érthető meg. A magyarázatot alulról, a harmadik szintnél kezdem, mert az a legegyszerűbb.

A forma azonnal a szemünkbe ütközik, ha ránézünk a számlára mint cetlire. Az egyik boltban ilyen, a másikban olyan számlát adnak. A papírdarabkákon a jelsorozatok hossza és alakja éppenúgy eltérő lehet, mint azoknak az egymáshoz képest való elhelyezése. Ez a különbség nemcsak papír adathordozók esetében áll fenn, hanem a számítógépeseknél is. Más és más formában ill. elrendezésben lehet a számlaadatokat tárolni az egyik eszközön, mint a másikon. Az ismeretek megjelenítési formája és ún. elhelyezési módja – így nevezik a számítógépen való elrendezést – együttesen jelenti az adat *fizikai szintjét*.

Azért fizikait, mert ekkor az adat már egy konkrét, megfogható, valós eszökhöz kötött. E ponton kell kitérnem az ún. fizikai adatfüggetlenség elvére. Ez az elv kimondja, hogy az adat tartalmának függetlennek kell lennie az adat formájától és tárolási módjától. Tehát ha az adat formája és tárolása megváltozik (fizikai), de jelentése nem, akkor nem volna 'ildomos', ha a programokat e változások miatt át kellene alakítani (függetlenség). Ma már léteznek olyan adatkezelő rendszerek, amelyekben a tartalom és a forma szétválasztható. Vagyis lehetőség van az olyan programozási megoldásokra, amelyeknél pl. az adat hosszának vagy típusának a változása nem igényelné sem az adat újratárolását az új formában (ezt hívják az adatbázis fizikai átszervezésének), sem az azt kezelő programok átírását.



Lehetőség van az olyan programozási megoldásokra, amelyeknél pl. az adat hosszának vagy típusának a változása nem igényelné sem az adat újratárolását az új formában, sem az azt kezelő programok átírását.



A mai korszerű adatkezelők képességeinek dacára a mai magyar programozók igen nagy hányada még mindig a hatvanas évek gyakorlatát követi.



A képzetlen programozó be tudja beszélni az adatbázisarchitektúrát szintén nem ismerő tájékozatlan felhasználónak, hogy „ezt csakis így lehet”. Egy frászt!



„Isten, óvd a királyt!” A felhasználó-királyt azoktól a programozóktól, akik 'csodálatos és ravasz' saját megoldásokat alkalmazva a felhasználók adatait önkényesen úgy értelmezik át, hogy a speciálisan tárolt adatok egyes bitjeinek/bájtjainak saját jelentéseket tulajdonítanak. Ebből azután logikusan következik az, hogy a legkisebb változásokor is azonnal az adatbázis újraszervezése és a programok átírása válik szükségesé, vagyis elveszik a függetlenség. Szomorúan kell megállapítanom, hogy a mai korszerű adatkezelők képességeinek dacára a mai magyar programozók igen nagy hányada még mindig a hatvanas évek gyakorlatát követi. Akkor valóban szükség volt a legapróbb bitek/bájtok rafinált kihasználására is. Ma viszont nem a tár, hanem a változtatás jelenti az igazi költséget. Ami azért növekedik egyre jobban, mert a képzetlen programozó be tudja beszélni az adatbázisarchitektúrát

szintén nem ismerő tájékozatlan felhasználónak, hogy „ezt csakis így lehet”. Egy frászt! Lehet másként is, lehet sokkal jobban is. A változások elkerülhetők, mert a fizikai szint már nem jelent megkötést.

Az ismeret lényege független a formájától, de nem független a felhasználójától. Például más az áfás számla tartalma, mint az áfa nélkülié. (A felhasználó ebben az esetben a vásárló!) Továbbá az nem biztos, hogy minden számlán megjelenik a vásárolt termék cikkszáma vagy a számlaazonosító. Azonban az bizonyos, hogy a számla ismereteinek a további kezeléséhez a kereskedő mint felhasználó valahol mindig kell, hogy vezesse ezeket az adatokat is. Például a könyveléshez és a készletkarbantartáshoz. Tegyük mindehhez, hogy nemcsak az alkalmazás függvénye az, hogy a számlán milyen ismeretek vannak, hanem a használt eszköze is. Kevés boltban tudnak géppel kiállított áfás számlát adni – az ilyen ma még sokszor kézzel készül.

Az ismereteknek az alkalmazási- és eszökökörtől függő tartalma jelenti az adatok *logikai szintjét*. Vagyis a számla az ugyan mindig számla, de az aktuális tartalma a felhasználói igényektől és a használt eszököktől függően más-más lehet. A logikai az ismeretek második, középső szintje.





Még beteszünk egy plusz jelet,  
annak egyfajta tartalmat  
tulajdonítunk ha esik; mást, ha fúj;  
arra meg nem gondolunk, hogy  
egyszerre eshet is, meg fújhat is.  
Csak annyi bizonyos, hogy az egész  
adatbázist fújhatjuk úgy,  
ahogyan van!



számlaszámnak mint adatnak is van tartalma, de a mögöttes két jelenségnek – a vevőnek és a számlának – is van jelentése. Az ismereteknek az alkalmazási és az eszközkörnyezettől független valós tartalma jelenti az adatok *fogalmi szintjét* és ez az első, a legfelsőbb szint.

Itt ismét meg kell állnunk egy pillanatra. A számítástechnikusok maguktól is képesek arra, hogy a legképtelenebb módon rendezgessék el a fogalmakat az adatokban. Itt ilyen ürüggyel

két vagy több fogalmat egy adatba gyömöszölnek (lásd számlaszám), ott pedig olyan 'ravasz' megfontolással egy fogalmat több adatba szórjanak szét. Mindez azonban semmi ahhoz képest, ahogyan a felhasználó mindent megtesz annak érdekében, hogy az adatokban totális fogalmi káoszt állítson elő. Elegendő itt-ellenem a furfangos kódokat, például a cikkszám felépítését. Ha annak első jele '1', akkor vasról van szó, ha '2', akkor alumíniumról. De ha a cikkszám első jele '9' és a második '5' és '7' közötti, akkor szintén vas az anyag. Ha a második jegy '4', akkor forrasztási technológiát alkalmaznak – és így tovább.

Nem folytatom. A felhasználók egyetlen vágya, hogy akár egy tucat fogalmat (alapanyag, forma, technológia, szín, ár, változat stb.) is egyetlen adatba (cikkszám) gyömöszöljenek. Az ilyen megoldás hármass következményének a megjósolásához nem kell madárjósna lenni. Először is a cégnél előbb-utóbb egyetlen lélek sem akad, aki tudná, hogy a cikkszám általában miképpen épül fel és annak egy-egy karaktere milyen fogalmi lényegyet is takar. Másodsor ezt azért jelenthetem ki ilyen bátran, mert az egyes karakterek 'tárolóképessége' előbb-utóbb kimerül, tehát a jelet az eredetitől eltérő célra kezdik használni. (Lásd a fentebbi példát, ahol a második jel kezdetben a technológiára volt fenntartva, de a 'kifogyás' miatt azt már az anyag megjelölésére kezdték használni.) Harmadrészt egy idő után kiderülnek az ellentmondások és megoldhatatlan módon bővülnek a jelentések. Az olvasó nyilván észrevette, hogy a fentiek szerint a '95' cikkszámkezdetű vas kütyük forrasztási technológiával nem készülhetnek. Ma. Holnap már igen. És akkor elkezdődik a cikkszám formai és tartalmi átpofozgatása... Még beteszünk egy plusz jelet, annak egyfajta tartalmat tulajdonítunk ha esik; mást, ha fúj; arra meg nem gondolunk, hogy egyszerre eshet is, meg fújhat is. Csak annyi bizonyos, hogy az egész adatbázist fújhatjuk úgy, ahogyan van!

Elnézést kérek a picit paprikás kijelentésekért. Mindig felbosszant a felhasználók végtelen és önkínzó

Mindeddig nem volt szó újszerű, falrengető dolgokról. Hiszen a józanul gondolkodó ember azonnal beláthatja, hogy a számlaszám lényege (logikai szint) független attól, hogy az a papíron hol jelenik meg és vajon numerikusan vagy karakteresen tárolják-e (fizikai szint). A harmadik szint – és főleg annak a másodikkal alkotott viszonya – már gondokat okozhat.

Az ismeret a valóság tükörképe már azt megelőzően is, hogy azt valamilyen eszközön

adott tartalommal és formában rögzítenénk! A számláról, a cikkről, a számlaszámról, az árról stb. anélkül is lehet *beszélni*, sőt beszélünk is, hogy valahol *léírt* adatokra gondolnánk! Ha például a barátommal a kocsijáról diskurálok, akkor annak nem feltétele, hogy előszedje a forgalmi igazolványt és a gépkocsi típusát egy adott helyen és módon rögzített adatként képzeljük el. A rögzített és értelmezhető ismeret – adat. Viszont a mindennapi kommunikáció fogalmak útján történik. Ezért az ismeret nem akkor kezd élni, amikor adat formáját ölti, hanem akkor, amikor megállapodunk abban, hogy mit jelent a számlaszám vagy éppen a gépkocsitípus.

Mármost ez a megállapodás a mindennapos beszélgetésben szintén nagyon fontos, nehogy fellépjen a „Jónapot, sógor!” effektus. Ha azt mondom neked, hogy az öltönyöm desszinje ez meg az, akkor atrocitásoknak leszek kitéve, mert *fogalmad* sincs arról, hogy mi az a 'desszin'. Ha pedig azt közlöd velem, hogy a gépkocsid 'Lada' típusú, akkor nem leszek elégedett, mert nem tudom, hogy azon belül milyen, mert számomra a 'Vaz 2104' jelenti a típust. Köznapi kommunikációink során a fogalmak tisztázása fontos, de nem jelent gondot, mert egyszerűen megkérdem, hogy „Jó, jó Lada, de milyen?” A számítógép még nem tud így visszakérdezni és szerintem erre sohasem lesz képes, mert a gép nem gondolkodik, mert nincsenek fogalmai.

Mindezt nem véletlenül kellett elmondanom. Ha a fogalmak és az adatok kölcsönösen és egyértelműen megfelelnek egymásnak, akkor persze feleslegesen jártatnám a számat. Ámde a valóságban más a helyzet. A fejlesztők sokszor egy fogalmat több adatban tükröznek illetve egy adatot többféle fogalom tükrözésére használnak fel. Az utóbbira kiváló példa a nem bolti számlaszám, amennyiben annak első pár jele a vevőre, a további magára a számlára utal. Lám itt két fogalom (vevő, számla) vált egyetlen adat tartalmává.

Tehát nem mondhatjuk egyszerűen azt, hogy a fizikai szint a formát, a logikai pedig a tartalmat jelenti. Azért nem, mert nem egyféle tartalomról van szó! A

csacsisága. Igenis önkínzó! Ha ugyanis a cikkszámot át kell alakítani, akkor... Akkor át kell szervezni az egész adatbázist. Át kell írni 999 programot. Újra kell nyomtatni ugyanennyi bizonylatot. Ki kell oktatni ezernyi felhasználót az új értelmezésre. Folytassam? Eláruljam, hogy az azonosító egyetlen jelének a módosítása mennyibe kerül? Elárulom! A General Motorsnál az alkatrészsorszám változtatása 1975-ben kereken 70 millió dollárt kóstált! Rá is tértek azonnal az egyszerű, nem beszélő alkatrészzazonosítóra úgy, hogy az alkatrész jellemző jegyeit külön-külön leíróadatokban rögzítették. Úgy, hogy egy-egy adat egy-egy valós fogalomnak (anyag, forma, technológia stb.) felelt meg. Azóta boldogan élnek.

### Mibe kerül?

Ezt a passzust akár le is zárhatnám abban a tudatban, hogy sikerült megvilágítanom az olvasó számára az adatok fogalmi tisztaságának a fontosságát. Azt, hogy a fogalmak és az adatok között csak szigorúan egyértelmű és kölcsönös viszonyoknak szabadna létezniük. Ez ugyanis az ún. fogalmi adatfüggetlenség záloga. Az IR eme kellemes vonásáról csak akkor beszélhetünk, ha az adatok fogalmi tartalmában bekövetkező mennyiségi változás nem vonja maga után a minőséget, az adatok átszerkesztésének az igényét is. Tehát ha az eredetileg elképzelt mondjuk 10 ezer vas alapanyagú alkatrész helyett 11 ezer ilyen tételünk lesz, akkor nem kell a cikkszámot átalakítani, átértelmezni; az adatbázist átszerkeszteni; a programokat átírni stb. Felesleges is emlékeztetnem arra, hogy itt milyen komoly pénzekről van szó.

Az alcím mondanivalóját mégsem zárhatom le. Ennek több oka is van. Egyrészt döbbenetes a számomra, hogy a fejlesztők mennyire felelőtlenek. Most említhetném azt a szervezőt, aki nálunk egy bizonyos biztosítási módozat számítógépes adatsorából egyszerűen kifelejtette a 'vizontbiztosítás' ismereteit. Kifelejtette? Mint kiderült, „fogalma sem volt” arról, hogy a vizontbiztosítás mit is jelent! Visszautalhatnék a cikksorozat előbbi részére, amelyben a programozócska azt feltételezte, hogy egy cégnek csak egy bankszámlája lehet. Ergo fogalma sem volt a cégek és a bankszámlák viszonyának a lé-

nyegéről. Ám ez nem gátolta meg abban, hogy több milliót (!) felvegyen egy soha el nem készült 'rendszerért'.

Másrészt igen kellemetlen tapasztalataim vannak a felhasználókkal is. Az illető egészen egyszerűen nem tudja megfogalmazni, hogy mit is akar, mert nincsenek világos és tiszta fogalmai a saját munkájához szükséges ismeretekről sem. Saját cégemnél – értsd: az engem foglalkoztatónál – egy biztosítási módozat számítógépes rendszere bevezetésének a napján (!) telefonált az érintett felhasználó, hogy jaj Istenkém, a pam-pam adat nem is azt jelenti, és azt valójában nem is úgy kellene kezelni, és különben is, nomeg...

A számítógépes bűvölet jegyében a felhasználók a fogalmak, a lényegek tisztázása előtt képesek arra, hogy 'rendszer' készíttessenek a fejlesztőkkel. Kerül amibe kerül. Nem egy céget ismerek, amelynek az 'adatbázisai' telis tele vannak szemét, használhatatlan adatokkal. Amelyek tárolása, kezelése évente milliókat visz el. Vajon mit kezdene az olvasó például azzal az ügyféllel, amelynek a neve nemes egyszerűséggel 'Ifj'? Lakcíme pedig ' '?

Végezetül arra kell felhívnom a figyelmet, hogy nemcsak az adatok, hanem az eljárások is absztrakciók. Emiatt nemcsak az előbbieket, hanem az utóbbiak esetében is különbséget kell tenni a fogalmi, a logikai és a fizikai szintek között. A különbségeket és az összefüggéseket az OTP-számla példájával fogom megvilágítani.

Tegyük fel, hogy a valóságban én egy pénzösszeget akarok elhelyezni egy OTP-számlán azért, hogy e befizetést majd adott jogcímű kifizetések kövessék. Ez a

vágyam információs eseményekben és tevékenységekben fog megnyilvánulni. Ha én már az OTP partnere vagyok, akkor fogalmilag csak egyetlen eseményről van szó, nevezetesen az 'új befizetésről'. Ha még nem, akkor *fogalmilag* két eseményben kell gondolkodni: 'új partner' és 'új befizetés'. Azért kell így tenni, mert a partner és a befizetés fogalmilag két külön dolog, amit kétféle adatsorral kell tükrözni.

A befizetés az befizetés. *Logikailag* a fejlesztő gondolkodhat úgy is, hogy az 'új partner' és az 'új befizetés' váltson ki két külön tevékenységsort, ebben a sorrendben. Azonban azért sem lehetne elítélni, ha a kettőt egy műveletsorba vonná össze. A jó fejlesztő az eseményeket és a te-



### A General Motorsnál az alkatrészsorszám változtatása 1975-ben kereken 70 millió dollárt kóstált!

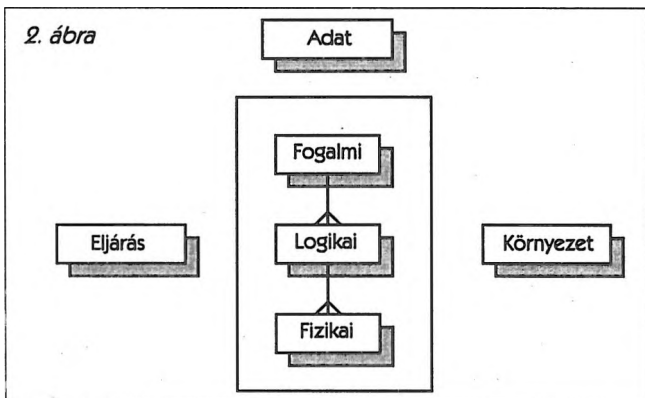


A számítógépes bűvölet jegyében a felhasználók a fogalmak, a lényegek tisztázása előtt képesek arra, hogy 'rendszer' készíttessenek a fejlesztőkkel. Kerül amibe kerül. Nem egy céget ismerek, amelynek az 'adatbázisai' telis tele vannak szemét, használhatatlan adatokkal.



vékenységeket is elemzi, és a tervezése nem pusztán a részletek meghatározására terjed ki. Írhat három programot úgy is, hogy az egyik a partner bevitelét, módosítását, törlését kezeli, a másik a számláét, a harmadik pedig a befizetését. Írhat három programot úgy is, hogy az egyik a partner, a számla, a befizetés bevitelére, a másik ezek módosítására, a harmadik pedig ezek törlésére szolgál. Mindegyik megoldásban van ráció. A körülmények alapos mérlegelése után kell választani a fogalmilag különálló események/tevékenységek logikai elrendezéséről.

A fizikai szintre itt nem térek ki. Egy műveletsor számtalan programnyelvben és igen eltérő technikával valósítható meg. Nos ez az a szint, ami végleg nem tartozik ránk. Elvégre vagy jó a programozó és akkor bízunk meg benne, vagy nem és akkor küldjük el.



Az adatbázisarchitektúra vetületek és szintek szerinti elrendezését mutatja a 2. ábra úgy, hogy az eljárásvetület szintjeit nem tartalmazza. A középpontban továbbra is az adat áll, mert nem a programok logikai/fizikai szervezése, hanem az adat korrekt meghatározása a döntő.

### Ideiglenes összefoglaló

Az adatbázisarchitektúra és az egészséges rendszer szemlélet kifejtését ebben a cikkben nem sikerült befejeznem. Folytatás következik. Nagyon remélem, hogy az olvasó már érzékeli: az adatbázisrendszert nem lehet egyoldalúan szemlélni. Azt, hogy lassan-lassan rádöbben arra, hogy az adatokat, azok feldolgozását és a körülményeket – a környezetet – meg kell tanulni egymástól független lényegekként is szemlélni. Még hozzá szintekben gondolkodva.

Miért akkora a vágyam, hogy ezt a látásmódot az olvasónak átadjam? Azért, mert nap mint nap látom a végzetes félreértéseket. Több oldalról is. Látom, hogy az eszköz miként veszi át az uralmat a lényeg, az ismeret tartalma fölött. Látom, hogy miként csúszik át a hatalom az ismeretekhez értők/használók kezéből

az azokhoz nem konyító programozókéba. Látom, hogy az alapvető fogalmakban már két felhasználó sem tud megegyezni és ezért mindegyik saját rendszert, saját adatbázist, saját eszközt óhajt. Nem közös habostortát, amiből mindenki vehet, hanem olyat, amelyet a saját zúgba félrevonultan felhabzsolhat.

Nem ért sokat az informatikához az, aki nem látja, hogy a számítógép – ha hagyják – az emberekből kemény, dacos, sarokbavonuló, riadt tekintetű, szorongásos és önző lényeket farag.

És mindezek következtében látom a változások hatásainak a lebecsülését; a tartós értékek semmibe vételét; az emiatt fellépő óriási költségeket, a feleslegesen elpazarolt pénzeket. Az információ menedzselésének a tökéletes képtelenségét és ami még ennél is fájóbb, a közös tudat, az egymásra való támaszkodás és az egymásban való bizakodás, hit elvesztését. Nem ért sokat az informatikához az, aki nem látja, hogy a számítógép – ha hagyják – az emberekből kemény, dacos, sarokbavonuló, riadt tekintetű, szorongásos és önző lényeket farag.

Még adós vagyok az első passzusban említett rébusz megfejtésével. Vajon mi köze van az én adataim megváltozásának a fogyasztási helyhez? Egészen világos, hogy a fogyasztó és a fogyasztási hely két teljesen eltérő valós fogalom, jóllehet kapcsolatban állnak egymással. Mármost én ezt nem teszem, de holnap elköltözhetnek a 'fogyasztási helyről'. Vajon mi lenne ennek az információs következménye? Az, hogy mivel a fogyasztó adatait a fogyasztási hely adataival 'összedrótózták', elköltözésem után fogyasztóként – részben jogosan – azt a valakit tüntetnék fel, aki megvette tőlem a lakásomat. Miután az én adataimat így elveszítették – hiszen a fogyasztási helynél ártírták a fogyasztó adatait – a fogyasztási hely új tulajdonosán próbálják majd behajtani az én fogyasztási tartozásaimat is! Ez a 'malőr' bizony csak úgy kerülhető el, ha a két különálló valós jelenség – fogyasztó és fogyasztási hely – adatait külön, de egymáshoz kapcsoltan vezetik. Barátomnak tehát igaza volt.

Viszont mit tehet a jó informatikus a rossz megoldásokat forszírozó felhasználókkal? Akik gyorsan elkészülő *szoftvert*, nem pedig tartósan igaz és érvényes *rendszert* várnak tőle?

Szerzőnk a közgazdaság- és informatikai tudományok kandidátusa