

## A közigazgatásban használt szoftverek komplex hatásai

A cikk célja az L-PEST elnevezésű elméleti modell bemutatása, amely a közszféra szoftverhasználatát által befolyásolt területeket kívánja integrálni és komplex módon kezelhetővé tenni. A szerző elméleti megfontolások és külföldi esettanulmányok alapján megállapítja, hogy a közszféra saját fogyasztása révén és szoftverhasználati szokásain keresztül jelentős hatást gyakorol az oktatásra, az ügyintézésre és az egész társadalom viselkedésére ezen a területen.

**Kulcsszavak:** *nyílt forráskód, közigazgatás*

### Szerzői információ:

**László Gábor**

Műszaki menedzser, egyetemi tanulmányait a Veszprémi Egyetemen végezte, majd a Budapesti Gazdasági Főiskola Külkereskedelmi Főiskolai Karán szerzett oklevelet. Jelenleg a BME Műszaki Menedzsment Doktori Iskolájának hallgatója. Kutatási témái: információstratégiák, nyílt forráskódú szoftverek felhasználása az oktatásban. A Budapesti Műszaki Főiskola Keleti Károly Gazdasági Karán működő Információs Társadalom Oktató és Kutató Csoport (ITOK) vezetője. Oktatóként az e-kereskedelem, az informatika és társadalom, valamint a döntéstámogatási rendszerek témaköreiben vezet kurzusokat.

E-mail: laszlo.gabor@kgk.bmf.hu

### Így hivatkozzon erre a cikkre:

László Gábor. „A közigazgatásban használt szoftverek komplex hatásai”.

*Információs Társadalom* VII, 2. szám (2007): 93–106.

<https://dx.doi.org/10.22503/infvars.VII.2007.2.6>

*A folyóiratban közölt művek*

*a Creative Commons Nevezd meg! – Ne add el! – Így add tovább! 4.0*

*Nemzetközi Licenc feltételeinek megfelelően használhatók.*

László Gábor

# A közigazgatásban használt szoftverek komplex hatásai

„Whoever controls software, controls life.”  
Eben Moglen

## Bevezetés

Az információs és kommunikációs technológiák (IKT) átformálták az egész társadalmat. Használatuk beépült a mindennapi életbe, a gazdaságba és a kormányzati munkába is. Az IKT-eszközök előnyeiről a szakirodalom sokrétűen beszámol, azonban az életminőség vagy akár a termelékenység<sup>1</sup> növekedéséhez történő hozzájárulásuk számszerűsíthetőségének problematikája továbbra is fennáll. Az előnyök mellett azt is figyelembe kell venni, hogy az IKT-eszközök újratерemthetik a társadalmi megosztottságot. A digitális megosztottság kérdésköre több szinten is felmerül, hiszen a potenciális munkavállalók képzettsége (digitális írástudása) ugyanúgy érinti a gazdaságot, mint az állampolgárok részvételét az e-demokráciában, a demokratikus folyamatokban.

Az utóbbi időben a figyelem középpontjába kerültek a nyílt forráskódú alkalmazások: rangos nemzetközi tudományos folyóiratok különszámokat szentelnek a nyílt forráskódú szoftverek<sup>2</sup> gazdasági, politikai és társadalmi hatásainak, és az ENSZ is többnapos szimpóziumot szervezett a témában.<sup>3</sup> 2007-re a nyílt forráskódú szoftverekről szóló hírek az IT-híradások fősodrába kerültek. Magyarországon az Információs Társadalom és Trendkutató Központ e-kormányzati kutatócsoportja – a Miniszterelnöki Hivatal megbízásából – már 2002-ben bekapcsolódott a szoftverekkel kapcsolatos kutatásokba, az „E-kormányzat első kézből” című kutatási projekt részeként. A kutatás keretében dolgoztam ki a „Nyílt forráskódú szoftverek” című zárótanulmányt. Z. Karvalics László „jövőbelátó” képességét dicséri a doktori dolgozatom témaválasztásánál nyújtott segítsége is.

Az ebben a cikkben ismertetett kutatás fő célja a közigazgatás által használt vagy támogatott szoftverek társadalmi-gazdasági hatásainak együttes vizsgálatára alkalmas modell felállítása volt, egyik speciális célkitűzése pedig a nyílt forráskódú szoftverek<sup>4</sup> és

<sup>1</sup> Solow 1987-ben mondta: „Manapság mindenütt a számítógépek koráról van szó, kivéve a termelékenységi statisztikákban.” Solow modelljéből az következne, hogy az IKT-ba történő befektetés a termelékenység és az output növekedéséhez vezet. In Triplett, J. E. (1999). The Solow productivity paradox: what do computers do to productivity? *Canadian Journal of Economics Review*, Vol. 32, No. 2 April, 309–334. [http://www.csls.ca/journals/sisspp/v32n2\\_04.pdf](http://www.csls.ca/journals/sisspp/v32n2_04.pdf) [2 February 2003]

<sup>2</sup> A nyílt forráskódú szoftverek megjelölésére több általánosan elfogadott rövidítés alakult ki. A tanulmány az OSS-rövidítést használja. FLOSS: Free/Libre/Open Source Software.

<sup>3</sup> UNITAR Symposium on ICT Policy Issues for Development [http://www.unitary.org/en/symposium\\_materials.html](http://www.unitary.org/en/symposium_materials.html)

<sup>4</sup> A nyílt forráskódú szoftverekkel kapcsolatos tudományos kutatások kezdetben magukra a fejlesztői közösségekre irányultak, ma már azonban a kormányzati felhasználás is a kutatók és a gyakorlati szakemberek figyelmének középpontjába került. Az Európai Unió is több ilyen kutatást támogatott. Ennek ellenére a nyílt

a zárt, tulajdonosi szoftverek összehasonlíthatóvá tétele tudományos szempontok figyelembevételével. Ebben a cikkben a kutatás háttérét és formálódó eredményeit kívánom bemutatni.

## A kutatás módszerei

A kutatási terület a társadalomtudományi és gazdasági megközelítés mellett műszaki ismereteket is igényel. Összességében azonban a *policy* típusú megközelítést alkalmazom.

Az anyaggyűjtés forrásait elsődlegesen az interneten található kutatási beszámolók, tanulmányok, konferenciaanyagok, valamint az *online* elérhető külföldi folyóiratok, adatbázisok alkották. A kutatás folyamán azonban gyakran váltottam e-mail üzeneteket is mind az akadémiai szféra hasonló vizsgálatokat folytató külföldi képviselőivel, mind pedig a gyakorlati oldal szakembereivel.

Nehéz volt megtalálni a nemzetközileg is elfogadott kutatásmetodikai eszköztárat: a megfelelő vizsgálati módra főként a nemzetközi konferenciákról érkezett visszajelzések világítottak rá, amelyek sokat segítettek munkám továbbfejlesztésében.

A kutatás módszere: longitudinális esettanulmányok feldolgozása. Az esettanulmányok kiválasztásakor különös hangsúlyt fektettem arra, hogy a minta megbízható és érvényes legyen, és egyoldalú esetek önkényes kiválasztása ne okozzon torzulást.

A modellalkotás során igyekeztem áttekinteni az elméleti alapokat és az információstratégiák korai szakaszára, a nemzeti információstratégiákra és a közsféra informatizálására összpontosítottam a figyelmet, azonosítva a közsférában felmerülő speciális stratégiai igényeket. A szoftver felértékelődésével párhuzamosan jelentek meg a nyílt forráskódú szoftverekkel (*open source software, OSS*) kapcsolatos első elgondolások a kormányzati programokban. Az ezek támogatására irányuló kezdeményezésekről készült esettanulmányok értékelése volt a következő lépés, majd – a szakirodalmi hiányosságok áthidalására – áttekintő országtanulmányokat készítettem, amelyekben a közsféra szoftverhasználatot igénylő területeinek azonosítására és e területek kölcsönös összefüggéseinek feltárására törekedtem.

## Háttér – a kezdetek

Az információstratégiák a korai szakaszban szinte kizárólag a hardverek beszerzésére és a hálózat kialakítására irányultak. A fejlődés azonban azt eredményezte, hogy az eszközöket működtető szoftverek (amelyeket a gyártók kezdetben a hardver részeként szállítottak) ugyanolyan fontossá váltak, mint maguk az eszközök. A szoftverek története viszonylag rövid időre tekint vissza, ám ezen belül a nyílt forráskódú szoftve-

---

forráskódú szoftvereknek a közigazgatásban történő alkalmazásáról, illetve annak hatásairól nincs elég adat és eredmény. A nyílt forráskódú szoftverekkel kapcsolatban nemzetközi szinten és Magyarországon is sok negatív prekoncepcióval, ismerethiánnyal és téves értelmezéssel lehet találkozni. A legfrissebb tanulmány: The impact of Free/Libre Open Source Software on innovation and competitiveness of the European Union <http://www.flossimpact.eu/>



reké a leghosszabb múlt. A számítástechnika korai szakaszában – amikor a hardvert és a szoftvert gyakran együttesen kezelték – a fejlesztés a szabad forráskódú szoftver modelljén nyugodott (Bessen 2002). Amikor az 1960-as években az IBM és más cégek piacra dobták első nagy teljesítményű kereskedelmi számítógépeiket, a hozzájuk tartozó szoftvereket a felhasználók megoszthatták a többi felhasználóval, rendelkezésre állt a forráskód, s ezáltal módosítani és javítani lehetett a programot. Az 1960-as évek végén és az 1970-es évek közepén megváltozott a helyzet: a felhasználók nem terjeszthették többé szabadon a programokat, és a forráskód sem volt már elérhető, tehát nem volt lehetőség a programok módosítására. Megszülettek a zárt vagy „tulajdonosi” szoftverek (*proprietary software*), és hamarosan ezek kerültek domináns helyzetbe a szoftverpiacon, úgyhogy sokan ezeknek az alkalmazását kezdték elismerni az egyetlen lehetséges felhasználási modellként.

A világ legnagyobb szoftvervásárlói általában a kormányzatok, az államigazgatási szervezetek. A közszféra mint modellfelhasználó viselkedése nemcsak gazdasági hatásokat fejt ki a piac közvetítésével, hanem befolyásolja a társadalom egészének a magatartását is. A kormányzatok által beszerzett és használt szoftverek hatása a mindennapi élet szinte minden területét érinti: befolyásolja az állampolgárok bevonását és részvételét a közéletben, a közszféra átláthatóságát, a nyílt kormányzást és a közérdekű adatokhoz való hozzáférést csakúgy, mint a digitális archívumok létrehozásának kérdéskörét. Napjaink információs társadalmaiban a biztonság és a bizalom központi kérdéssé vált. Az információs eszközök és technológiák központi szerepet játszanak az országok infrastruktúrájának kialakításában és működtetésében, így ezek biztonsága kulcsfontosságú. A kormányzati szolgáltatásokkal vagy éppen az elektronikus szavazórendszerrel kapcsolatos állampolgári bizalom kiemelt jelentőséget kapott.

## Szabad szoftverek (free software)

A mai nyílt forráskódú mozgalom alapjait Richard Stallman, a *MIT Lab* korábbi programozója teremtette meg, aki 1984-ben elindította a *GNU* projektet (a *GNU* rekurzív mozaikszó, jelentése: „*a GNU nem UNIX*”), és 1985-ben megalapította a Szabad Szoftver Mozgalmat (*Free Software Foundation*). Elsődleges célja egy szabadon felhasználható operációs rendszer kifejlesztése volt. A *GNU GPL*-licenc célja nemcsak az volt, hogy a védőernyője alatt fejlesztett *GNU* ingyenes maradjon, hanem az is, hogy egyre több szabad szoftver fejlesztését mozdítsa elő.<sup>5</sup>

<sup>5</sup> A *free software* kifejezésben szereplő angol *free* szó kettős jelentésű: nemcsak azt jelenti, hogy „szabad”, hanem azt is, hogy „ingyenes”, ami félreértésekre ad módot. A szabad szoftver kifejezésben a szabad nem az árra, hanem a felhasználók szabadságára utal. Azt jelenti, hogy a felhasználók jogosultak futtatni, másolni, közzétenni, tanulmányozni, megváltoztatni és tőkéletesíteni a szoftvert.

• A felhasználók négy különböző joga:

- A jog arra, hogy futtassák a programot, bármilyen céllal.
- A jog arra, hogy tanulmányozzák a program működését, és azt a szükségleteikhez igazíthassák. Ennek előfeltétele a forráskód elérhetősége.
- A jog arra, hogy másolatokat tegyenek közzé a többi felhasználó segítése érdekében.

## Nyílt forráskódú szoftverek (open source software)

A szabad szoftveres mozgalmat az üzleti élet – túlzottan ideológiai alapú megközelítése miatt – fenntartásokkal kezelte, és nem ismerte el a modell működőképességét. Az angol *free* szó kettős jelentéstartalma miatt a szabad szoftvereket sokszor nemcsak „szabadnak”, hanem egyszersmind „ingyenesnek” is tekintették, és ezt sok esetben összekapcsolták a szellemi tulajdonjogok megsértésével. További téves felfogás volt, hogy az ingyenes vagy olcsóbb szoftver egyben rosszabb minőségű is.

Miután a *Netscape* internetes böngésző programja egyre nagyobb piaci részesedést veszített a *Microsoft* böngészőjével szemben, 1998 februárjában a *Netscape* szokatlan lépésre szánta el magát: nyílttá tette a böngésző forráskódját. A *Netscape*-ügy kapcsán, azzal szoros összefüggésben alakult ki a nyílt forráskód definíciója, melyet a „Nyílt Forráskód Kezdeményezés” (*Open Source Initiative*) alkotott meg. A nyílt forráskódú mozgalom a szabad szoftverek alapjain fejlődött ki, de hívei elsősorban a technikai előnyöket (a zárt szoftverekhez képest magasabb szintű megbízhatóságot és magasabb biztonsági szintet) hangsúlyozzák, míg a szabad szoftver fogalom szószólói inkább a szabadság felől, morális szempontból határozzák meg törekvéseiket.

Az új kezdeményezés üzletbarát szemléletű marketingmegközelítést nyújtott a közösségi fejlesztésű szoftverekhez, amit már az üzleti döntéshozók is elfogadtak, és hajlandók voltak fizetni is a régi-új típusú termékekért. A nyílt forráskód definíciójának marketingkampánya elérte célját, és sikeres üzleti vállalkozások alakultak a nyílt forráskódú szoftverek fejlesztésére.<sup>6</sup>

A *Sun Microsystems* 2006-ban a *GPL*-licenc alapján nyílttá tette a *Java*<sup>7</sup> forráskódját a fejlesztői közösségek számára. Ez a lépés a *Netscape* 1998. évi, történelmi mérföldkőnek számító lépéséhez hasonlítható, de azon túlmutató is lehet, hiszen ez az eddigi legnagyobb ilyen típusú „fegyvertény”. Szintén a *Sun* volt az a cég, amely az általa megvásárolt *StarOffice* irodai programcsomag forráskódját kettős licencvédelem mellett nyílttá tette: ez azt jelenti, hogy *StarOffice* néven kereskedelmi szoftver is készül, míg a közösség által fejlesztett *OpenOffice* teljesen nyílt. Mindkét programcsomag fejlesztésének eredményei beépülnek a másik csomagba is. Az irodai programcsomagok körében a *Microsoft Office* versenytársává vált *OpenOffice* fileformátuma, az *Open Document Format (ODF)* ISO-szabvánnyá vált.

A nyílt szabványnak köszönhetően sok közigazgatási rendszer, illetve intézmény előnyben részesítette a nyílt formátumot, ami arra kényszerítette a *Microsoft*-ot, hogy programcsomagjához készített kiegészítéssel alkalmassá tegye saját alkalmazását az általa használt zárt fájlformátum és a nyílt fájlformátum közötti konverzió elvégzésére.

– A jog arra, hogy tökéletesítsék a programot, és a tökéletesített változatot közzétegyék, hogy az egész közösség élvezhesse annak előnyeit. Ennek előfeltétele a forráskód elérhetősége.

Egy program szabad szoftver, ha a felhasználók ezen jogok mindegyikével rendelkeznek.

<sup>6</sup> A két megközelítés között egyik tábor képviselői sem látnak kibékíthetetlen ellentétet, azonban mindkét oldal a maga igazságát igyekszik kihangsúlyozni. Stallman ezt úgy fogalmazta meg, hogy a két irányvonal két politikai párthoz hasonlítható – ugyanazon a szabad szoftveres fejlesztői közösségen belül. A szabad szoftver és a nyílt forráskódú szoftver elnevezéseket sokszor – helytelenül – szinonimaként kezelik.

<sup>7</sup> A *Java*-technológia egységes szoftverfejlesztési platformot kínál a mobiltelefonoktól kezdve a szuperszámítógépekig. A *Java*-platformot több milliárd készülék használja.



## Nyílt fejlesztői modell

A hackerkultúra<sup>8</sup> és maga a *hacker* elnevezés is a számítástechnika fejlődésének korai szakaszából ered. A *hacker* fogalma a számítástechnika kereskedelmivé válása és az internet térhódítása miatt a köznyelvben hatalmas jelentésbeli változáson ment keresztül: a *hackereket* napjainkban – helytelenül – a számítógépes bűnözőkkel azonosítják.

A nyílt forráskódú programok fejlesztése az 1980-as évek végén és az 1990-es évek elején kezdetben néhány egymástól viszonylag elszigetelt csoportban zajlott. A *USENET* és az internet elterjedése segítette ezeknek a kezdeményezéseknek az összekapcsolódását és összefogását: egyre erősebb közösségek épültek ki, s végül sikerült ezeknek a csoportoknak a munkáját „egyesíteni”. Így jöhettek létre a zárt szoftverek alternatívái az 1990-es évek végére. A programozók munkájának társadalmi vonatkozásaira több tanulmány is rámutatott, és a csoportkohézió, illetve a vezetői szerepek kutatása is hangsúlyt kapott.

A nyílt forráskódú, nyílt fejlesztői modell lényege abban áll, hogy az ilyen programok mindenki számára elérhetők, aki internetkapcsolattal rendelkezik. Bárki letöltheti, tesztelheti és fejlesztheti a programokat, vagy erre irányuló javaslatokat küldhet be. A közösség tagjai segítik egymást, és megosztják a tapasztalataikat, ami tanulási folyamatot és tudástermelést eredményez az egész közösségen belül (Hemetsberger-Reinhardt 2004). A fejlesztés eredményét az egész közösség élvezi, nemcsak azok, akik közvetlenül részt vettek a munkában, s így közösségi tudásvagyon jön létre.

## Az OSS használatára irányuló kezdeményezések

Ma – bár nem szervezett kampányok keretében, de szerte a világon és az EU országain belül is – az OSS egyre nagyobb térhódítása figyelhető meg. Az ezekre való átállást irányító szakemberek, illetve az átállást támogató politikusok három érvet szoktak kiemelni, amelyek az OSS mellett szólnak: az *olcsóságot*, a *tulajdonosi jelleget* és a *biztonságot* (László–Rab 2005). Kutatásom során arra törekedtem, hogy a felsorolt érveken túlmenően is feltárjam azokat a szempontokat, amelyek befolyásolták a nyílt forráskódú szoftvereket előnyben részesítő különféle szervezetek döntéseit.

A cél egy olyan egységes modell megalkotása volt, amelyben a szoftverek által érintett területek azonosításán keresztül összehasonlíthatóvá válnak a zárt és a nyílt forráskódú szoftverek, figyelembe véve azokat az egyre nagyobb teret nyerő mértéktartó véleményeket is, amelyek azt hangsúlyozzák, hogy nem kell élesen választani az OSS és a zárt szoftverek között, hanem a kettő kombinációja adja az ideális megoldást. A jelen cikk terjedelmi keretei között vázlatos áttekintést adok az ezzel kapcsolatban általam feldolgozott esettanulmányokról.

Az EU az elmúlt években több száz OSS-projektet indított el, saját nyílt forráskódú szoftvereket fejleszt és, az OSS-mozgalmat a lisszaboni célok egyik nagy segítőjeként

<sup>8</sup> A *hacker* a szó eredeti jelentése szerint olyan szakember, aki a saját területéhez mélységben ért, és akit a megismerés vágya késztet az előrehaladásra és a fejlődésre. Az úttörő programozók önmagukra vonatkoztatták. A szó használata a Massachusetts Institute of Technology (MIT) számítástechnikai műhelyéhez vezethető vissza.

apoztrofálja. A legsikeresebb európai OSS-projekt végrehajtására Spanyolország Extremadura tartományában került sor, amely az EU egyik legszegényebb térsége volt, de az OSS segítségével kitört, és négy egymást követő évben elnyerte „az EU leginnovatívabb tartománya” címet a vele járó díjakkal együtt.

A Dél-afrikai Köztársaság kormánya volt az első országos kormányzat, amely hivatalosan is állást foglalt a nyílt forráskódú szoftverek használata mellett. Itt ezeket költséghatékonyság szempontjából az egyedi (*proprietary*) szoftverek preferált alternatívájának tekintik a kormányzati és közszolgálati munkában. A dél-afrikai kormányzat szerint a nyílt forráskódú szoftverek előnyei nem merülnek ki a több milliárd randot kitevő megtakarításban. Fontos további szempontok, hogy az OSS felhasználói nincsenek kiszolgáltatva a meghatározott szoftvergyártóknak, az OSS alkalmazása segít az informatikai képzésben, és végső soron hozzájárul a növekedéshez és a fejlődéshez.

Kína saját Linux disztribúciót fejlesztett *Red Flag Linux* néven, és ezt a csomagot támogatja az iskolákban, a közintézményekben és a felhasználók otthoni számítógépein egyaránt. Ennek fő oka – az illegális szoftverhasználat visszaszorításán kívül, amire Kína a Világkereskedelmi Szervezethez (*World Trade Organization, WTO*) való csatlakozása miatt kényszerült – elsődlegesen a *Microsoft*tól való függetlenség megteremtése volt (Einhorn 2003).

Peruban olyan törvényt fogadtak el, amely a közbeszerzések során biztosítja a nyílt forráskódú szoftverek beszerzésének esélyegyenlőségét. Brazília esetében a nyílt forráskód melletti döntés indokolása tisztán gazdasági jellegű volt, azonban a háttérben itt is a függőség megszüntetésére tett lépések érzékelhetők. Brazília lett az első olyan ország, ahol a közpénzekből finanszírozott szoftverfejlesztések eredményeit nyílt forráskódú licenc alapján kell kibocsátani.

A világ kormányzatainak állásfoglalása az OSS támogatását illetően – a fenti példák kivételével – korántsem egységes, illetve legtöbbször egyáltalán nincs tisztázva, és ez kiváltképpen igaz az EU-ra.

## Az L-PEST-modell

A feldolgozott esztanulmányok és az EU „Interoperábilis páneurópai elektronikus kormányzati szolgáltatások közigazgatási rendszerek, gazdasági szervezetek és állampolgárok részére” (*Interoperable Delivery of Pan-European eGovernment Services to Public Administrations, Businesses and Citizens, IDABC*) című dokumentuma alapján komplex modellé kívántam fejleszteni az eredményeket. A modell az egyes részelemek angol megnevezéseinek kezdőbetűiből alkotott *L-PEST* nevet kapta (László 2005).

*Legal environment* – Jogi környezet

Szerzői jog, licencek  
 Felelősség  
 Szoftverkalózkodás (*piracy*)  
 Szabadalmak

*Political environment* – Politikai környezet

Személyes adatok védelme (*privacy*)  
 Digitális állandóság  
 Digitális örökség  
 Nyílt kormányzat  
 Közbeszerzések

*Social environment* – Társadalmi környezet

Szabadság és esélyegyenlőség  
 Oktatás  
 Szoftverhasználati szokások  
 Digitális megosztottság

*Economical environment* – Gazdasági környezet

Költségsökkentés  
 Piaci egyensúly  
 Innováció  
 Munkahelyteremtés  
 Függőség

*Technological environment* – Technológiai környezet

Minőség  
 Funkcionalitás  
 Interoperabilitás  
 Átláthatóság  
 Szabványok támogatása  
 Biztonság

1. ábra. Az *L-PEST-modell*

## Jogi környezet

A jogi környezet körülöleli a modell többi összetevőit, mindazonáltal saját jellemzői vannak. Hatással van a politikai életre, a gazdaságra és a társadalom egészére, valamint a technológiai kérdéseket is szabályozza.

## Szerzői jog, licencek

A szerzői jog a leggyakrabban alkalmazott eszköz a szoftvertermékek védelmére. A szerzői jog automatikusan védelmezi a szellemi alkotásokat, beleértve a szoftvereket is.<sup>9</sup> Valójában a nyílt forráskódú szoftverek licencei esetében is a szerzői jogot alkalmazzák. A termékek továbbterjesztése csak úgy tehető meg, ha a licencben speciális engedélyeket biztosítanak erre, mivel alapesetben a szoftvernek sem a továbbterjesztése, sem a használata nincs engedélyezve. A licenc kikényszerítheti, hogy a továbbterjesztő eleget tegyen bizonyos feltételeknek (Working Group on Libre Software, 2000, 20–21). Sokféle nyílt forráskódú licenc létezik,<sup>10</sup> a legtöbb ilyen licenc az USA szigorú jogi kör-

<sup>9</sup> A *Creative Commons* licencrendszer a hagyományos „minden jog fenntartva” szerzői jog alapján jött létre, nagyobb szabadságot biztosítva a szerzőknek („néhány jog fenntartva”). <http://www.creativecommons.org>

<sup>10</sup> A nyílt forráskód licencpolitikája a nyílt forráskód és a szabad szoftveres mozgalom különbségeit mutatja. A szabad szoftveres licencek nem engedik a forráskód „bezárását”, míg a nyílt forráskódú licencek között találhatók olyanok, amelyek zárt forrású fejlesztéseket is engedélyeznek. Philosophy on: Why „Free Software” is better than „Open Source”. <http://www.gnu.org/philosophy/free-software-for-freedom.html>; Free Software licencek: <http://www.fsf.org/licensing/>; Open Source licencek: <http://www.opensource.org/licenses/>



nyezetében készült. Ezek közül a legismertebb a szabad szoftverek alapvető licence, az „Általános Nyilvános Licenc” (*General Public License, GPL*), amelyre gyakran alkalmazták a *copyleft* megnevezést is. A nyílt forráskódú szoftverek licencei a későbbi üzleti felhasználás tekintetében megengedőbb feltételeket szabnak. Az utóbbi idők kutatásai és bírósági döntései alapján<sup>11</sup> a GPL-licenc Európa legtöbb országában már alkalmazható. Az Európai Unió országaiban elfogadott szabad szoftver licencek közül a *GNU General Public License (GNU GPL)* használata esetében a felhasználó kötelezettséget vállal arra, hogy a továbbfejlesztett programot az eredeti feltételekkel terjeszti tovább.

A felelősség ebben az esetben az előállító felelősségét jelenti. A szoftverek esetében a hagyományos „fizikai” termékek előállítóival vagy az ilyen szolgáltatásokkal szemben valójában a „dobozos” (zárt forráskódú) szoftverek fejlesztői is kizárnak szinte minden felelősséget a termékek használata során okozott vagy a termékek használhatatlansága miatt bekövetkezett károkért. Sok ország jogrendszere azonban tiltja az ilyen jellegű felelősség kizárását vagy korlátozását.

A szoftverkalózkodás, az illegális szoftverhasználat kérdésköre minden országot érint, és a gazdaság sok területén fejt ki hatását. A szellemi tulajdonra vonatkozóan gyenge szabályozást érvényesítő országok nem vonzzák a külföldi tőkebefektetőket. Ez volt Kína egyik oka arra, hogy csökkentse az illegális szoftverek használatának arányát, amióta a WTO tagja lett. A fejlődő országok esetében sokkal nagyobb az illegális szoftverek aránya, mint a gazdaságilag fejlett országokban, az utóbbiak azonban a szoftverek illegális használata miatt nagyobb gazdasági veszteségeket szenvednek el (BSA, 2005).

## Szabadalmak

A szabadalmaztathatóság szempontjából az informatika különleges terület, mivel az új informatikai termékek nagyon gyakran kész elemekből, mintákból épülnek fel. Egy szoftver sosem egyetlen ötletet tartalmaz, hanem ötletek füzereit, és mások rész-megoldásainak felhasználása nélkül újat alkotni gyakorlatilag lehetetlen.<sup>12</sup>

Az 1973-ban Münchenben elfogadott Európai Szabadalmi Egyezmény (*European Patent Convention, EPC*) értelmében Európában számítógépes eljárásokat önmagukban nem lehet szabadalmi oltalom alá vonni. Az Európa Tanács 1991. május 4-ei 91/250/EGK-irányelve ugyanakkor kimondja, hogy az „eredeti számítógépi programok bármely formában történő kifejezése irodalmi műként szerzői jogi védelemben részesül” (10. preambulumi pont). Az EU-ban (is) jelenleg érvényben levő jogi környezet egyértelműen szabályozza ezt a kérdést, a szabadalmak igénylői ennek ellenére – különféle indoklásokkal – igyekeznek a szabályokat másképpen értelmezni, illetve kiderül: napjainkban az EU-ban 20-30 ezer szabadalom vár „aktiválásra”. (Az USA jogi környezetében elfogadott a szoftverszabadalom.) Minden elismert szabadalom 20 éves monopóliu-

<sup>11</sup> A német bírósági ítélet: [http://www.jbb.de/urteil\\_lg\\_muenchen\\_gpl.pdf](http://www.jbb.de/urteil_lg_muenchen_gpl.pdf)

<sup>12</sup> A szoftverszabadalmakról bővebben: Software Patents in the EU

(<http://www.oreillynet.com/pub/a/network/2005/03/08/softwarepatents.html>) és Software Patents vs. Parliamentary Democracy (<http://swpat.ffii.org/>).

mot biztosít a szabadalom első bejegyzője számára. Bár a szabadalmak benyújtásakor elvileg meg kell felelni bizonyos kritériumoknak, mostanra számos olyan szabadalom bejegyzése is megtörtént, amelyeknek a tartalma minden szakavatott felhasználó számára triviális, mert azt a gyakorlatban évek óta alkalmazzák.

## Politikai szempontok

A politikai szempontokat a kormányzatok funkcióival és kötelezettségeivel összefüggésben tettem vizsgálat tárgyává. Kötelezettségnek számító feladatuknak tekinthető például a társadalmi igazságosság biztosítása, míg egyik funkciójuk például az adóztatás (Lanvin 2003).

## A személyes adatok védelme

A magánélet és a személyes adatok védelme alkotmányos jog. A kormányzatra hárul azoknak a szabályoknak a megalkotása és betartatása, amelyek ezt a jogot az állampolgáraik számára biztosítják. Azok a szoftverek, amelyek személyes adatok továbbítására alkalmasak, vagy lehetővé teszik számítógépes rendszerek adatainak harmadik fél által végrehajtható módosítását, illetve az azokba való betekintést – a felhasználó kimondott hozzájárulása nélkül –, megsértik az állampolgári jogokat. Nem igazolható az az állítás, miszerint a piac az adott viszonyok között kellőképpen biztosítani képes az állampolgári jogok védelmét. A szoftverek felhasználásában a „hálózati hatások” alapelvei érvényesülnek: egy bizonyos pont elérése után megszűnik a választási szabadság, és az azonos programokat alkalmazó felhasználók az interoperabilitás kedvéért egy csoportba gyűlnek. Azáltal, hogy a nyílt forráskódú szoftverek – annak köszönhetően, hogy több ezer hozzáértő programozó vagy akár maga a végfelhasználó is átvizsgálhatja őket – nagyobb biztonságot nyújtanak az adatok titkos kezelésében.

A kormányzat rengeteg adat (például születési anyakönyvi adatok, adózással kapcsolatos adatok, szociális adatok stb.) megőrzéséért felelős. Létfontosságú, hogy az adatok hosszú időn át is biztonságban megőrződjenek, biztosítva legyen a „digitális állandóság”, anélkül azonban, hogy éppen emiatt régi szoftververziókat kellene használni. A vezető „dobozos” szoftverek saját régi változataikkal való kompatibilitása egyáltalán nem tekinthető garantáltnak, és ez a gyakorlatban nem is mindig valósul meg.

Massachusetts államnak a közigazgatásért és pénzügyekért felelős minisztere, Erik Kriss találó megjegyzése szerint „az egyik központi kérdés az, hogy miképp biztosítsuk a közadatok operációs rendszertől és alkalmazásoktól való függetlenségét hosszú időtávlatban. Az információs technológia területén a hosszú táv úgy 18 hónap, a kormányzatban ez körülbelül 300 évet jelent, tehát egy kissé különböző távlatban gondolkodunk.”<sup>13</sup>

A digitális örökség megőrzésének célja annak biztosítása, hogy a digitális formában tárolt tudás és adatvagyon hosszú időn át elérhető maradjon mindenki számára.

<sup>13</sup> Informal comments on open formats. [http://www.mass.gov/coaf/open\\_formats\\_comments.html](http://www.mass.gov/coaf/open_formats_comments.html)



A tárolás és a visszakéreshetőség szempontjából a digitális formára átalakított anyagokat azonos eljárás alá tartozónak tekinthetjük azokkal, amelyek már eleve elektronikusan keletkeztek. A közös probléma mindkét esetben olyan informatikai eszközök és szoftverek létrehozása, amelyek biztosítják a digitális állandóságot.<sup>14</sup>

A „nyílt kormányzat” fogalmát az általam kidolgozott modell tágan értelmezi, az átlátható kormányzat mellett ideértve a közadatokhoz való szabad hozzáférést is.

## Gazdasági szempontok

A gazdasági szempontok közül elsőként a szoftverhasználat költsége merül fel. A költségek azonban a szoftvereknek nemcsak a beszerzési árát, hanem az üzemeltetésük költségeit is magukban foglalják. A szoftverek esetében fontos mérőszám a tulajdonlás teljes költsége (*Total Cost of Ownership, TCO*). Az, hogy melyik szoftver üzemeltetési költségei a legalacsonyabbak, valójában a felhasználási környezettől és a szükségletektől függ. A *TCO* a ténylegesen felmerülő költségeket fejezi ki, de nem mindegy, hogy milyen költségcsoportok figyelembevételével. A *TCO* megállapításához először meg kell határozni a fő költségelemeket, majd azok tényleges költségeit. Ezekbe beletartoznak a „rejtett” költségek is, például az adminisztráció, a frissítés (*upgrade*), valamint a technikai támogatás és az üzemeltetés költségei is (Wheeler 2005). Egy másik megközelítés a befektetés megtérülésének (*return on investment, ROI*) vizsgálatával tárja fel a gazdasági hatásokat. Mindkét esetben megállapítható azonban, hogy az eredmények – az előzetesen figyelembe vett tényezőktől függően – erősen változók lehetnek.

A piaczgazdaság keretei között is előfordulhat, hogy a kormányzatoknak a közérdek védelmében a törvényi szabályozáson túlmenően is be kell avatkozniuk a piac működésébe. A szoftverbeszállítók monopolizálódó helyzete felborítja a piac egyensúlyát, pénzügyileg kiszolgáltatottá teszi a megrendelőt, és leszűkíti a választási lehetőségeket. A nyílt forráskódú szoftverek professzionális szereplőként való megjelenése visszahozza a piac egyensúlyát, és újra kibővíti a lehetőségeket. A kormányzatok szerepe itt a semlegesség és a különböző termékek esélyegyenlőségének megteremtése lehet, például a közbeszerzések során. A világ számos országában történtek kezdeményezések arra vonatkozóan, hogy a közpénzekből finanszírozott kutatási és fejlesztési programok eredményeit nyílt forráskódú dokumentációs licenc alapján kellene kibocsátani. Ez a megoldás vagy a kettős licencelés lehetővé teszi a kutatási eredmények megosztását és a későbbiekben azok üzleti célú felhasználását is, miközben nagymértékben serkenti az innovációt (Wong 2004).

<sup>14</sup> „A világ digitális örökségét a végleges eltűnés veszélye fenyegeti. Ehhez hozzájárul az azt tartalmazó hardware és software gyors elavulása, bizonytalanság a forrásokkal, felelősségekkel és a fenntartás és megőrzés módszereivel kapcsolatban, valamint a támogató jogi szabályozás hiánya. A hozzáállásbeli változások elmaradtak a technológiai változások mögött. A digitális fejlődés túl gyorsan és a kormányok és intézmények számára túl költséges módon ment végbe ahhoz, hogy a megfelelő időben lehessen megőrzési stratégiákat életbe léptetni. Az örökség gazdasági, társadalmi, szellemi és kulturális potenciálját – a jövő építő elemeit – érintő fenyegetéseket még mindig nem tudták teljesen megragadni.” Charta a digitális örökség védelméről. (Elfogadva az UNESCO Közgyűlésének 32. ülésén, 2003. október 17-én.) A Magyar UNESCO Bizottság nem hivatalos fordítása <http://www.unesco.hu/index.php?id=508&type=node>



A nyílt forráskód üzleti modellje hasonló a specializált szolgáltatóiparához, valamint a jogtudomány, az orvostudomány vagy a mérnöki tudományok területén kialakult modellekhez. A kormányzati rendszerek nyílt forráskódú rendszerek felé történő elmozdulásának eredményeként több jól fizetett információs technológiai munkakör keletkezik az adott országon belül. A szoftverbeszerzésre fordított összegek, amelyeket nem külföldi cégeknek fizetnek ki, hanem saját fejlesztésű szoftverekre költenek az országon belül, a jelentős költségmegtakarítás mellett nyilvánvalóan növelik az állam adóbevételeit is (Stanco 2003).

A függőség azt jelenti, hogy a felhasználó mennyire van kiszolgáltatva a szoftver szállítójának. Amíg a program megfelel a felhasználó igényeinek, nincs szükség magára a forráskódra. Amikor azonban valamely felhasználó változtatni akar a program tulajdonságain, illetve az általa nyújtott szolgáltatásokon, a forráskód megváltoztatására is szükség lehet, amihez nélkülözhetetlen maga a forráskód és a megváltoztatásához szükséges engedély. A zárt szoftvereknél a szoftverfejlesztő tulajdonában van a forráskód, és a program *reengineering* típusú visszafejtése általában tiltva van, így a felhasználó nem változtathat magán a forráskódon. Az ilyen jellegű termécsapdát a felhasználó nem tudja elkerülni, mivel ha egy újabb változat jelenik meg a programból, akkor általában kénytelen jelentős összegek ráfordításával frissíteni, vagyis áttérni az új változatra.

A függőség szorosan kapcsolódik a biztonság kérdéséhez is. Mint a digitális állandóság esetében láthattuk, megfontolást igényel az a kérdés is, hogy mennyire biztonságos zárt formátumú állományban tárolni az adatokat. Abban az esetben, ha a cég megszünteti a termék forgalmazását, a felhasználó még kiszolgáltatottabb helyzetbe kerülhet. A szabad szoftverek esetében a forráskód elérhető, így könnyen készíthető kompatibilis illesztés a már meglevő szoftverekhez akár évtizedekkel később is.

## Társadalmi hatások

Az IKT-javak elterjedése és használata hatalmas potenciált rejt magában a mindennapi élet jobbá tételére az egészségügy területétől kezdve az állampolgároknak a döntéshozatali mechanizmusokban való aktív részvételéig, és a megfelelő környezet biztosítása esetén az esélyegyenlőség megteremtésében is fontos szerepet játszhat. Az oktatás nagy hatást gyakorol egy ország gazdaságára, ezért a megfelelő környezet biztosítása szempontjából kulcsfontosságú a kormányzatok szerepe. A digitális írástudás széles körű megteremtésével a digitális szakadék csökkenthető, s ezáltal a munkaerőpiacon való elhelyezkedés is könnyebbé válhat. Az oktatás területén többnyire komoly megfontolás tárgyát képezi a hallgatóknak oktatott szoftverek beszerzése, ami jelentős költségekkel jár. Ez két problémát is felvet. Az oktatási intézménynek egyrészt rendelkeznie kell az oktatott szoftverrel, megfelelő mennyiségű licenccel ellátott gépre telepítve, másrészt ezeket a programokat a hallgatók önálló munkájához is biztosítania kell vagy az intézmény által üzemeltetett gépteremben, vagy lehetővé téve a hallgatók számára a szoftver használatát otthoni gépükön is. Az általában amúgy is alulfinanszírozással küzdő oktatási intézmények számára a hagyományos szoftvertermékek esetében mindkét megoldás számottevő költségtöbbletet jelent.

A szoftverhasználattal kapcsolatban megállapítható, hogy a modellfelhasználóként viselkedő kormányzatok az oktatáson keresztül is befolyásolhatják a szoftverhasználati szokásokat. Fontos és az illegális szoftverhasználattal szorosan összefüggő kérdés továbbá az is, hogy a kormányzatok milyen formában biztosítják a kommunikációt és a közérdekű adatok elérhetőségét.<sup>15</sup> Sok felhasználó nem tudja, vagy nem akarja megvásárolni a hagyományos kereskedelmi szoftvercsomagokat. Számukra a nyílt forráskódú szoftverek vonzó, technikailag és jogilag egyaránt elfogadható alternatívát biztosíthatnak, és ezek alkalmazása folytán a digitális megosztottság is inkább csökken, mint növekszik.

## Technikai szempontok

A modell többi részénél tárgyalt elemek mindegyikében szerepet játszanak technikai jellegű tényezők is. Az említett kérdések egy része visszavezethető technikai problémákra, amelyeknek a megoldása viszont nem csupán technikai szemléletmódot kíván. Bár a nyílt forráskódú szoftverek fejlesztésének folyamata kaotikusnak tűnik, a gyakorlat azt mutatja, hogy maga a végtermék nagyon sok esetben azonos értékű vagy magasabb színvonalú, mint „dobozos” versenytársai. A szoftverek megbízhatósága pedig napjainkban és a jövőben is egyre fontosabb tényező, hiszen egyre nagyobb mértékben függünk tőlük, befolyásolják a mindennapi életünket (is). A megbízhatóság, a teljesítmény és a rendszer skálázhatósága olyan mérhető technikai paraméterek, amelyeknek a segítségével elvégezhető két különböző rendszer technikai összehasonlítása (Wheeler 2005), míg a különböző rendszerek funkcionalitása a szoftvertermékek esetében is a felhasználók igényeinek és elvárásainak való megfelelés alapján ítéltető meg.

Az interoperabilitás – a kölcsönös együttműködő-képesség – szintje azt fejezi ki, hogy a különböző gyártóktól származó szoftverek és hardverek mennyiben képesek egymással együttműködni, például azonos protokollok és eljárások használata mellett adatok cseréjének a megvalósításával. Monopólium kialakulásához vezethet, ha valamelyik gyártó termékei nem interoperábilisak. A nyílt forráskódú szoftverek teljes mértékben átvilágíthatók, s ez megkönnyíti működésük áttekintését és megértését. Az Európai Bizottság állásfoglalása és az „Európa összekapcsolása: az együttműködő-képesség fontossága az e-kormányzati szolgáltatásokért” címet viselő munkaanyag hangsúlyozza, hogy a tervezett európai együttműködő-képesség keretprogramjának „nyílt szabványokon kellene alapulnia, és ösztönöznie kellene a nyílt forráskódú szoftverek használatát”.

A nyílt forráskódú szoftverekhez nem jár támogatás, ez azonban azt is jelenti, hogy felhasználójuk nincs egy támogatóhálózathoz kötve, nem kell elfogadnia annak az árait és feltételeit, hanem sok olyan cég vagy szervezet közül választhat, amelyek támogatják az OSS felhasználását. Bárki nyújthat támogatást, és a problémákat bárki megoldhatja.

<sup>15</sup> A magyarországi helyzetet jól illusztrálja az APEH ABEV szoftvere, amely csak Windows alapú verzióban érhető el. Emiatt a nyílt forráskódú szoftvereket használó vállalkozások az Alkotmánybírósághoz fordultak. <http://abiweb.obh.hu/abi/index.php?menu=81&did=15005>



Az informatikai rendszerek az országok kritikus infrastruktúrájává váltak, az energiaellátás, a telekommunikáció, gyakorlatilag valamennyi gazdasági alrendszer, a tőzsde és a közlekedés is ezekről a rendszerektől függ. A modern technológia által vezérelt infrastruktúrát ma már nem a hagyományos értelemben vett fizikai megsemmisítés veszélye fenyegeti elsődlegesen, hiszen ez az infrastruktúra távolról – elektronikus úton, szinte azonosíthatatlan módszerekkel – kisebb csoportok vagy akár egyének által is támadható, megbénítható, sőt megsemmisíthető. Éppen ezért kiemelt fontosságú, hogy az ilyen infrastruktúrát mennyire biztonságos megoldásokat alkalmazó szoftverekkel üzemeltetik (Steele-Vivas 1996).

Azok az országok, ahol nem az angol a hivatalos nyelv, az IKT-javak elterjedése terén hátrányos helyzetben vannak, ha nem állnak rendelkezésükre a szoftverek helyi, lokalizált változatai. A lokalizáció fontos része a fordítás, azonban ezen túlmenően szükség van a szoftvereknek a helyi kulturális és technikai igényekhez illeszkedő „testre szabására” is.

## Összefoglalás

A bemutatott általános modell azokat a területeket integrálja, amelyekre a közszféra – nagyfogyasztóként közvetlenül, szabályozóként pedig közvetve, a piacon keresztül – befolyást gyakorol. A kutatás során ezeket a területeket a nyílt forráskódú szoftverek felhasználásának szempontjából tettem vizsgálat tárgyává, az általános modell segítségével összehasonlíthatóvá téve a hagyományos és a nyílt forráskódú szoftvereket.

A végleges modell kidolgozása és az összetevők közötti kapcsolatok feltárása, valamint a modell verifikálása további munkát igényel. A téma elméleti hasznossága a nyílt forráskódon túlmutatóan a közigazgatásban használt szoftverrendszerek gazdasági és interoperabilitási kérdéseire vonatkozó elméleti munkák feltárásában és összegyűjtésében mutatkozhat meg. A kutatás eredményei azt mutatják, hogy a közszféra – saját fogyasztása, valamint az oktatás és az elektronikus ügyintézés révén – igen jelentős hatást fejt ki a társadalom szoftverhasználati gyakorlatára.

## Irodalom

- Bessen, J. (2003). What Good Is Free Software? In Hahn, W. R. (ed.): *Government Policy Toward Open Source Software*. Washington, D. C., AEI-Brookings Joint Center for Regulatory Studies, 12–33.
- BSA (2005). *Piracy study*. Letöltve: 2006. január 8.,  
<http://www.bsa.org/globalstudy/upload/2005-Global-Study-English.pdf>
- Einhorn, B. (2003). *Why Gates Opened Windows in China Business Week Online, 2003*. March 3. Letöltve: 2005. december 12.,  
[http://www.businessweek.com/technology/content/mar2003/tc2003033\\_6406\\_tc058.htm?tc](http://www.businessweek.com/technology/content/mar2003/tc2003033_6406_tc058.htm?tc)
- Hemetsberger, A. – Reinhardt, C. (2004). *Sharing and Creating Knowledge in Open-Source Communities – The case of KDE*. Letöltve: 2004. december 12.,  
<http://opensource.mit.edu/papers/hemreinh.pdf>



- László G. – Rab Á. (2005). *Bevezetés a nyílt forráskódú szoftverek világába*. Kutatási jelentés, 2005. Szerk. Horváth Andrea és Molnár Szilárd. Budapest, INFONIA Alapítvány, 128–154.
- Laszlo, G. (2005). *Open source for governments: Are the governments ready for transparency and interoperability?* Proceedings of the International Conference on e-Government (ICEG 2005). Ottawa, Canada October 27–28, 2005. Ed. Dan Remenyi. Reading, UK, Academic Conferences Ltd., 163–174.
- Lanvin, B. (2003). Leaders and Facilitators: The New Roles of Governments in Digital Economies. In Dutta, S. – Lanvin, B. – Paua, F. (ed.): *The Global Information Technology Report 2002–2003 – Readiness for the Networked World*. Oxford, Oxford University Press, 74–83.
- Stanco, T. (2003). *US: Testimony*. Letöltve: 2005. október 1.,  
<http://www.egovos.org/Resources/Testimony>
- Steele-Vivas, R. D. (1996). Creating a Smart Nation: Strategy, Policy, Intelligence, and Information. *Government Information Quarterly*, 13, 159–173.
- Wheeler, D. A. (2005). *Why Open Source Software / Free Software (OSS/FS, FLOSS, or FOSS)? Look at the Numbers!* (November 14, 2005) Letöltve: 2005. december 12.,  
[http://www.dwheeler.com/oss\\_fs\\_why.html](http://www.dwheeler.com/oss_fs_why.html)
- Wong, K. (2004). *Free/Open Source Software*. Government Policy. International Open Source Network, Elsevier. Letöltve: 2005. október 12.,  
<http://www.iosn.net/government/foss-government-primer/foss-govt-policy.pdf>
- Working group on Libre Software (2000). *Free Software/Open Source: Information Society Opportunities*.