

Kritikus infrastruktúrák és szolgáltatások modellezése



Gönczy László



**NEMZETI
KÖZSZOLGÁLATI EGYETEM
BUDAPEST**



MAGYARORSZÁG
KORMÁNYA

SZÉCHENYI 2020

2020

Európai Unió
Európai Szociális
Alap



BEFEKTETÉS A JÖVŐBE

OKOSVÁROS-TECHNOLÓGIÁK
A technológia fejlődésének irányai és hatása
XVII. kötet

Sorozatszerkesztő:

Sallai Gyula

Gönczy László

**KRITIKUS
INFRASTRUKTÚRÁK
ÉS SZOLGÁLTATÁSOK
MODELLEZÉSE**



Nemzeti Közszerológati Egyetem
Közigerológati Továbbkérzési Intézet
Budapest, 2020

A kötet a Nemzeti Közszolgálati Egyetem **KÖFOP-2.1.2-VEKOP-15-2016-00001**
„A jó kormányzást megalapozó közszolgálat-fejlesztés” projektje keretében,
a Budapesti Műszaki és Gazdaságtudományi Egyetem Villamosmérnöki és
Informatikai Karán létesült **„Okos város – okos közigazgatás”** kutatóműhelyben
(2017/162/BME-VIK) készült.

Szakmai lektor:

Jakab László, egyetemi tanár, BME-VIK

A kézirat lezárásának dátuma:

2017. október 31.

© Nemzeti Közszolgálati Egyetem
Közigazgatási Továbbképzési Intézet, 2020

© Dr. Gönczy László, 2020

A mű szerzői jogilag védett. Minden jog, így különösen a sokszorosítás, terjesztés
és fordítás joga fenntartva. A mű a kiadó írásbeli hozzájárulása nélkül részeiben sem
reprodukálható, elektronikus rendszerek felhasználásával nem dolgozható fel,
azokban nem tárolható, azokkal nem sokszorosítható és nem terjeszthető.

TARTALOM

1. BEVEZETÉS	7
1.1. Kritikus infrastruktúra fogalmának meghatározása	7
1.2. Függőségek kritikus infrastruktúrák közt	8
2. A MODELLEZÉS ALAPFOGALMAI	10
2.1. Gráfmodellek alkalmazása okos városban	12
2.2. Függőségek modellezése	13
3. KRITIKUS INFRASTRUKTÚRÁK MINŐSÉGI TULAJDONSÁGAINAK MODELLEZÉSE	16
4. ONTOLÓGIAMODELLEK HASZNÁLATA OKOS VÁROS LEÍRÁSÁHOZ	20
5. SZOLGÁLTATÁSOK MINŐSÉGÉNEK MODELLEZÉSE	22
6. FOLYAMATMODELLEZÉS	25
6.1. IDEF-0	26
6.2. Business Process Modelling Notation (BPMN)	27
6.3. A Business Process Execution Language (BPEL) nyelv	28
6.4. Üzleti szabályok modellezése és a DMN szabvány	29
6.5. A TOGAF módszertan és a kapcsolódó ArchiMate modellezési nyelv	30
7. ADATVEZÉRELT MODELLEZÉS	32
7.1. Tipikus adatforrások és kapcsolódási pontok okos városban	33
7.1.1. Adatok és API kezelése Európában – CitySDK	34
7.1.2. Adatvezérelt modellek a jó kormányzásban – Michigan	35
7.2. Esettanulmány: városüzemeltetési bejelentések kiértékelése	36
8. MODELLEZÉSI ESETTANULMÁNYOK	41
8.1. Kritikus infrastruktúrák optimalizált rekonfigurációja	41
8.2. Pénzügyi infrastruktúrák magas szintű modellezése	43
8.3. Példa magyar infrastruktúra-modellre: Egységes Forgalmi Modell	45

9. KRITIKUS INFRASTRUKTÚRA MODELLEK ALKALMAZÁSI LEHETŐSÉGEI	46
9.1. Kockázatmodellezés és katasztrófavédelem	46
9.2. Infrastruktúra-fejlesztések támogatása szimulációs hatásvizsgálatokkal	46
9.3. Adat- és szolgáltatáskatalógus létrehozása	46
9.4. Integráció tervezése	47
IRODALOMJEGYZÉK	48

1. BEVEZETÉS

Jelen tanulmány célja olyan módszereknek, leíró nyelveknek és alkalmazási területeknek az áttekintése, amelyek kritikus infrastruktúrák, illetve az ezekre épülő szolgáltatások modellezésére alkalmasak. Modellezés alatt itt olyan megközelítéseket értünk, melyek az infrastruktúra vagy szolgáltatás bizonyos jellemzőin keresztül összefüggések feltárását célozzák meg. Nem cél tehát egy-egy ilyen modellnél a teljes pontosság (például közlekedési csomópont esetében a táblák elhelyezkedésének leírása), sokkal inkább olyan modellekkel foglalkozunk, amelyek az infrastruktúrák alapvető felépítését, használati módjait, minőségének főbb meghatározó faktorait, illetve esetenként szűk keresztmetszeteit írják le.

Az egyes infrastruktúra-típusok leírására és tervezésére számtalan szakterület-specifikus modellezési nyelv létezik, így például egy elektromos hálózat műszaki tervezésekor nyilván annak szakterületi modellje használatos. A főbb komponensek, a rajtuk végzett műveletek, illetve a rendszer határai, valamint az ezeken értelmezett lehetséges lépések azonban mind olyan elemek, amelyeket érdemes lehet kiemelni annak érdekében, hogy egységes legyenek leírva akár a különböző alrendszerek, akár a különböző infrastruktúrák. Ennek a megközelítésnek az előnye, hogy lehetővé teszi több kritikus infrastruktúra átfogó védelmét (a szakirodalomban Critical Infrastructure Protection, CIP).

1.1. KRITIKUS INFRASTRUKTÚRA FOGALMÁNAK MEGHATÁROZÁSA

A kritikus infrastruktúrák definíciója országonként eltérő, a témával leginkább komolyan foglalkozó és legrégebben publikus anyagokat megjelentető Department of Homeland Security (USA) definíciója szerint kritikus infrastruktúra minden olyan eszköz, rendszer vagy hálózat (legyen az fizikai vagy virtuális), amelynek tönkretétele vagy működésének jelentős akadályozása gyengítené az ország biztonságát, a nemzetgazdaság teljesítményét, a közegészségügyet vagy -biztonságot, vagy ezek bármilyen kombinációját.

Magának az infrastruktúrának a definíciója 1997-ben fogalmazódott meg az alábbiak szerint:

Szektorspecifikusan vizsgálva 16 olyan területet azonosítottak, mely kritikusnak minősül:

- Vegyi üzemek
- Kereskedelmi központok (beleértve a bevásárlóközpontokat, sőt, a kaszinókat is)
- Telekommunikáció
- Kritikus gyártás
- Gátak
- Védelmi ipar
- Sürgősségi ellátás (mentők, rendőrség, katasztrófavédelem)
- Energiaszektor
- Pénzügyi szolgáltatások

- Élelmiszeripar és agrárium
- Kormányzati szolgáltatások
- Egészségügyi ellátás
- Információtechnológia
- Nukleáris technológia
- Közlekedés
- Vízellátás és szennyvízkezelés

Jelen tanulmánynak nem célja minden szektor részletes áttekintése, inkább az okos városban történő felhasználás aspektusainak bemutatása, a szektorok egymásra hatásának vizsgálata, illetve az egységes modellezés lehetőségeinek bemutatása.

Bár a fenti szektorokra vonatkozó specifikus modellezés megközelítések léteznek (például gyártási vagy logisztikai folyamatok esetében), fontos kiemelni, hogy a manapság elérhető modellezőeszközök hatékony támogatást tudnak adni minden szektor folyamatainak leírására.

A folyamatok leírása mellett kiemelten fontos a rendszerekben keletkezett adatok kezelése, ideértve az adatok életciklusát és a lehetséges adatváltozást okozó beavatkozásokat. Emellett a rendszerek magas szintű felépítésének és lehetséges kapcsolatainak modellje is megalkotható topológiamodellek használatával.

1.2. FÜGGŐSÉGEK KRITIKUS INFRASTRUKTÚRÁK KÖZT

Az infrastruktúrák nem önmagukban, hanem egymással szorosan összekapcsolódva, sokféle függőség mentén működnek.



1. ábra: Kritikus infrastruktúrák függőségeinek főbb fajtái

Forrás: a szerző saját szerkesztése, [1] alapján

Az 1. ábra mutatja be a kritikus infrastruktúrák közti függőségek tipikus fajtáit, a nemzetközi szakirodalom alapján [1] [2]. A magyar kritikus infrastruktúra egy elemzését mutatja be [3].

Fontos kiemelni, hogy ezen függőségek tervezési időben sokszor nem ismertek, illetve az infrastruktúra-elemek kapcsolatainak dinamikus változása miatt változhatnak is, így a tervezési időben érvényesnek tekintett függőségeket a rendelkezésre álló adatok alapján ellenőrizni kell azok **teljessége** és **helyessége** alapján. Előbbi tulajdonság arra utal, hogy minden fontos függőséget ismerünk a modellben, míg utóbbi a függőségek érvényességére.

A tanulmány az alábbi felépítést követi:

- Bemutatjuk a modellezés alapfogalmait és a gráfalapú modellezési megközelítést.
- Bemutatjuk, milyen főbb aspektusok szerint írhatók le a kritikus szolgáltatások *tervezési* és *használati* elvárásai.
- Bemutatjuk az ontológia alapú modellezési megközelítéseket.
- Ismertetjük a szolgáltatások leírásának főbb szabványait.
- Összekapcsolt infrastruktúrák esetén különösen fontos az ezeken átívelő folyamatok megteremtésének lehetősége, így bemutatjuk a folyamatmodellezési megközelítéseket is.
- Az infrastruktúramodellek egyik fő kérdése a mennyiségi analízis, illetve a modellépítés helyességének és a modell aktualitásának ellenőrzése. Ehhez szükséges az elérhető adatok feldolgozása. Bemutatjuk, milyen publikusan elérhető adatforrások léteznek az okos városban, és mutatunk példát ilyen adatforrásból származó incidensadatok elemzésére.
- Végül esettanulmányokkal és összefoglalóval zárul a tanulmány.

A szerző¹

¹ *Dr. Gönczy László*, okl. mérnök-informatikus és mérnök-közgazdász, a Budapesti Műszaki és Gazdaságtudományi Egyetem Villamosmérnöki és Informatikai Kara Méréstechnikai és Információs Rendszerek Tanszékének adjunktusa. PhD-fokozatát a BME-en 2019-ben szerezte. Fő kutatási területe üzleti és kritikus IT-rendszerek tervezése és elemzése, különös tekintettel az adatvezérelt módszerek alkalmazására a teljesítmény- és teljesítőképesség meghatározásában.

2. A MODELLEZÉS ALAPFOGALMAI

Jelen fejezetben a modellezés alapfogalmait tekintjük át. Megvizsgáljuk, mit értünk modell alatt (elsősorban az általános rendszermodellezés fogalmai szerint), milyen főbb tulajdonságokat modellezhetünk, illetve milyen céllal alkothatunk modelleket.

A mérnöki modellezés területén számtalan megközelítés ismert. Manapság már kevés olyan modell van, amelynek nem létezik szoftveres támogatása, ez azonban azt igényli, hogy a modell elemkészlete és az elemek közti kapcsolat előzetesen definiálva legyen, hiszen ez szükséges a modellek tárolásához, megjelenítéséhez és a rajtuk értelmezhető műveletek (például elemek módosítása, lehetséges viselkedési utak meghatározása, származtatott értékek meghatározása stb.) végrehajtásához. Bár fókuszunkban most elsődlegesen nem a szoftvermodellek állnak, a szoftverek által használt modellezésmegközelítés megértése fontos annak érdekében, hogy a modellek építésének és felhasználásának tipikus lépéseit megértsük.

Modell: a modell alapvetően a valóság valamilyen egyszerűsített képe. Bár e szempont triviálisnak tűnik, a publikusan elérhető okosváros-koncepciók egyik legnagyobb kihívása épp a modellezés céljának és ezáltal az alkalmazott, elfogadható közelítéseknek a definiálása. Ilyen lehet például az, hogy egy közlekedési hálózat esetén szükséges-e a modellbe felvenni az egyes útvonalak burkolatának tulajdonságait vagy az adott területen található tűzcsapokat.

A modell alapvetően elemekből és a köztük felépített kapcsolatokból áll. Az elemek (és a modellezési megközelítéstől függően) a kapcsolatok rendelkezhetnek típusokkal és tulajdonságokkal. Egy város közlekedési modelljét példának véve, egy járatnak jellemzője a típusa (metró/busz/villamos stb.), a járatnak van útvonala, mely megállókból áll. A járat tulajdonsága lehet az, hogy alacsonypadlós-e, az útvonal tulajdonsága lehet a hossza, stb.

A **metamodell** egy olyan modell, amely leírja egy adott modellezési nyelv lehetséges fogalmait és a köztük megengedett kapcsolatokat. A kapcsolatok megadásánál fontos azok számosságának megadása is (például egy megálló tartozhat több járathoz, de egy konkrét járatnak mindig két végállomása van, stb.).

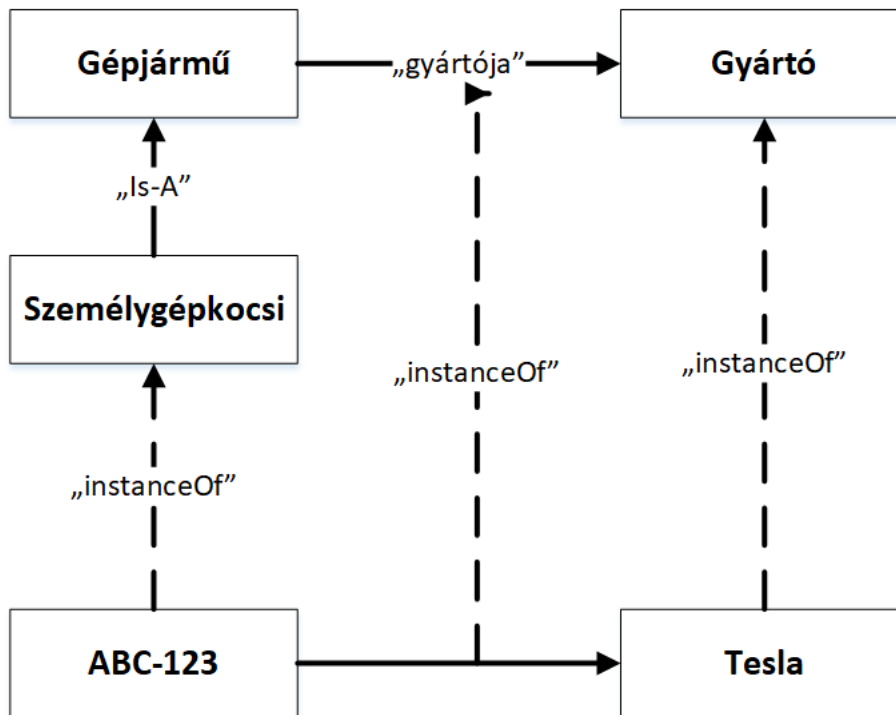
A számosságon kívül további úgynevezett **megkötések** is megadhatók, amelyek több modellelem vagy akár az egész modell együttes vizsgálatát igénylik. Ilyen lehet például annak tiltása/megengedése, hogy egy járat két végállomása megegyezzen, vagy annak előírása, hogy ne legyen olyan járat, amelynek útvonala nem kapcsolódik egyetlen más járatához sem.

Hasonló megkötéseket használhatunk arra is, hogy olyan követelményeket írjunk le, amelyeknek a modell által leírt rendszer működésének minden pillanatában teljesülnie kell. Ilyen lehet például, hogy egy kitüntetett csomópontnak minden pillanatban rendelkeznie kell legalább kettő, azt kiszolgáló járatral. Ezen kifejezések egy speciális, matematikai logikával támogatott leírási formája lehetővé teszi azt is, hogy a felírt modell alapján a rendszer működésének helyességéről és biztonságosságáról bizonyítékot szerezzünk, úgynevezett modellellenőrzés végrehajtásával.

A metamodellel elemei, és a metamodellel alapján felírt egyes elemei közt a „példánya” (instanceOf) logikai kapcsolat értelmezhető, amely megadja, hogy egy adott modellbeli elem milyen metamodellelbeli elem példányának tekinthető (azaz például egy adott elem közlekedési csomópont típusú).

Ez megkülönböztetendő attól az esettől, amikor magát a modellezési nyelvet (metamodellel) finomítjuk, ez esetben az „is-A” jellegű relációt használjuk, amely a metamodellel elemei közti kapcsolatot fejezi ki. Ezzel pontosítjuk a felhasználható elem típusokat (tehát például egy közlekedési hálózat eleme lehet csomópont vagy útvonal, és megfordítva, minden útvonal vagy csomópont egyben a közlekedési hálózat eleme is).

A programozási nyelvekben jártas olvasónak feltűnhet, hogy nem tárgyaljuk azt a kérdést, miszerint egy modellelem lehet-e több metamodellel elem példánya (ezt hívják programozási nyelvek, illetve paradigmák esetében többszörös öröklésnek). Jelen tanulmány szempontjából ennek nincs kiemelt fontossága, ugyanakkor a legtöbb helyességvizsgálat kifejezetten keresi azokat az elemeket, amelyek adott esetben két különböző szerepet is betöltenek.



2. ábra: Modell–metamodellel kapcsolat, példa

Forrás: a szerző saját szerkesztése

A metamodellel–modell közti kapcsolatra példát mutat a 2. ábra. Ezen látható, hogy adott esetben több szinten keresztül érvényes lehet egy specifikációs információ (a „gyártója” reláció értelmezett a gépjárművekre és a személygépkocsikra is), illetve a modellelemek közti relációk fajtáit is megszabja az ezeknek megfelelő metamodellel elemek közt felvehető relációk halmaza.

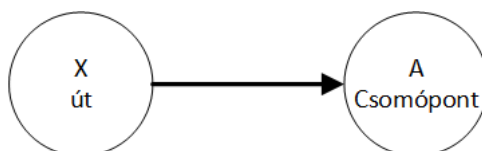
A modellek leírhatják egy rendszer **statikus** vagy **dinamikus** aspektusait. A statikus aspektusok alatt értjük alapvetően a rendszer felépítésére, a rendszer alkotóelemeire és az azok közti kapcsolatokra koncentráló modelleket. Ilyen modellekkel írhatjuk le egy rendszer topológiáját, összetételét, a rendszerben alkalmazott adattípusokat, a rendszer konfigurációját, a komponenseket és a köztük értelmezett kapcsolatot stb. Nagy infrastruktúráknál legtöbbször olyan modellek készülnek, melyek megadják az infrastruktúra felépítését.

Dinamikus modellekkel írjuk le egy rendszer viselkedését. A viselkedés maga lehet állapotalapú (amikor a modellben a rendszer egymástól elkülöníthető állapotaira és az azok közti átmenetekre koncentrálunk), vagy tevékenység/folyamatalapú, amikor az egymás után végrehajtott lépések sorrendiségének és függőségeinek meghatározása a modellezés fő feladata.

2.1. GRÁFMODELLEK ALKALMAZÁSA OKOS VÁROSBAN

Fontos kiemelni, hogy a legtöbb szakirodalom az infrastruktúra-modellezést visszavezeti gráfok leírására. Egy **gráf** alapvetően **csomópontok** és **élek** halmaza, ahol az élek csomópontokat kötnek össze. A csomópontok és az élek is lehetnek típusosak, az éleken értelmezhető súly, amely tipikusan az adott éllel kifejezett kapcsolat valamilyen tulajdonságától függ (például egy útvonal hossza). A gráf lehet irányított vagy irányítatlan, előbbi esetben egy élnek egyértelmű kezdő- és végpontja van, értelemszerűen a kezdőpontból a végpont felé létezik kapcsolat, amennyiben két pont közt oda-vissza kapcsolat létezik, akkor mindkét él fel kell venni a gráfba.

Az alábbi példa azt szemlélteti, hogy az X útvonal az A csomópontba vezet.



3. ábra: Irányított gráf

Forrás: a szerző saját szerkesztése

Egy gráfalapú modellen számos művelet elvégezhető, íme egy nem kimerítő lista a tipikus gráfalapú kérdésekről:

- Bejárás/transzitiv lezárás keresése. Egy adott kezdőpontból milyen más csomópontok érhetők el a gráfban? Ennek specifikus esete lehet, amikor megadjuk az utak maximális hosszát vagy költségét. A fenti példánál maradva: mely közlekedési csomópontok érhetők el egy adott kiindulási pontból, maximalizált utazási idő/megállók száma/átszállások száma mellett?
- Feszítőfa keresése. A feszítőfa keresése során egy adott gráf éleinek olyan részalmazát keressük, amelyek nem alkotnak kört (vagyis a lehető legkevesebb éllel lehet elérni a gráf elemeit). Ezen probléma egyik gyakori alosa, amikor minimális súlyú feszítőfát keresünk, illetve egy adott forráscsomóponthoz keressük a minimális súlyú feszítőfát. Infrastruktúra-modellezés esetén ez megfelel annak a kérdésnek, hogy hogyan és milyen költséggel tartható fenn egy olyan hálózat, amelynek minden pontjáról el lehet jutni egy adott kitüntetett csomópontba. Ilyen kérdéseket szoktak vizsgálni például szenzoradatok továbbítása esetén is.

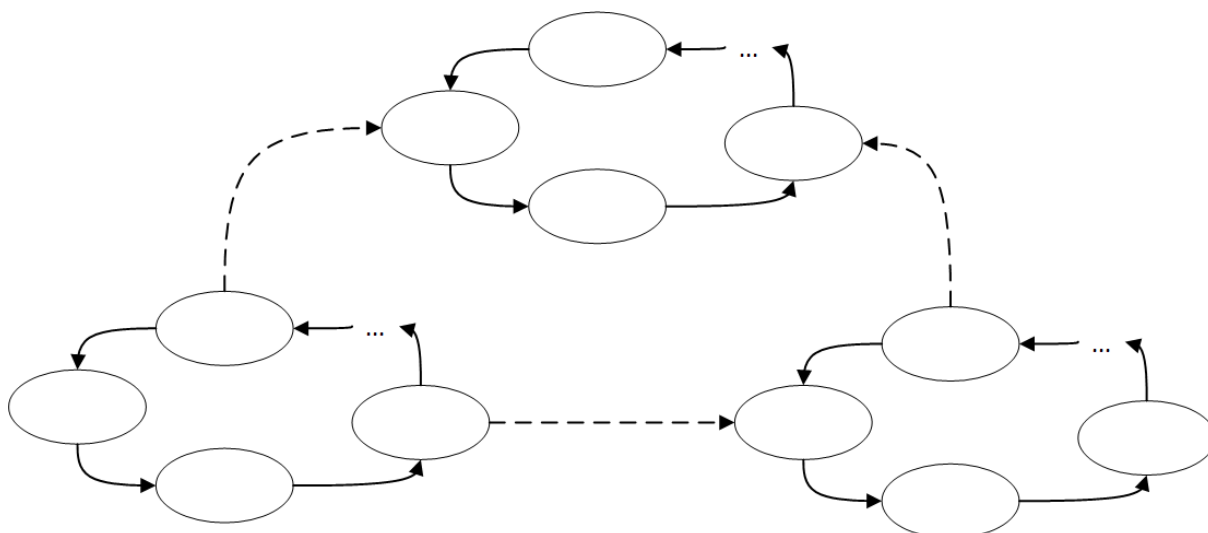
- Vágáskeresés. Adott kezdőpont és célpont esetén vágásnak nevezzük az éleknek azt a halmazát, amit elhagyva a kezdőpontból nem lehet eljutni a célpontba. Például katasztrófavédelmi szempontokból érdekes, hogy egy város két pontja közti összeköttetés megszűnéséhez hány útvonalnak kell elérhetetlenné válnia.
- Összefüggőség vizsgálata. Egy nagy gráf esetén érdekes a kérdés, hogy eljuthatunk-e a gráf minden csomópontjából minden másikba. Ennek további vizsgálati esetei, ha azt is vizsgáljuk, hogy több lehetséges út esetén az érintett csomópontok/útvonalszakaszok legalább hány elemben különböznek egymástól.
- Centralitás és kiemelt csomópontok vizsgálata. A gráfok csomópontjain definiálható az úgynevezett fokszám, amely megfelel az adott csomópontot érintő élek számának. Irányított gráfok esetén megkülönböztetjük a bejövő és kimenő élek fokszámát, valamint típusos gráfok esetén léteznek eszközök, amelyek adott típusú élek alapján számolnak fokszámot. A fokszámok önmagában megadják, hogy melyek a gráf legsűrűbben érintett elemei, ezzel utalnak az elemek vélhető fontosságára. Amennyiben az egyes csomópontok fokszáma mellett azok elhelyezkedését, például a különböző utak hosszát és jellegét is figyelembe vesszük, meghatározhatók a gráf „központi” elemei. Az elmúlt években ezek a fogalmak főleg a közösségi hálózatokkal kapcsolatos kutatások miatt kerültek előtérbe, de hasonlóan használhatók infrastruktúraelemek vagy összekapcsolt szolgáltatások és adatforrások, szenzorok fontosságának, központiságának vizsgálatára. Amennyiben a város szerkezete mellett annak használatát is modellezni akarjuk, például a mobiltelefon-hálózat és a hálózatban létrejött adatkapcsolatok használati mintái alapján, akkor a fokszám és egyéb fontossági metrikák felhasználhatók az infrastruktúra igény és használati oldala közti különbségek feltárására.

A gráfalapú modell-leírás előnye, hogy számtalan tárolási, megosztási és lekérdezési mechanizmus létezik ilyen modellekhez, emellett hatékonyan lehet nagy gráfokon műveleteket végezni, főleg abban az esetben, ha a gráfok elemei, például típusaik előfordulásának kötöttségei miatt bizonyos mintákat követnek. Így a gráfok alkalmasak adattárolásra, modellek mintáinak leírására és keresésére, valamint gráfalapon írhatjuk le a rendszerről alkotott tudásunkat úgynevezett ontológiák segítségével. Emellett a ma elterjedt okos város lekérdezési lehetőségek, úgynevezett API-k (alkalmazási felületek) tipikusan gráfmintákkal leírt, a konkrét technológiai megvalósítástól független módon írják le a lekérdezhető adatot. Ilyen leírásra példa a Resource Description Framework (RDF), a JavaScript Object Notation (JSON) vagy az eXtensible Markup Language (XML).

Ha nem ismerjük a nagy hálózatok pontos topológiáját, akkor elemzésre alkalmasak az úgynevezett véletlen gráfmodellek, mint a Barabási-Albert modell [8], illetve az Erdős-Rényi modell [9], amelyekkel jól közelíthető akár az internet, akár egy szociális háló felépítése. Ezeket a modelleket alkalmazták a bitcoinbányász-hálózat mint infrastruktúra tulajdonságainak leírására is annak érdekében, hogy a hálózat centralizáltságát vizsgálják [10].

2.2. FÜGGŐSÉGEK MODELLEZÉSE

Infrastruktúrák közti függőségek felfoghatók az egyes infrastruktúra-modellek elemei közti függőségnek, ahogyan azt az alábbi ábra szemlélteti:



4. ábra: Infrastruktúrák közti függőségek sematikus modellje

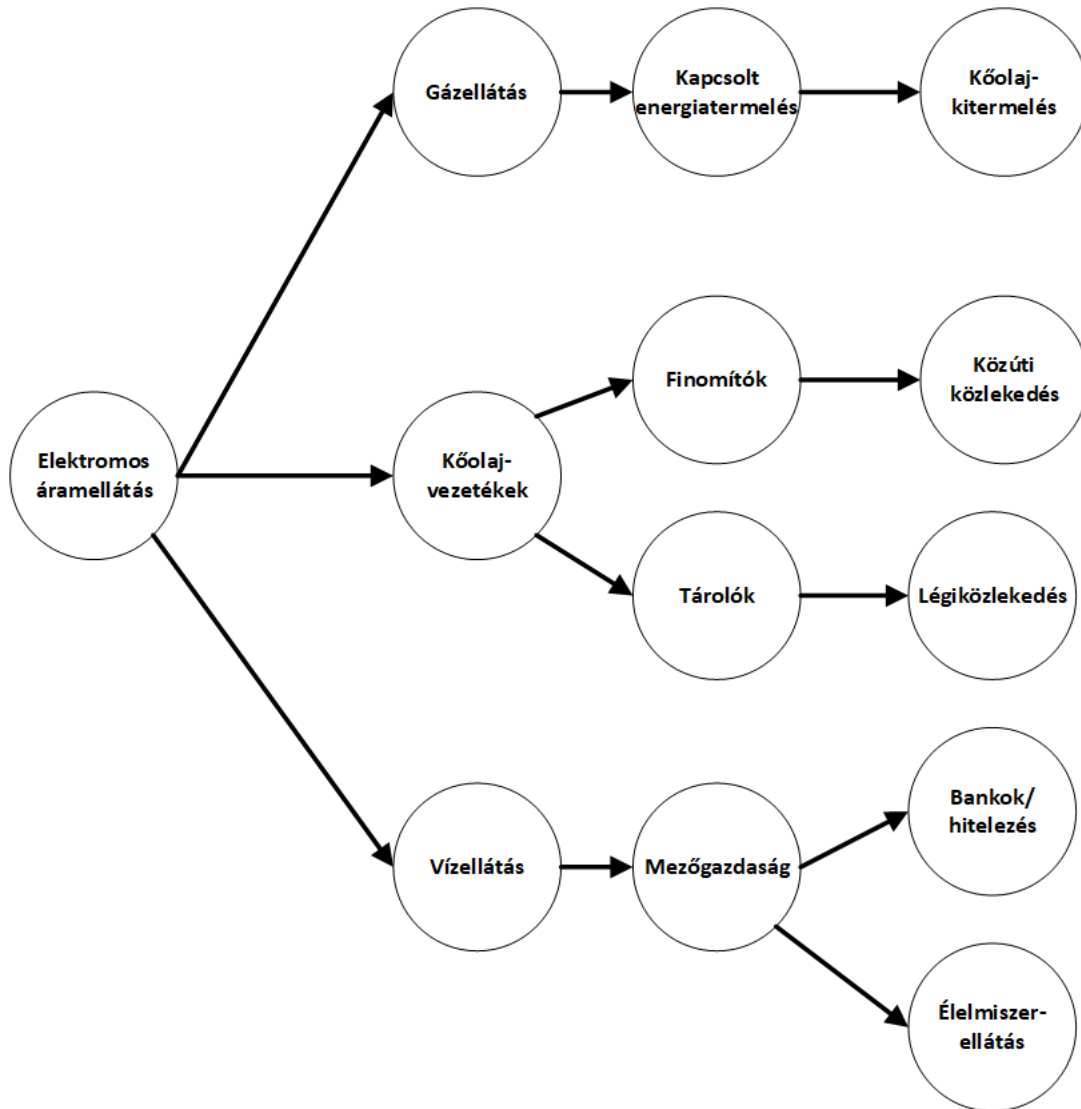
Forrás: a szerző saját szerkesztése

A bevezető fejezetben is említett függőségeket különböző felfogásban közelíti meg a szakirodalom. Az alábbiakban [4] és [5] megközelítését alkalmazva a következő főbb csoportokba sorolhatjuk ezeket:

- Empirikus/tapasztalati módszerek. A historikus adatokból, például hibaminták keresésével jobbra kvalitatív, azaz minőségi jellegű modelleket állítanak elő. Ezek a modellek tipikusan aránylag könnyen megalkothatók, azonban pont empirikus mivoltuk miatt kérdéses az általánosíthatóságuk.
- Rendszerdinamika (angolul: system dynamics) alapú módszerek. Ezek a módszerek kombinálják a mérnöki jellegű (például IDEF-0) leírónyelveket az ezeken a nyelveken leírt, tipikusan a komponensek felületén megjelenő, folytonos, differenciálegyenlet formájában leírható összefüggések kiértékelésével.
- Ágensalapú módszerek. Ezeknél az egyes szereplőket mint önálló viselkedéssel (akciókkal, preferenciákkal stb.) rendelkező entitásokat írnak le, és ezek együttes viselkedését vizsgálják.
- Közgazdasági módszerek. Ezek a módszerek alapvetően az anyagi javak, illetve a pénz áramlásán keresztül próbálják leírni az infrastruktúrák közti függőségeket.
- Holisztikus vagy átfogó megközelítések. Ezek a módszerek egy infrastruktúrát egy egész, tovább már nem bontott egységnek véve, alapvetően kvalitatív szempontok szerint vizsgálják az infrastruktúrák egymásra hatását, például makrogazdasági adatok figyelembevételével. Előnyük, hogy kezdeti modellek gyorsan felépíthetők így (például annak vizsgálatával, hogy az energiaárak változásának milyen hatása van a közösségi közlekedés igénybevételére), hátrányuk a becslésből adódó pontatlanság, illetve bizonyos jelenségek elnagyolt vagy hiányos modellezése. Ezek a modellek tipikusan alkalmazhatók magas szintű döntéstámogatásra és kockázatelemzésre, de nem alkalmasak operatív döntések validálására. Ilyen módszerre ad példát például [6].
- Hálózatalapú megközelítések. Ezek a megközelítések felépítik az egyes infrastruktúrák tipikusan gráfolapú modelljét, ahol a függőségeket a gráf csomópontjai közt behúzott élek jelentik. Ezen éleket behúzhatjuk

tervezési időben, azaz a modell kezdeti megalkotásakor, vagy bekerülhetnek a rendszer működésének vizsgálata közben, például rejtett függőségek esetén.

- Folyamalapú megközelítések. Ezeknél a rendszert mint egy folyamatot fogjuk fel, ahol az egyes infrastruktúrák közt adat áramlik, és ennek a hatását vizsgáljuk.



5. ábra. Az elektromos áramellátástól függő infrastrukturális elemek, példa

Forrás: a szerző saját szerkesztése [5] alapján

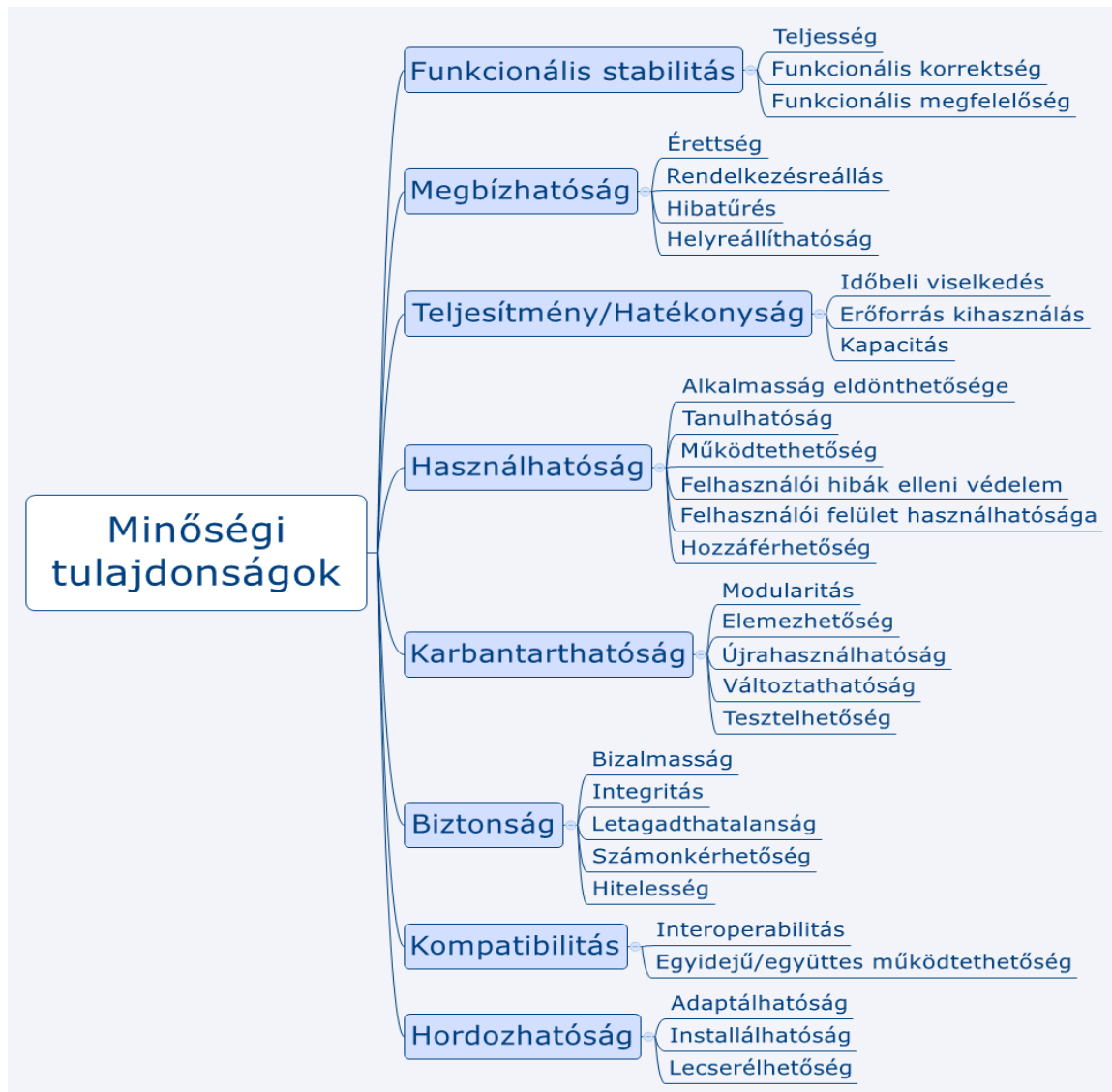
Az 5. ábra [7] kiterjesztésével példát mutat az áramellátástól való függések egy lehetséges modellezése, a 2003-as nagy kaliforniai (az USA több államára kiterjedő) áramszünet tanulságai alapján.

A fenti ábra valójában többfajta (fizikai, gazdasági stb.) függőség együttes kezelésével adja meg az elektromos áramkimaradáshoz mint hibához tartozó hatásláncot.

3. KRITIKUS INFRASTRUKTÚRÁK MINŐSÉGI TULAJDONSÁGAINAK MODELLEZÉSE

Kritikus infrastruktúrák esetében a funkcionális helyesség mellett a robusztussági követelmények igazolása is szükséges. Extrafunkcionális vagy nemfunkcionális tulajdonság alatt értjük egy rendszer azon tulajdonságait, amelyek a szorosan vett funkcionális elvárások mellett döntően meghatározzák a rendszer által nyújtott szolgáltatás minőségét.

Informatikai rendszerek követelményszabványában nemfunkcionális követelményként definiálja az ISO/IEC 9126 szabvány és utódja, az ISO/IEC 25010 szerint [11] az alábbiakat (6. ábra):



6. ábra: A termékminőség jellemzői

Forrás: ISO/IEC 25010 [11] alapján

Egy termék vagy szolgáltatás minőségének főbb aspektusai láthatók a 7. ábrán.



7. ábra: A használati minőség jellemzői

Forrás: a szerző saját szerkesztése [11] alapján

A minőségi modellek egy konkrét alkalmazására mutat példát [12]. Ebben a konkrét esetben felhőalapú szolgáltatások tervezésénél és felügyeleténél alkalmaznak egy „Key Quality Indicator” (kulcs minőségi mutató) modellt, ami elsősorban a teljesítmény, a működés pontossága, megbízhatósága és a biztonság kérdései felől közelíti meg egy rendszer működését. A modell alkalmazási lehetőségei kiterjednek mind a tervezési, mind a futási idejű feladatok támogatására, egyben releváns példát mutatnak a „smart” felhőalapú alkalmazások tervezésére.

Az alkalmazások/szolgáltatások **érzékenységvizsgálata** során felhasználható a minőségi modell mind a skálázási módszerek, mind a hibainjektálás tervezésére. Utóbbi során szisztematikusan hibákat próbálunk a rendszerbe juttatni, például célzott zavaró keresztterhelés, úgynevezett **interferencia** előállításával (ezzel a teljesítmény mellett a rendszer teljesítőképességét, vagyis a hiba során a hibakezelés többletköltségét is figyelembe vevő teljesítménymutatókat vizsgálva).

Az így nyert információt felhasználhatjuk a rendszer monitorozásának tervezéséhez, a rendszer alatti platform kiválasztásához, megoldási alternatívák összehasonlításához vagy a tárolandó adatok körének kiválasztásához.

A modellben meghatározott KQI (Key Quality Indicator) mutatók egyben hasznosak lehetnek a szimulációs kísérletek kiértékelésénél is.

Szimuláció során alapvetően egy rendszer viselkedésére adunk becslést anélkül, hogy a rendszert valós körülmények közt működtetni kellene. A szimuláció feltételezi, hogy a rendszer működésének kulcsfeltételei, tipikusan az erőforrás és időzítés jellegű paraméterek ismertek, valamint definiálhatók a rendszer működésének mérőszámai (például Key Performance Indicator formájában). A szimuláció előnye az analitikus modellek megalkotásával szemben, hogy könnyebben tudja kezelni a rendszer változásait, sok esetben egyszerűbben megalkotható és kiértékelhető, mint például egy valós, komplex rendszer differenciálegyenletekkel történő közelítése. Hátránya,

hogy a szimuláció eredményének, illetve maguknak a bemeneti paramétereknek és a modellezés során alkotott feltételezéseknek a validálása szükséges.

A szimuláció hihetőségét egyfelől alapvető kísérlettervezési módszerekkel, másfelől az adatok előzetes elemzésével és a megfelelően reprezentatív paraméterek kiválasztásával biztosíthatjuk.

Szimuláció során tipikusan megadható **bemeneti** paraméterek/beállítások:

- A rendszerbe beérkező kérések intenzitása, időzítése és eloszlása.
- Erőforrások száma, kapacitása és feladathoz rendelése.
- Az egyes feladat-erőforrás hozzárendeléseknél az adott feladathoz az adott erőforráson szükséges idő.
- Időzítési konfigurációs paraméterek (tevékenységek határideje, várakozási sorok maximális hossza, maximális várakozási idő feladatonként, erőforrások időbeli rendelkezésre állása stb.).
- A feladatok végrehajtásának egyéb feltételei (tevékenységek párhuzamosítási lehetőségei, túlterheléses helyzet esetén követendő stratégia, feladatok kiválasztási prioritása stb.).
- A szimuláció egyik fontos tulajdonsága, hogy az adott modell tudja-e a különböző felhasználói eseteket egységesen kezelni. Ennek egy lehetséges módja, ha a különböző inputokat (például egy pénzügyi rendszerbe beérkező különböző kéréseket) meg tudjuk különböztetni, és van lehetőség a bemeneti adat alapján tevékenységet/erőforrást/stb. választani. Így tudjuk modellezni azt is, ha egyes kivételes esetek kezelése másként zajlik, ami váratlan helyzetek, katasztrófák vagy rosszindulatú támadások modellezését is lehetővé teszi. Szimulációs eszköztől függően **értékfüggők** lehetnek a döntések (tehát minden döntési feltételnél a rendszer pontosan követi, hogy a valóságban egy adott kérést hogyan dolgozna fel), vagy a szimuláció során **valószínűségi alapon** döntünk. Utóbbi esetben például megadható, ha tipikusan a beérkező kérések 10%-a igényel rendkívüli kezelést. Egy adott folyamaton belül azonban ebben az esetben nem minden kérést tudunk pontosan lekövetni, hiszen több egymás utáni döntés véletlen választásánál előfordulhatnak olyan kombinációk, amelyek a valóságban nem, ez azonban tipikusan nem okoz jelentős torzítást.

Szimuláció során tipikusan **meghatározható rendszertulajdonságok**:

- A felhasználó/külső szereplő szemszögéből megfigyelhető tulajdonságok: válaszidő, áteresztőképesség, elutasított kérések száma/aránya.
- A rendszer üzemeltetője szemszögéből megfigyelhető további tulajdonságok: az erőforrások kihasználtsága, az egyes feladatoknál/erőforrásokon eltöltött idő, a rendszer komponensein megfigyelhető várakozási sorok stb.

A fenti tulajdonságok felhasználásával az alábbi **vizsgálatokra** alkalmas a szimuláció:

- **Szűk keresztmetszetek** keresése. Adott erőforrások és feladatok mellett mely erőforrás/feladat elvégzése korlátozza a rendszer teljesítményének növelését vagy több kérés kiszolgálását.
- Érzékenységvizsgálat. Általánosítva: milyen bemeneti paraméterek vagy paraméterkombinációk változása (angolul parameter sweep) hol okoz változást a rendszer megfigyelhető tulajdonságaiban. Egy összetett rendszer esetén ez jelentheti például annak vizsgálatát, hogy mely erőforrások kapacitásának változása okoz jelentős változást a felhasználók által érzékelt válaszidőben.

- **Általánosított „what-if” elemzések.** Az erőforrás-kiosztási stratégia, a terhelés vagy a rendszer erőforrásainak skálázása mind olyan kérdések, amelyeket több eset szimulációjával közelítőleg lehet tervezni.

A szimulációs megközelítéseket külön-külön sok szakterületen alkalmazzák (például közlekedéstervezés, energiaszektor), de nem elterjedt még az összekapcsolt területek együttes szimulációja. A váratlan helyzetek, illetve nem tervezett egymásra hatások felfedezhetők például a kölcsönösen lefoglalni kívánt vagy használt erőforrásokon keresztül, de a módszerből adódóan nem garantálható, hogy minden esetben felderíti ezeket.

A szimuláció fenti hiányosságait küszöbölik ki a formális modellellenőrzési módszerek. Ezen módszerek matematikailag bizonyított módon előállítják a vizsgált rendszer összes lehetséges állapotát (az úgynevezett teljes állapotteret vagy annak egy bizonyított módon helyes közelítését), majd azon matematikai formulaként megfogalmazott kifejezéseket értékelnek ki.

Komplex rendszerek esetében azonban ezek számos esetben nehezen alkalmazhatók, mert a rendszerek lehetséges állapottere olyan nagy lesz, hogy a kiértékelés gyakorlati korlátokba ütközik. A módszerek továbbfejlesztése mellett az infrastruktúra-modellezés szempontjából fontos a hierarchikus modellezés bevezetése, vagyis a rendszerek interakciós pontjai mellett komponens szinten megfogalmazott kérdések definiálása.

Kritikus infrastruktúrák szimulációjára mutat példát a [13], amely több szinten modellezve az infrastruktúrákat, megpróbálja szimulációval felderíteni az egyes (előzetesen kritikusnak gondolt) forgatókönyvek hatását.

4. ONTOLÓGIAMODELLEK HASZNÁLATA OKOS VÁROS LEÍRÁSÁHOZ

Az ontológia alapvetően olyan tudásleíró nyelvet és módszertant jelöl, ahol a világról alkotott tudásunkat egyedek és kapcsolatok képében akarjuk reprezentálni. Ezen egyedek és kapcsolatok alapvetően tulajdonságokkal rendelkeznek, mely tulajdonságok hierarchikusak lehetnek. Az egyedek és kapcsolatok leírására gráfalapú reprezentációt használunk. A kapcsolatok leírása finomítható, azaz megadható például, hogy egy közlekedési infrastruktúrában két elem össze van kötve, vagy ezen összeköttetés megfelel egy közúti vagy egy kötöttpályás közlekedési elemnek. Az iménti esetben a közúti és a kötöttpályás mint típus finomítja az összeköttetés általános típusát.

Emellett az ontológiák alkalmasak az egyed és a típus, pontosabban a fogalom és a példány kapcsolatának a leírására is. Ilyen lehet például az, hogy a Deák tér és az Astoria közötti metrószakasz a kötöttpályás kapcsolatok egy példánya. Amennyiben a világról alkotott tudásunkat ontológia formájában írjuk le, annak előnye lesz az, hogy az így leírt gráfalapú modell fölött egyrészt lekérdezési jellegű műveleteket hajthatunk végre, másfelől következtetési módszereket (inference) alkalmazhatunk.

A következtetési módszerek arra építenek, hogy a világról alkotott tudásunk alapján lehetséges-e bizonyos feltételek egyidejű kielégítése. Például igaz-e az, hogy két adott pont között létezik kötöttpályás és közúti közlekedés is. Fontos hozzátenni, hogy az ontológiaalapú modellezés jól illeszkedik az infrastruktúrák feladataihoz abból a szempontból is, hogy támogatja az úgynevezett nyílt világ szemantika szerinti modellezését. Ez, ellentétben a mérnöki modellezési nyelvekkel, például UML (Unified Modelling Language) által zárt szemantikával, azt feltételezi, hogy a modellünkben az elemtípus értékkészlete nem zárt, hanem kiegészíthető további elemekkel.

Az előbbi példánál maradva: az ontológiák lehetővé teszik azt is, hogy előzetes információ megadása nélkül tudjunk kezelni olyan útvonalat, amely egyfelől kötöttpályás, másfelől közúti. Amennyiben azt a megkötést be akarjuk vezetni egy modellbe, hogy egy útvonal egyszerre csak egy típussal bír el, azaz nem lehet egyszerre kötöttpályás és közúti, akkor ezt egy további, úgynevezett kényszer formájában tehetjük meg. Ez a fajta építkezés lehetővé teszi, hogy amennyiben a modellépítés elején a világról alkotott tudásunk még csak részleges, akkor azt a későbbiekben a modell felépítése közben finomítsuk.

Az ontológiamodellezés létrehozásának egyik fő motivációja az úgynevezett szemantikus web-konceptió megteremtése volt, amely a világháló terjedésével az egymástól függetlenül, szigetszerűen létrehozott adatforrások összekapcsolásának lehetőségét kívánta megteremteni. Ezzel a példával vezérelve az okosváros-konceptió megteremtésénél az ontológiák felhasználásának számtalan lehetősége van:

- **Szolgáltatások felfedezése** (discovery). Szolgáltatások vagy adatforrások felfedezésére alkalmas lehet, hogyha ezen adatforrások és szolgáltatások jellegét és főbb tartalmát hierarchikus, egymásra hivatkozó fogalmakkal (például tag jellegű modellel látjuk el, ahol ez a fogalmi modell építhet ontológia alapokra. Példa erre a [38], mely a LinkedData megközelítést használva tesz elérhetővé/kereshetővé okosváros-szolgáltatásokat/adatokat.

- **Szolgáltatásmodellek közötti összekapcsolás vagy leképezés tervezése** (mapping). Amennyiben két modell vagy szolgáltatás ontológiai alapja adott, ezen modellek között kölcsönösen egyértelmű leképezés tehető meg, ugyanakkor nem kell a két külön részmodell minden fogalmát, illetve minden kapcsolatát egyértelműen megfeleltetni a másiknak. Ennek előnye lehet például, hogyha két különböző város szolgáltatásait szeretnénk összekapcsolni, akkor ezen szolgáltatások összekapcsolásához csak a közös elemek és köztük lévő kapcsolatok egyértelmű leírására van szükség.
- **Megfelelőség (compliance) ellenőrzése.** Az okosvárosban definiált szolgáltatásmodell esetén a különböző szolgáltatások, illetve az ezen szolgáltatások által fogadott és visszaadott adatok megfelelése akár több megvalósítás között ellenőrizhető. Így például, hogyha a város környezeti paramétereit, térképét vagy aktuális forgalmi helyzetét egységesen szeretnénk kezelni, definiálható egy olyan közös forgalmi modell, amelyet az összes szolgáltatónak, például az összes tömegközlekedési szolgáltatónak közlekedéskérés esetén be kell tartania.
- **Egységes modellezési keret támogatáshoz.** Amennyiben városüzemeltetési elveket és házirendeket (policy) szeretnénk támogatni, elengedhetetlen, hogy a különböző infrastruktúra-elemeket egységes keretben tudjuk kezelni. Ennek egy lehetséges eleme, hogyha az összekapcsolt infrastruktúra egyes elemeit szakterületüktől függetlenül egy közös ontológia leíró nyelvvel kezeljük. Ez egyben azt is segíti, hogy a különböző leíró formátumoktól és szakspecifikus (közlekedés, építészet, elektromos hálózat stb.) leíró eszközöktől függetlenül tudjuk megadni a döntéstámogatáskor rendelkezésre álló más információk felépítését.

Üzleti folyamatmodellek okos városban történő felhasználásának egyik első lehetséges köre a kommunikáció céljából megalkotott modellek felhasználása. Ilyen lehet például az okos város egyik szolgáltatásának specifikációja, vagy egy tenderkiírásnál a rendszer már működő feladatainak kiegészítése. Ehhez hasonlóan a már említett (policy) rendelet jellegű előírásokat is folyamatmodellel lehet célszerű kommunikálni; ilyen lehet például a kedvezményes parkolás megigénylésének folyamata.

Okosváros-ontológiákat gyűjt és osztályoz a [14] weboldal, mely 2017. novemberében 123 ontológia adatait tette lekérdezhetővé a szabványos SPARQL (Semantic Search Query Language) nyelven. Az oldal emellett összegyűjtött 34 alkalmazást is.

Az itt elérhető leírások a közlekedéstől az egészségügyön, a kultúrán és az oktatáson keresztül a közigazgatáson át a környezetszennyezésig az okosváros számtalan aspektusát bemutatják. Jelen tanulmánynak nem célja az összes ontológia és konkrét modell bemutatása, de innen kiindulva jelentős mennyiségű információ áll az olvasó rendelkezésére, weben keresztül lekérdezhető, ugyanakkor strukturált modell formájában.

Kritikus infrastruktúrák minőségi paramétereinek és kockázati tényezőinek ontológiaalapú leírására ad példát [15].

Példaként említhetünk egy olyan módszert, amely eseményforrások ontológiaalapú besorolása alapján támogatja a szisztematikus eseményfeldolgozás tervezését [16].

Az eseményfeldolgozás tipikusan egy olyan terület, ahol előre nem feltétlenül ismert az eseményforrások tulajdonságainak összes lehetséges kombinációja, ugyanakkor szükséges bizonyos fontos eseményminták futásidejű azonosítása. Ilyen lehet például, ha egy kritikus erőforrás terhelése legalább T ideig eléri egy bizonyos szintet, miközben a tartalékul szolgáló erőforrás is terhelve van.

5. SZOLGÁLTATÁSOK MINŐSÉGÉNEK MODELLEZÉSE

Szolgáltatások informatikai megvalósításának egyik legelterjedtebb módja az XML (eXtensible Markup Language)-webszolgáltatások használata. Ezek alapvetően heterogén platformok felett megvalósított, egymástól elcsatoltan működő, integrált funkcionalitás megvalósítását támogatják.



8. ábra: Webszolgáltatások leírása

Forrás: a szerző saját szerkesztése

Webszolgáltatások leírásánál megkülönböztetjük a leíró dokumentum absztrakt és konkrét szakaszait, azaz azt, hogy szakasz foglalja össze az információkat, amelyek egyfajta specifikációként használhatóak fel. Ilyenek lehetnek például adattípusok definíciói vagy az interfész megadása. Egy interfészt műveletek összességéként definiálunk, ahol egy műveletnek van bemenete, kimenete, illetve visszatérhet hibaüzenettel. Mind a bemenet, mind a kimenet, mind a hibaüzenet az adattípusokra hivatkozik.

A leíró dokumentum konkrét szakasza tartalmazza egyfelől a kötéseket, amelyek a különböző műveleteket protokoll szinten definiálják, például hogy egy adott kérést TCP (Transmission Control Protocol)-protokoll fölött fogunk elküldeni, illetve magát a szolgáltatásdefiníciót, amelyet a végpontok definíciójával adunk meg. Ezen végpontok már konkrét hálózati címeket, pontosabban erőforrás-azonosítót (Uniform Resource Identifier, URI) tartalmaznak.

Szolgáltatások definíciójánál tervezési időben megadható mind az absztrakt, mind a konkrét specifikáció. A szolgáltatás leíró dokumentuma felhasználható arra is, hogy specifikációként pusztán az absztrakt szakasz meghatározásával megadja, hogy milyen jellegű viselkedést várhatunk el egy később megvalósítandó szolgáltatástól, melyek technológiai részleteit (például konkrét hálózati elérhetőségét) nem határozzuk meg. Ilyen módon a tervezési feladatok egyben modellfinomítási lépéseknek felelnek meg.

Léteznek iparági szabványok, például az energiakereskedelemben, ahol egyes gyakran használt műveletekhez szabványos leírók érhetőek el, amelyeknek közös az absztrakt szakasz része.

Hasonlóképp egy szervezet szolgáltatásainak publikálásakor dönthet úgy, hogy a mai világ számára csak az absztrakt szakaszt teszi elérhetővé, és csak a regisztrált partnerek láthatják a leírók konkrét szakaszrészzeit is.

Futási időben az úgynevezett webszolgáltatás-keretrendszerek feladata annak biztosítása, hogy az alkalmazások felől érkező üzeneteket lefordítsák arra a köztes nyelvre, amelyet a konkrét szolgáltatásprotokoll definiált (a kötések részben), majd eljuttassák a címzetthez az üzenetet. Ezeknek a rendszereknek a feladata egyfelől a konkrét alkalmazás programnyelvéről a webszolgáltatás reprezentációjára történő fordítás (például egy Java programozási nyelven megírt programban előálló, memóriabeli objektum leképzése egy programfüggetlen, „szöveges” XML-reprezentációra), másrészt az üzenet átküldése a hálózaton. Amennyiben a keretrendszer betartja a szabvány előírásait, akkor az így előállított üzenetek bármely más, hasonlóképp szabványos megvalósítást használó partnernél olvashatók lesznek, függetlenül az adott partner által használt programozási nyelvtől, operációs rendszertől stb.

Ez a fajta platformfüggetlenség teszi lehetővé például a banki, biztosítói rendszerek egységes szolgáltatásainak megvalósítását, és ezen az elven működnek többek közt az egységes tarifáló/broker alkalmazások, amelyeknél a broker és a konkrét biztosító közti kapcsolat a webszolgáltatás igénybevételére korlátozódik. Hasonlóképp léteznek webszolgáltatás-alapú interfészek az államigazgatás alrendszerei közt (például a MÁK felülete).

Ez a fajta integráció nemcsak nagyvállalati rendszerek fejlesztésében, hanem például infrastruktúramenedzsment esetén is megvalósítható, ahol a szétcsatolt alrendszerek közti kommunikáció webszolgáltatás-alapú.

Amennyiben szeretnénk a szolgáltatáshoz minőségi követelményeket is csatolni, és ezeket futásidőben kikényszeríteni, vagy legalább a megsértésüket jelezni (például üzenetek hibás vagy lassú kézbesítése esetén), akkor a webszolgáltatás-leíró és a futatókörnyezet kiegészítése szükséges. A korábban felsorolt extrafunkcionális követelmények közül jól támogatott a megbízható üzenetküldés (angol terminológiával Reliable messaging, [18]) és a biztonság (Security[19]). Ezekkel a szabványokkal, illetve az ezeket megvalósító keretrendszerekkel biztosítható, hogy ne csak az alapvető funkcionális követelményeknek („az üzenet érkezzon meg a címzetthez”), hanem a minőségi elvárásoknak is megfeleljen a kommunikáció („az üzenet megérkezésének sikerességében biztosak lehessünk”, „az üzenet módosítások nélkül érkezzon meg a címzetthez”).

Ahhoz, hogy ez megvalósuljon, szükséges a specifikáció módosítása, az interfész kiegészítése tervezési időben, illetve a borítékok/üzenetek módosítása és az módosításokat (például digitális aláírás beillesztése) kezelő kiegészítő modulok használata. Bár technikai szinten (a szolgáltatások leírásához csatolható XML-dokumentumok szintjén) a specifikáció könnyen módosítható, ennek modell szintű követése nem egységes, erre ad megoldási javaslatot az UML4SOA modellezési kiegészítés [20].

Futási időben a borítékok fejléce módosul, amelyre az XML-szabvány kiterjesztési mechanizmusa ad lehetőséget. Emellett a legtöbb futatókörnyezet megszorításokkal él a lehetséges „nemfunkcionális” (azaz az alapvető

működésen túlmutató) követelményekkel kapcsolatban, például ha egy adott interfészhez már titkosított csatornán történő elérhetőséget írtunk elő, akkor az adott interfész már nem publikálható titkosítás nélkül.

Hiányzik ugyanakkor a szolgáltatások fölötti tranzakciók, visszaléptetések, beszélgetési minták, koreográfiák definiálásának lehetősége, amelyre majd folyamat alapon lesz lehetőségünk.

Webszolgáltatások minőségének leírására ad példát egyfelől a Web Service Level Agreement (WSLA) XML-alapú leírónyelv [17], másfelől modell szinten az UML4SOA modellezési csomag [20].

6. FOLYAMATMODELLEZÉS

Folyamatmodellezés esetén tipikusan a rendszerek dinamikus viselkedését adjuk meg, ahol a viselkedést lépések egymásutánjaként értelmezzük. Ezek a lépések megadott munkatípusokon végeznek el megadott műveleteket. A művelet elvégzése függhet a beérkező munkadarabtól, például egy engedély jóváhagyása függhet az engedélykérés tartalmától, de az egyes lépések nem függenek más, előzetesen végrehajtott lépésektől, illetve a különböző folyamatok végrehajtása között nincsen közvetlen egymásra hatás. Ez azt is jelenti, hogy folyamatmodellel jól leírhatók olyan rendszerbeli történések, amelyek többször végrehajtandók, végrehajtásuk megengedett sorrendje és időbeli lefutása pedig eredeti szabályok mentén megadható.

Az üzletifolyamat-alapú fejlesztés széles körben elterjedt a nagy rendszerek szoftverjeinek a fejlesztésében, ugyanakkor gyakran alkalmazzák arra is, hogy házirend (policy) jellegű folyamatokat és előírásokat megadjanak.

Kritikus infrastruktúrák esetében ilyenek lehetnek például az infrastruktúra üzemeltetésének egyes részfolyamatai, valamint az infrastruktúra-üzemeltetés változásában bekövetkezett lépések egymásutánisága. Folyamat alapon lehet megadni például a rendkívüli esemény vagy katasztrófa esetén végrehajtandó lépéseket. Fontos megjegyezni, hogy az üzletifolyamat-modellező eszközök támogatják egyfelől az üzleti folyamat modellekből történő akár végrehajtható kódgenerálást, vagyis olyan rendszerek fejlesztését, amelyek működése bizonyítottan a folyamat által leírt logikát követi. Másrészt támogatják azt is, hogy az üzleti folyamat által leírt működést akár a kész rendszer működése nélkül, szimuláció útján ki lehessen próbálni, valamint össze lehessen hasonlítani különböző folyamatalternatívákat úgynevezett „what-if” (mi van ha?) analízis segítségével.

Fejezetünkben áttekintjük az üzleti folyamatmodellezés leginkább elterjedt szabványának számító BPMN-t (Business Process Modelling Notation), valamint példát adunk arra, hogyan lehet BPMN-folyamattal egy kritikus infrastruktúra üzemeltetésének lépéseit leírni. A későbbiekben példát fogunk látni arra is, hogy hogyan lehet BPMN-folyamattal leírt rendszerben az infrastruktúrának a szolgáltatásra gyakorolt hibáit, valamint a szolgáltatásoknak az infrastruktúrára gyakorolt hatását együttesen, kimerítő módon vizsgálni.

Folyamatmodellek további felhasználási területe az ellenőrzés támogatása; ennek egyik lehetséges formája, amikor magának a folyamatnak a helyességét vizsgáljuk meg. A folyamat helyessége alatt azt értjük, hogy a folyamat által leírt rendszer mentes bizonyos, a szakirodalomban definiált hibáktól. Ilyen hiba például a deadlock (holtpon) vagy a végtelen ciklus, de ilyenek lehetnek a különböző időzítési problémák is. Holtpontra vezethet például az, amikor egy folyamat két párhuzamosan végrehajtott ága ugyanazt az erőforrást akarja lefoglalni és ezáltal nem tudnak továbblépni. Ezeknek a hibáknak egy jó részét tervezési időben is fel lehet deríteni, azaz detektálható magának a modellnek a felépítése során, hogyha a rendszerben már potenciális veszélyforrás van. Ha már igazoltuk azt, hogy a folyamat által leírt rendszer nem mentes ezektől a hibáktól, akkor előírhatunk a rendszer működésére jellemző elvárt, úgynevezett biztonságos lefutásra jellemző mintákat.

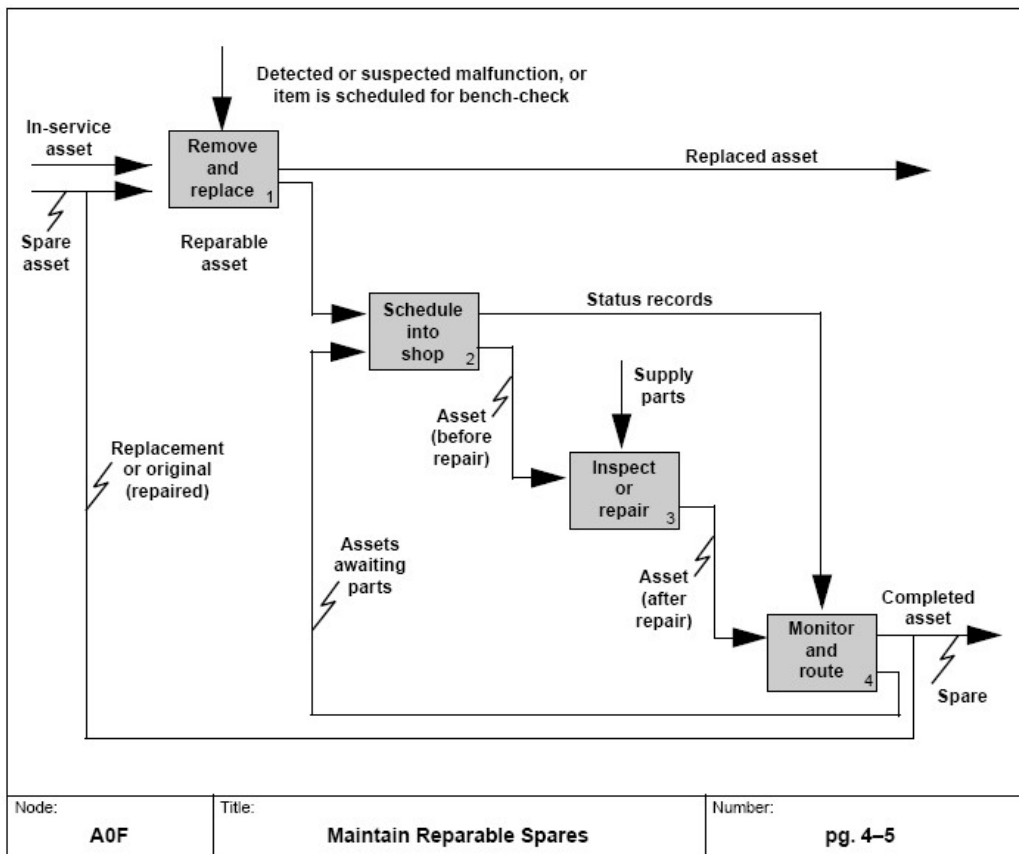
Ezeknek a feltételeit megfogalmazhatjuk olyan formában, minthogy például minden kritikus kérvényt legalább két különböző felhasználó lásson, vagy soha ne lehessen kiadni jóváhagyás nélküli igazolást. Ezeket a fajta mintákat megfogalmazhatjuk úgynevezett **invariánsok** formájában, ahol az invariáns szó arra utal, hogy a rendszer minden pillanatában igaznak kell lennie a követelménynek, azaz soha nem lehet olyan pillanat – az előző példára visszautalva –, hogy egy kérvényt kibocsájtottunk visszaigazolás nélkül.

Ezeknek a követelményeknek az ellenőrzéséhez sok esetben szükséges lehet az, hogy a folyamattal leírt rendszer működését is definiáljuk és elemezzük, ennek egyik eszköze a modellellenőrzés (lásd később). Emellett, ha a folyamatnak nemcsak a felépítését, hanem a folyamat által használt adatok helyes kezelését is ellenőrizni szeretnénk, akkor lehetőségünk van az adatok életciklusának és a folyamat működésének együttes vizsgálatára. Ennek egyik egyszerű példája az amerikai hadseregben is használt Bell–LaPadula-modell használata, amely vizsgálja a No-Write-Down és a No-Read-Up követelmények teljesülését. Ez azt feltételezi, hogy a rendszerben található felhasználók, illetve csoportok, valamint a dokumentumok rendelkeznek valamiféle minősítéssel, és vizsgálható ezen minősítések megfelelő kézben tartása.

6.1. IDEF-0

Az IDEF-0 leírónyelv az IDEF (Icam DEFinition for Function Modeling) modellezési nyelvcsalád eleme, többek közt az amerikai hadsereg vezette be az 1980-as években a beszállítóival történő egységes kommunikáció céljából. Az amerikai szabványügyi hivatal (NIST) által elfogadott szabvány, amelyet elsősorban adatintenzív alkalmazások leírásához használnak. Mind automatizált, mind humán folyamatok leírására használják, például a gyártás területén.

Az IDEF-0 alapvetően tevékenységekből épül fel, amelyeknek különböző jellegű függőségeik lehetnek. Ilyen például az adatfüggőség (input-output) vagy a funkcionális függőség (control), illetve a modell értelmezi az erőforrás (resource) fogalmát is. A modell mindig a bal felső sarokból indul ki, és lehetővé teszi a hierarchikus modellezést is egy-egy doboz tartalmának bővebb IDEF-0-kifejtésén keresztül.



9. ábra: IDEF-0 mintamodell

Forrás: <https://en.wikipedia.org/wiki/IDEF0>, CC licenz

Bár az IDEF-0 felépítéséből adódóan nem elsősorban végrehajtható specifikációk tervezésére kitalált nyelv, a függőségek és események kezelésével alkalmas az infrastruktúrák közti egymásra hatások leírására. [7] bemutat egy módszert arra, hogy lehet például a kommunikációs hálózat, az üzemanyagok, a vízellátás és az energiaellátás (mint egy-egy elemi egység) függőségeit IDEF-0-modellekkel ábrázolni, illetve ezekből a modellekből hibaterjedésre vagy allokációra (feladat-hozzárendelésre) utaló következtetéseket levonni.

A mintaként bemutatott scenárió egy áramkimaradás hatását vizsgálja, és annak fényében próbál optimális helyreállítási stratégiát (szakszóval: recovery) javasolni, hogy milyen előnyei lennének a kapcsolódó infrastruktúrákra nézve, illetve milyen költségekkel jár egy-egy stratégia. Az IDEF-0-struktúra, illetve -függőségek ebben az esetben matematikai egyenletekké (differenciálegyenlet) lefordítva felhasználhatók arra, hogy főbb összefüggéseket leírjanak. Ilyen lehet például a generátorok kapacitásának és az újraindítási időnek a kapcsolata, összevetve a gyorsabb helyreállítás idejével.

6.2. BUSINESS PROCESS MODELLING NOTATION (BPMN)

A Business Process Modelling Notation (BPMN) szabvány több korábbi folyamatleíró szabványt helyettesítve jött létre (többek közt az Event Driven Process Chain, a Value-Added Chain Diagramming, de bizonyos értelemben maga az IDEF vagy az UML [Unified Modelling Language] is beleértendő). A kezdetekben célja egy olyan jelölésrendszer definiálása volt, amelyet egységesen lehet használni folyamat alapú rendszerek leírására. Ma a BPMN 2 szabvány egy elterjedt, automatizálható, szimulációval ellenőrizhető szabvány, amelyhez számtalan implementáció létezik.

A BPMN szabványt felépítése kifejezetten alkalmassá teszi kritikus infrastruktúrák folyamatainak leírására, hiszen támogatja az alábbiakat:

- Pool: egy adott szereplő által végrehajtandó lépések összessége. Ilyen lehet például városüzemeltetés esetében a központi okmányiroda lépéseinek csoportosítása.
- Lane (sáv): egy adott szereplő által elvégzendő lépések egy csoportja.
- Események használata: egy folyamat indításakor vagy közben kezelendő, külső eseményre, belső hibára vagy időzítésre bekövetkező események definiálása. Alkalmas lehet belső hibák kezelésére, időzítési eltérések jelzésére vagy külső, a folyamat menetét befolyásoló események leírására.
- Vezérlési elemek: a teljesség igénye nélkül, a BPMN szabvány megengedi a programozási nyelvekben megszokott vezérlési szerkezetek használatát, úgy is mint elágazások (kizáró „VAGY”, vagy több választási lehetőséget megengedő), párhuzamos végrehajtás („ÉS”), kivételek kezelése (exception handling), újrafelhasználható alfolyamatok létrehozása (subprocess), adatobjektumok kezelése (data objects), folyamatok vagy folyamatrészek közti aszinkron kommunikáció leírása.

A BPMN nyelv által közvetlenül le nem fedett elemek:

- Erőforrások (Resource). Az erőforrások közvetlenül nem alkotják a BPMN nyelv részét, a legtöbb szerkesztőeszköz alapvetően az egyes lépéseket megvalósító alkalmazásokra történő hivatkozásokat támogatja. Léteznek ugyanakkor olyan modellezési eszközök, ahol egyes lépésekhez UML komponens diagram szintjén, illetve erre a kérdésre ad megoldást az ArchiMate nyelv is (lásd 6.5 fejezet).

- Időzítések. A BPMN alapvetően nincs felkészítve az elemek időzítés tulajdonságainak, illetve az erőforrások időbeli rendelkezésre állásának kezelésére. Ugyanakkor több modellezőeszköz is létezik, amelyek szabvány BPMN-diagramok kiegészítését engedik meg azáltal, hogy „nem szabványos” kiegészítésként ezeket a tulajdonságokat is modellezővé, sőt, szimulációval vizsgálhatóvá teszik.

Példaként [22] mutat be egy, a MOL BUBI kerékpárbérléshez hasonló, az okos város infrastruktúrájára és fizetési szolgáltatásaira építő folyamatot.

Az érdeklődő olvasó számos üzleti folyamat példát találhat például a Business Process Incubator oldalon [23].

6.3. A BUSINESS PROCESS EXECUTION LANGUAGE (BPEL) NYELV

Automatizált, több szervezeten/platformon átívelő folyamatok létrehozásának egyik elterjedt eszköze a Business Process Execution Language (BPEL). A BPEL nyelvet a nagy informatikai gyártók (többek közt az IBM és a Microsoft) együttesen hozták létre a 2000-es évek elején azzal a céllal, hogy az akkor terjedőben lévő XML webszolgáltatás-technológiák fölé adjanak egy üzletifolyamat-tervező réteget. A BPEL ennek megfelelően jól fel lett készítve az automatizált szolgáltatások kombinációjának/hangszerelésének (angolul: orchestration) problémáira, azonban nem kezelte kellő alapossggal azokat a kérdéseket, amelyek az üzleti kontextusból adódnak. Ilyen például az emberi beavatkozást igénylő lépések vagy az adatintenzív funkcionalitás kezelése, ahol sok esetben az adatobjektum állapota határozza meg a folyamat lefutását. Emellett a BPEL alapjául szolgáló SOAP (Simple Object Access Protocol)/WSDL (Web Service Description Language) technológiáról is bebizonyosodott, hogy merev felépítése és a viszonylag jelentős futásidejű többletköltsége miatt nem alkalmas minden szolgáltatás/komponens felületének leírására. Mindezen korlátok mellett a BPEL továbbra is széles körben használt, a legtöbb nagy gyártó által támogatott, ugyanakkor számtalan nyílt implementációval is rendelkező, integrált szolgáltatások létrehozását támogató modellezési/programozási nyelv.

A BPEL alapját az egyes aktivitások adják, melyekről alapvetően az a feltételezés, hogy azok leírása elérhető egy WSDL (Web Services Description Language) leíró részeként. Ezen aktivitásokat szervezhetjük végrehajtási egységekbe (scope), illetve vezérlési elemek (elágazás, döntés, párhuzamos végrehajtás stb.) segítségével építhetünk összetett folyamatokat. A BPEL emellett támogatja a hibakezelés mind kivételkezelésre, mind kompenzációra épülő megvalósítását.

A BPEL alkalmas a folyamatok specifikációjára és végrehajtására is, mivel lehetséges úgynevezett absztrakt és végrehajtható folyamatokat is modellezni. Előbbiekben nem kötelező például az összes szolgáltatás konkrét végpontjának megadása, illetve a folyamat egyes részei akár el is rejtethetők. Az absztrakt BPEL például alkalmas lehet arra, hogy egyfajta protokollt írjon le, vagy a beszállítókkal történő kommunikáció eszköze legyen, míg a végrehajtható BPEL egyben önhordó, dokumentált programkódnak is tekinthető.

A Business Entity Definition Language (BEDL, üzleti entitás definíciós nyelv [21] egy szabványosítási javaslat volt az IBM részéről, melyből szabvány nem keletkezett, de tervezőeszközbe beépült. Jelen tanulmány szempontjából kiemelendő, hogy a modell általánosságban írja le az adatobjektumok helyes használatát. A modell azt feltételezi, hogy a szoftverfejlesztésben is megszokott CRUD (Create, Read, Update, Delete – létrehozás, olvasás, módosítás, törlés) jellegű műveletek hajthatók végre egy objektumon. Ezek után az objektumon végzett különböző akciók (események hatására bekövetkező) végrehajtását, a felhasználók szerepköreit és a folyamat egyes lépéseit

összekapcsolhatjuk olyan módon, hogy például a dokumentumon végzett műveletek engedélyezése függhet attól is, hogy a folyamat mely részénél tartunk, így az előzőekben említett Bell–LaPadula-modellnél még rugalmasabb, az okos város folyamataihoz jobban illeszkedő leírási formát kapunk.

6.4. ÜZLETI SZABÁLYOK MODELLEZÉSE ÉS A DMN SZABVÁNY

A Decision Management Notation (DMN) szabvány [26] az Object Management Group konzorcium javaslata arra, hogyan lehet integrálni folyamatokat és szabályalapú döntéseket. Utóbbiak gyakoriak minden olyan területen, ahol a döntési folyamat főbb elemei (például egy beérkező esemény kiértékelése/minősítése, az eseményre adott válaszok/stratégiák kiértékelése, az esemény szükség szerinti elhárítása) mellett az egyes lépésekben (lásd az előbbi példánál maradva az események kiértékelésénél) a folyamat logikájától független, ugyanakkor az esemény paramétereire kapcsolódó kiértékelésre van szükség.

Ezen kiértékeléseket a gyakorlatban sokszor Business Rules (üzleti szabály) alapon valósítják meg. Az üzleti szabály-alapú fejlesztés nagy előnye az **externalizáció**, vagyis az egyes döntési pontok szétválasztása a döntések alkalmazási területeitől. Üzleti rendszerekben ennek klasszikus példája az áfaszámítás, amely egyszerre függ az értékesítésben részt vevő felektől és a termék jellegétől, független ugyanakkor a konkrét alkalmazási kontextustól (utólagos ellenőrzés, online tranzakciókezelés, előzetes kalkuláció stb.).

Az okos város alkalmazási területein belül különleges relevanciája van a szabályalapú megközelítés alkalmazásának. A szabályok ebben az esetben az egyes információforrások különböző területeken történő, ugyanakkor egységes alkalmazásának felelnek meg. Példa lehet erre egy város közlekedési csomópontjain kiépített, a tehergépjármű-forgalmat figyelő rendszer:

- A gépjárművek számossága alapján besorolható egy-egy csomópont vagy útvonal egy kategóriába (forgalmi szint szerint).
- A gépjárművek szállítmányuk szerint besorolhatók kategóriákba (kommunális szállítmány, általános kiskereskedelmi célú szállítmány stb.).
- A szállítmányok kritikussági/veszélyességi szintjük alapján is kategorizálhatók.
- A járművek tulajdonosai által fizetendő behajtási adó ellenőrizhető a jármű fajtája és a behajtás helye szerint.
- A jármű által szállítható árucikkek köre ellenőrizhető a járművet tulajdonló cég tevékenységi köre alapján.

Mint látható, a fenti szabályok megfogalmazása független a konkrét alkalmazási területtől, ugyanakkor együttesen szükségesek annak érdekében, hogy az adatok konzisztensen kiértékelhetők legyenek.

Ennek érdekében érdemes az adatok konkrét előfordulási statisztikáiból származtatott eredményeket és a javasolt beavatkozásokat együttesen kezelni, hiszen egy paraméterben bekövetkezett változás adott esetben más üzleti folyamatok indítását kezdeményezheti (lásd Trigger Event fogalma).

Az OMG erre a kérdésre adott megoldási javaslatot a Decision Modelling Notation (DMN) szabvány bevezetésével. Ezen szabvány az üzleti folyamatokba ágyazható, környezet-/kontextusfüggetlen szabályok definiálására ad lehetőséget, és mint ilyen, platform-/végrehajtókörnyezet-független.

A DMN által definiált szabályleíró nyelv lényegében illeszkedik az előtte elterjedt szabályleírók (például RuleML, [27]) szemantikai és szintaktikai megkötéseire, amennyiben egy szabály bal oldalán található, a szabály kiértéke-

lésekor ismertnek tekinthető modellrészlet bizonyos tulajdonságai alapján ad becslést/javaslatot a szabályalkalmazás után a modellrészlet tulajdonsághalmazának változására.

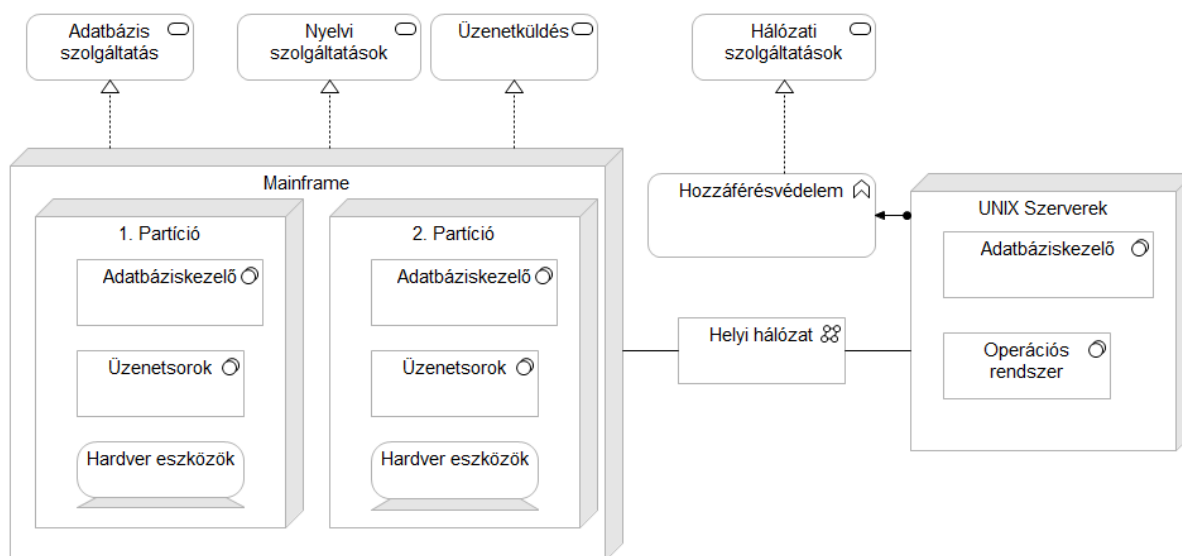
A DMN nyelv főbb felhasználási esetei a következők:

- Emberi döntéshozatal modellezése. Hierarchikus felépítése mellett alkalmas arra, hogy végigkövessük, mely adat milyen döntéseket befolyásol. Az okos város kontextusában ilyen lehet például a városüzemeltetési beavatkozási eljárások modellezése.
- Az automatizált döntéshozatal követelményeinek modellezésére. Abban az esetben, ha maga a döntéshozatali logika szabályalapú, ugyanakkor az implementáció más rendszerben történik meg (például egy SCADA – Supervisory Control and Data Acquisition jellegű rendszer beavatkozási szabályaiban), a DMN-alapú modellezés kapcsolatot teremthet a magas szintű tervezés, a szabályrendszer-ellenőrzés és a konkrét végrehajtás közt.
- Automatizált kiértékelések megfogalmazása és implementációja. Amennyiben olyan környezetben kerül egy szabályalapú logika megfogalmazásra, ahol ennek technikai feltételei adottak (például nagyobb rendszerek online integrációjánál), lehetőség nyílik olyan komponensek (például Java alkalmazásszerverek) üzemeltetésére, amilyenek DMN futtató/kiértékelő modulok működhetnek.

Az üzleti szabályok jól alkalmazhatók szimuláció támogatására.

6.5. A TOGAF MÓDSZERTAN ÉS A KAPCSOLÓDÓ ARCHIMATE MODELLEZÉSI NYELV

Nagyvállalati informatikai rendszerek modellezésében elterjedten használt a The Open Group Architecture (TOGAF) szabvány, amely egyszerre ajánl modellezési módszertant Architecture Development Method (ADM) néven és konkrét modellezési nyelvet és jelölést (ArchiMate). Utóbbihoz elérhető nyílt referenciainplementáció is (Archi), mely az Eclipse ökoszisztémára épül, így adott esetben kiegészíthető/integrálható olyan eszközökkel, amelyek mélyebb analízisfunkcionalitással rendelkeznek.



10. ábra: Infrastruktúramodell ArchiMate leírása

Forrás: a szerző saját szerkesztése

A 10. ábra egy okos gyár infrastruktúrájának egy részletét írja le, (<https://github.com/archimatetool/ArchiModels/tree/master/ArchiMetal> példa alapján). Látható, hogy a modell integráltan kezeli az egyes rendszereken belül megjelenő infrastrukturális elemeket (például üzenetkezelő alrendszerek, adatbáziskezelő stb.), és az ezekre épülő szolgáltatásokat egységesen jeleníti meg a modellezőeszköz, így akár a hardver szintű többszörözés szolgáltatás szintű hatása is elemezhető lesz (mint a többszörözött mainframe komponensek fenti esetében).

Az ArciMate és a hozzá kapcsolódó módszertan központjában a követelmények állnak. A módszertan összetett informatikai rendszerek architektúrájának tervezését és kezelését támogatja.

A TOGAF egyben kidolgozott egy modellt arra is, hogyan lehet felmérni egy szervezet szolgáltatásainak érettségét azok integrálhatósága szempontjából (Open Group Service Integration Maturity Model – OSIMM).

Az OSIMM hét szintet definiál a szolgáltatások érettségére vonatkozóan:

- Siló alrendszerek. A szakirodalomban silónak nevezik azokat a rendszereket, amelyek egymástól elszigetelve, lényegi adatáramlás vagy közös vezérlési folyamatok nélkül működnek. Ha ilyen rendszerek szolgáltatásait integrálni akarjuk, az szinte mindig jelentős újratervezést és esetekben manuális műveleteket, például az adatok kiegészítését igényli.
- Integrált rendszer. A rendszerek fel vannak készítve az integrációra, akár adat, akár folyamat szinten, azonban a konkrét összekapcsolás adattranszformációs műveleteket igényel, mert a rendszerek továbbra is elkülönülten működnek.
- Részekre bontott (angolul componentized) rendszer szintje. A rendszer szolgáltatásai szét vannak bontva, de tipikusan redundánsak, és egy funkció több helyen is implementálva van. Technológiailag fel van készítve a rendszer az integrációra, de üzleti elemzés tipikusan nem előzte meg az egyes komponensek/ szolgáltatások definícióját.
- Szolgáltatás szint. A rendszer alkalmazásai lazán csatolt szolgáltatások összességéből állnak össze, amelyek átnyúlhatnak az egyes szervezeti egységek határain. A szolgáltatások létrehozása többnyire jól definiált technológiai interfészeken keresztül lehetséges (például a már említett WSDL szabványt használva). Ezen az érettségi szinten azonban még tipikusan programozói/fejlesztői tudás szükséges új szolgáltatások létrehozásához.
- Összetett szolgáltatások szintje. A rendszer alkalmas arra, hogy alapvetően üzleti/folyamat szempontból (például a BPEL nyelvet használva) összetett szolgáltatásokat építsünk lényeges szoftverfejlesztési igény nélkül.
- Virtualizált szolgáltatások szintje. Az üzleti és a technikai szolgáltatások szintje elválik egymástól, így a szolgáltatást igénybe vevő kliens (hasonlóan a számítástechnikában máshol is alkalmazott „virtualizáció” megközelítéséhez) nem a tényleges, IT-infrastruktúra szintű szolgáltatást, hanem annak egy „virtualizált” változatát veheti igénybe. Ennek a szintnek előnye, hogy lehetővé teszi a különböző felhasználói csoportok számára különböző szintű, illetve adattartalmú/adatformátumú szolgáltatás kijánlását ugyanazon technológiai implementáció alapján.
- Dinamikusan újrakonfigurálható szolgáltatás. Ezen a szinten a szolgáltatások át-, illetve újrakonfigurálása már dinamikusan, futási időben is lehetséges, adott esetben például a megváltozott felhasználói igényekhez vagy a megváltozott környezeti feltételekhez igazodva.

Az OSIMM modell alapvetően (a szoftverfejlesztési cégeket vizsgáló érettségi modellekhez hasonlóan) arra használható, hogy egy-egy szerv, például egy állami infrastruktúra-üzemeltető működését felmérje abból a szempontból, hogy mennyire alkalmas rugalmas, ugyanakkor megbízható szolgáltatások nyújtására.

7. ADATVEZÉRELT MODELLEZÉS

Infrastruktúra-modellek megalkotásának egyik fő kihívása a modellbeli feltételezések és a valós, mért adatok összehangolása, illetve a modell számszerű paramétereinek meghatározása. Ilyen lehet például egy teljesítménymodell esetében a válaszidő, az áteresztőképesség vagy a kihasználtság, egy szolgáltatásbiztonsági modell esetében a hiba bekövetkezési valószínűsége vagy a rendszer által egy adott időszakban biztosított rendelkezésre állás.

Adatvezérelt modellezés során akár a már részben előre megalkotott felhasználói vagy mérnöki szintű modelleket kiegészítve, akár új modelleket alkotva a rendszerben mért adatok alapján alkotunk meg egy olyan modellt, amely a rendszer eddig megfigyelt viselkedésének megfelelő. Ez a fajta modellezési folyamat alkalmas például a magas szintű szakértői validálásra. Erre mutatunk példát a későbbiekben a katasztrófavédelem szolgáltatásának elemzésével.

Emellett felhasználható ez a fajta megközelítés modellek parametrizálására is, amikor egy már létező modell különböző számszerű paramétereit, például egy lépés átlagos lefutási idejét vagy következési gyakoriságát próbáljuk a mért adatok alapján meghatározni.

Kiemelten fontos alkalmazási területe az adatvezérelt modellezésnek a kiugró vagy ritka esetek (outlyer) azonosítása, amelyek nem illeszkednek a modell többi részéhez, jelenthetnek modellhibát, vagy épp ellenkezőleg, fontos kezelendő, ritka vagy kezelendő eseményt.

Az adatvezérelt modellezés használható arra is, hogy további vizsgálatokat, például szimulációt vagy statisztikai felmérést tervezzünk az alapján, hogy a rendszer mely viselkedésére nincs elegendő mennyiségű vagy minőségű adat.

Fontos lehetőség még a beavatkozási pontok beazonosítása, a leggyakrabban használt szolgáltatások azonosítása, illetve a rendszerben a kritikus lefutási utak, valamint a szűk keresztmetszetek keresése.

A feltáró modellezés eszközeiről bővebben ír [29], ismertetve a főbb diagramtípusokat aszerint, hogy egy-, két- vagy n-dimenziós adatok megjelenítését támogatják.

Egydimenziós plotokkal (ábrákkal) egy-egy változó (tapasztalati) eloszlását lehet ábrázolni, vagyis az egyes értékek vagy értéktartományok előfordulási gyakoriságát. Emellett kiugró értékek keresésére is alkalmasak, amelyek lehetnek mérési hibák, rossz modellalkotási feltételezések, tranzienst jelenségek vagy olyan, a háttérben eddig még fel nem tárt folyamatok eredményei, amelyeket a modellbe be kell építeni.

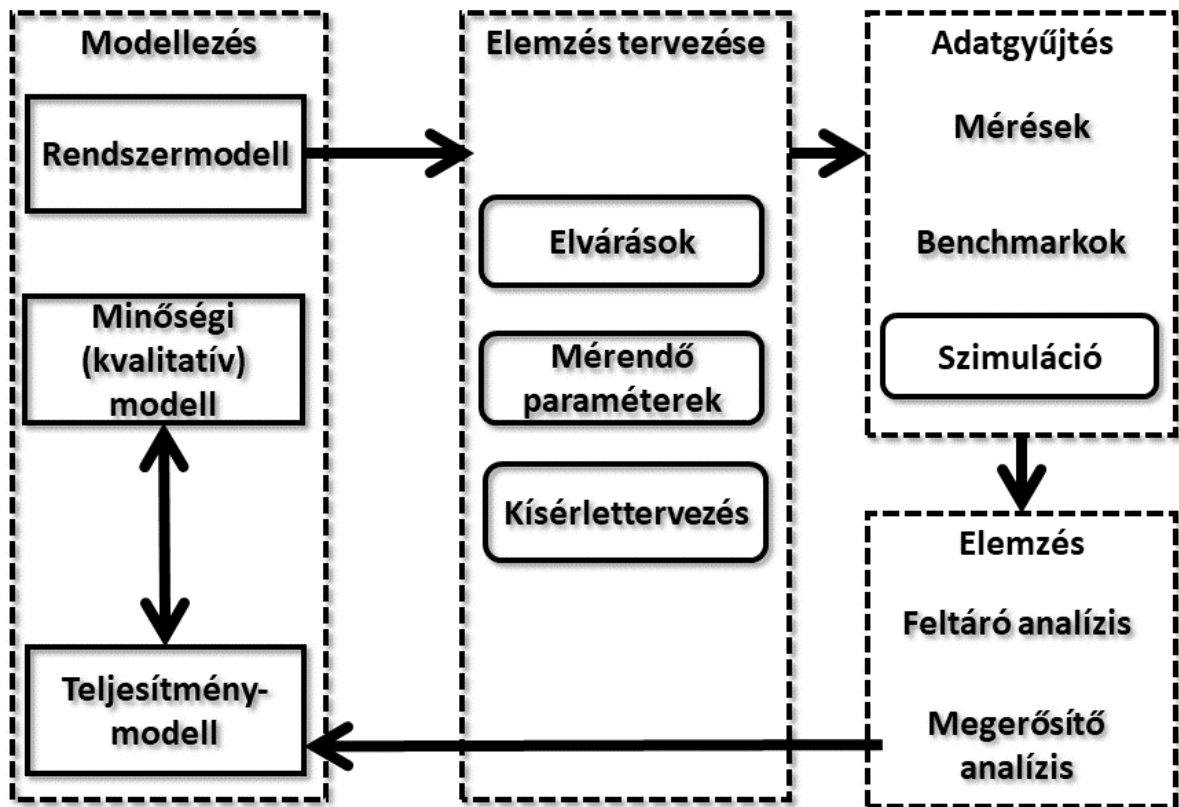
Kétdimenziós plotokkal alapvetően változók egymáshoz képesti viszonyát lehet leírni (melyik mennyire hat a másikra, milyen kombinációban milyen gyakran fordulnak elő együttesen, stb.).

Természetesen nem mindig lehet két dimenzióban jól leírni a rendszerbeli összefüggéseket, az összes változó-pár pedig túl sok vizsgálandó plotot adna ki. A 3D ábrázolás ezen részben segít, azonban például a diagnosztikai feladatoknál akár tízes nagyságrendű változó együttes megfigyelésére is szükség lehet.

Ennek eszköze a párhuzamos koordináták diagram, amelynek segítségével tetszőleges számú változót meg lehet figyelni akkor is, ha ezek dimenziója/mértékegysége teljesen eltérő[29].

Itt kiemelendő, hogy programozási tudás nélkül is lehetséges interaktív, színezés szerinti értelmezéssel (color brushing) ellátott, összekapcsolt kijelölést (angolul linked highlighting) támogató diagramokat, akár webes dashboardokat (összefoglaló jelentéseket) készíteni. Ezen diagramok sok esetben alkalmasak arra is, hogy a megjelenített vagy a kijelölt adathalmaz főbb statisztikai jellemzőit megjelenítsék (például medián, átlag,

szórás, minimum-maximum stb.), így segítve egy áttekintő nézet megalkotását. Emellett a diagramok könnyen kiegészíthetők egyszerűbb, trendelemzésre és előrejelzésekre építő matematikai eszközökkel (például regresszió), amelyek segítik egyben egy kvalitatív modell megalkotását. Például ha a tömegközlekedést igénybe vevő felhasználók számát vizsgáljuk, és a historikus adatok alapján a növekedés üteme az előrejelzés szerint gyorsul, akkor a dinamikus modelleknél ezzel az információval számolhatunk.



11. ábra: Modellezés és adatelemzés kapcsolata

Forrás: a BME Rendszermodellezés tantárgy anyaga, a szerző saját szerkesztése

A kapacitástervezés kérdéseiről bővebben ír az NKE-BME Okos város – okos közigazgatás kutatóműhely keretében elkészült kismonográfia [31].

7.1. TIPIKUS ADATFORRÁSOK ÉS KAPCSOLÓDÁSI PONTOK OKOS VÁROSBAN

Okos városok modellezéséhez számos külföldi és részben magyar példa is elérhető, amennyiben az adatvezérelt modellezés alkalmazási lehetőségeit vizsgáljuk. Ahogyan az alábbiakban bemutatjuk, számos nagyváros széles körben kínál hozzáférést az adataihoz, amelyeknek elérhetővé tételével többek közt az alábbi előnyökre tehet szert:

- Közvetlen visszacsatolást kap arról, hogy az adatok jól vannak kezelve, a valóságnak megfelelnek.
- A leginkább érintettek tudnak adatokkal alátámasztott javaslatokat adni fejlesztésekre.

- A technológiai fejlődés kihívásaiban segítséget kapnak a városok tipikusan erőforrásszükében lévő közigazgatási szervei.

Az egyik első, adatokkal támogatott, nyilvános elérést biztosító város San Francisco volt, amelynek adatai elérhetőek a <https://data.sfgov.org/> oldalon. A legtöbb adatkészlet néhány órás gyakorisággal automatikusan frissül a honlapon, ahol a többek között a város energiafogyasztása, közlekedési adatai, a környezeti terhelés adatai, a kiadott építési engedélyek statisztikái és a biztonsági incidensek is publikusan elérhetőek, de például az engedélyezett drónrepülések adatai is regisztrációkötelesek. Az adatkészlethez kapcsolódóan rendszeresen hirdetnek meg versenyeket, amelyeken valós problémákra keresnek megoldást.

A későbbiekben példával is bemutatjuk a 311 kóddal hivatkozott adatok felhasználását. A 311 a legtöbb észak-amerikai nagyvárosban működő hívószám, azokat a lakossági bejelentéseket fogadja, amelyek nem sürgősek annyira, hogy a 911-es segélyhívót legyen indokolt használni. Ilyenekre példa a zajártalom, a kátyúk, az elhagyott járművek, szemét, illegális területfoglalás stb. bejelentése. A 311-es hívások statisztikái alapján képalkotható a városi infrastruktúra és az üzemeltetés hiányosságairól, illetve ezek az adatok felhasználhatók arra, hogy az infrastruktúra-fejlesztés és a városrendezés projektjeinek bemenetet adjanak. Amerikában kifejlesztettek egy Open311 (vagy GeoReporting 2) nevű egységes alkalmazásfejlesztési felületet (API), melynek segítségével földrajzi koordinátákat, eseménytípusokat és akár médiaanyagokat (például videó- vagy hangfelvételt) kezelő alkalmazások készíthetők, amelyek az Open311 API-t használó városok közt hordozhatók lesznek. Az API specifikáció elérhető a <http://www.open311.org/> címen.

Hasonlóan elérhetőek Seattle adatai (<https://data.seattle.gov/>), New York (<https://opendata.cityofnewyork.us/>), London (<https://data.london.gov.uk/>), Berlin (<https://daten.berlin.de/>) vagy Hollandia (<https://www.cbs.nl/en-gb/our-services/open-data>) adatai is, mind elemzés, mind alkalmazásépítés céljából.

7.1.1. Adatok és API kezelése Európában – CitySDK

Az adatforrás jellegű felhasználási módon túlmutat az európai CitySDK kezdeményezés, amely a <http://www.citysdk.eu/> oldalon érhető el. Ennek keretein belül nem csupán az adatok elérésének formai adatait (hálózati cím, adatformátum), hanem a felületek elvárt működését is specifikálták.

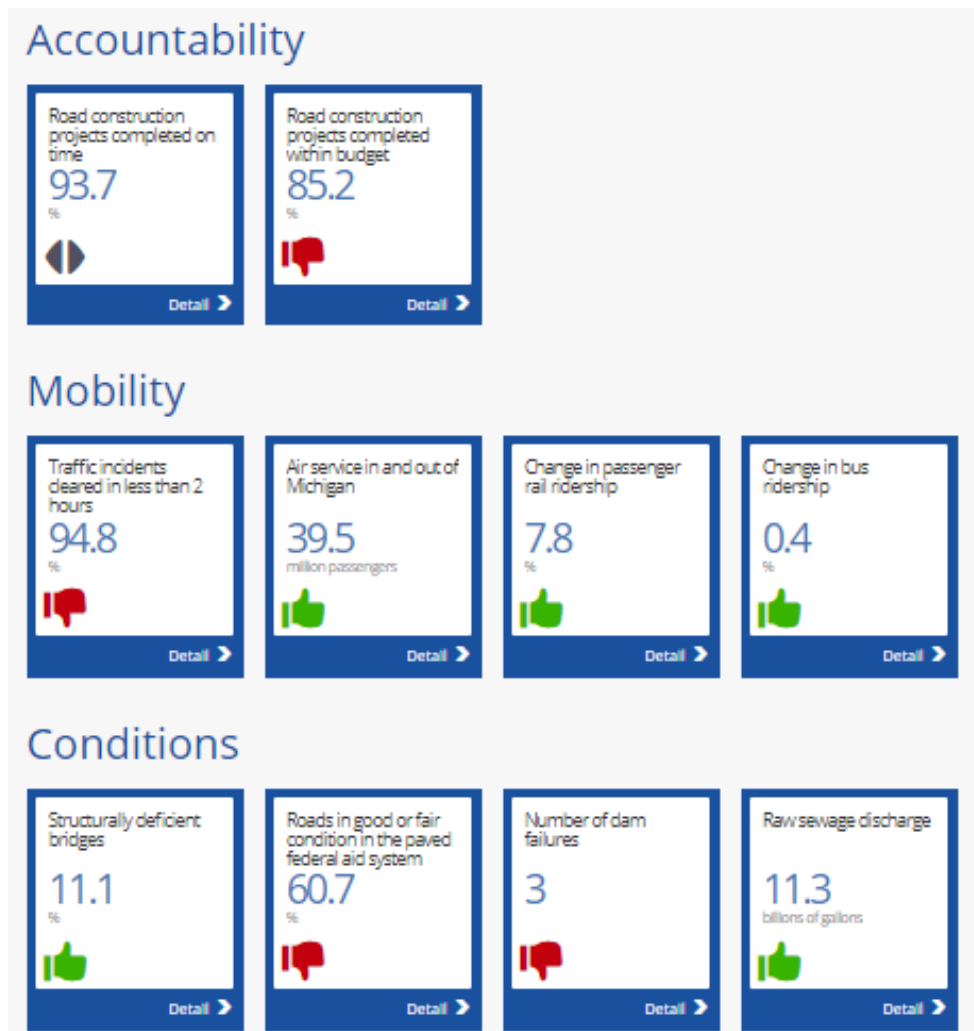
A CitySDK jelenleg 5 különböző szolgáltatáscsoportot definiál API formájában:

- Discovery service (felderítési szolgáltatás): Lekérdezhető az adott városban elérhető adatok és szolgáltatások.
- Open311 API: A már említett 311-bejelentéshez, illetve lekérdezésekhez kapcsolódó API.
- Linked Data API: Közlekedési, mobilitási és geográfiai adatok lekérdezése LinkedData formátumban, OpenStreetMap alapon.
- Tourism API (turizmus): A turizmus fejlesztését szolgáló, az eseményekre, az érdekes helyekre (Points of Interest, POI) és az ezekből alkotott útvonalakra fókuszáló felület. Érdemes kiemelni, hogy a POI mint adattípus külön szabványos definícióval rendelkezik, amelyet az internetes szabványokat gondozó W3C hozott létre. A POI adatstruktúra saját UML modellel is rendelkezik (<https://www.w3.org/2010/POI/documents/Core/core-20111216.html#poi-data-model>).
- LinkedEvents API: a városban tervezet események kérdezhetőek le.

A CitySDK erősen kötődik a 2014–2020 közt futó 6Aika Open Innovation Platform projekthez, mely a hat legnagyobb finn város okos szolgáltatásait támogató platformot és városüzemeltetési modellt dolgoz ki. A 6Aika egyben érettségi szinteket is definiál az innovációs platformokhoz, azaz voltaképp az okos város szolgáltatásainak fejlesztését támogató környezetet értékeli ki [32].

7.1.2. Adatvezérelt modellek a jó kormányzásban – Michigan

Az alábbi példa jól szemlélteti, hogyan lehet egy magas szintű mérőszámokra építő áttekintést adni egy város vagy egy nagyobb közigazgatási egység (esetünkben az USA Michigan tagállama) állapotára. Michigan kormányzójának felkérésére készült egy olyan oldal, amely az adatok elérhetővé tétele mellett néhány kulcsterületre koncentrált metrikát és annak változását mutatja be, így áttekinthetővé téve a kormányzat teljesítményét.



12. ábra: Michigan-dashboard, példa

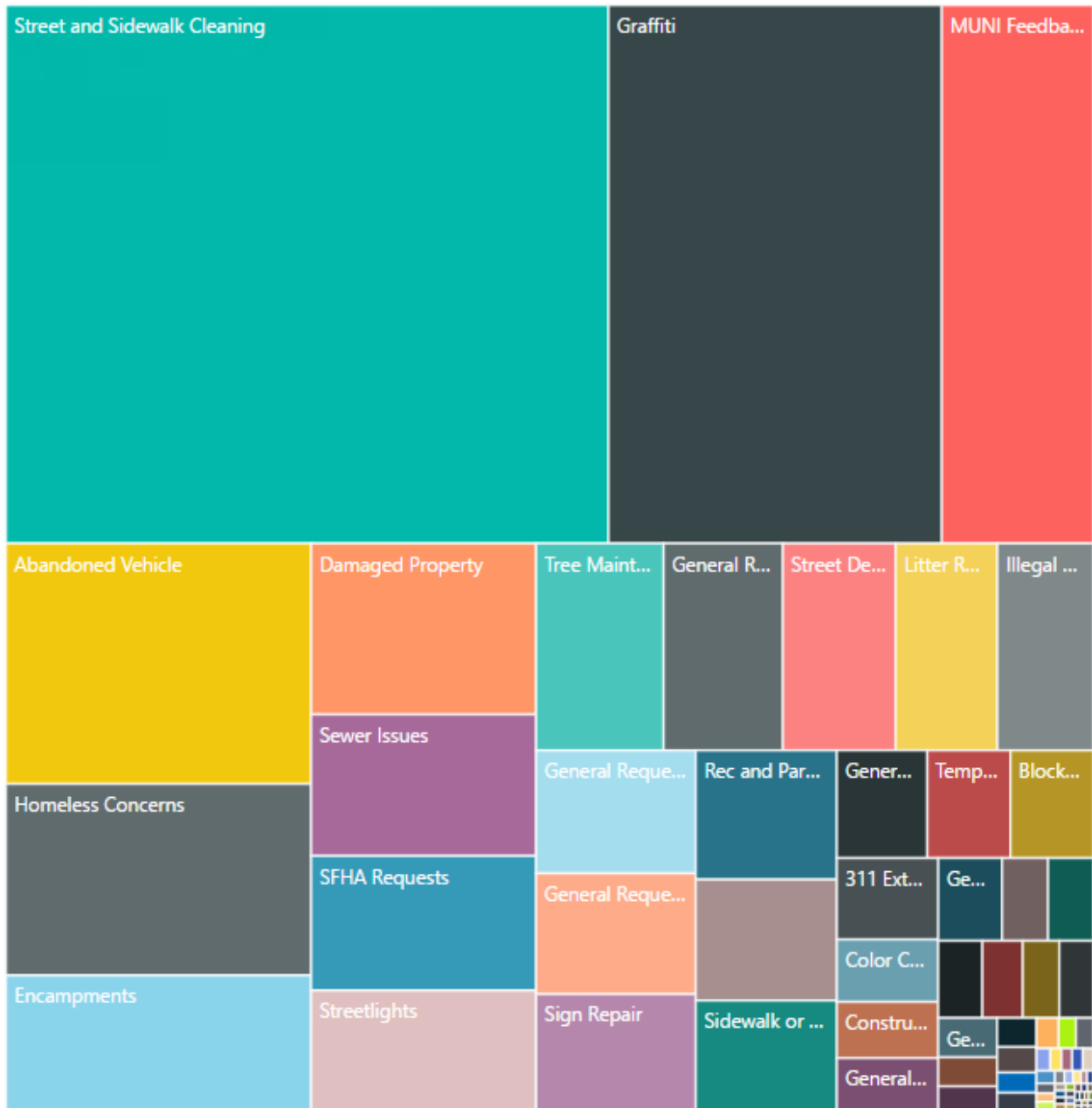
Forrás: Michigan város hivatalos oldala <https://midashboard.michigan.gov/midashboard>

A dashboard (irányítópult) főbb kategóriái: gazdaság, oktatás, egészségügy, biztonság, környezet, helyrajzi adatok, energiafelhasználás, valamint infrastruktúra. A fenti dashboard az infrastruktúra-elemek egy részének teljesítményét mutatja, a projektek elszámoltathatóságára, a mobilitásra és a környezeti feltételekre koncentrálna. A fenti ábrán például látszik, hogy a két órán belül fel nem számolt baleseti akadályok aránya romlott, amit idősoron is meg lehet tekinteni, sőt, a pontos metrikadefiníciók és -célok is elérhetők a Department of Transportation oldaláról (http://www.michigan.gov/openmichigan/0,4648,7-266-60201_60965---,00.html).

Bár ezek a mérőszámok önmagukba nem alkalmasak a döntéshozatal támogatására, alkalmasak lehetnek arra, hogy segítségükkel az infrastruktúra-modelleket, az azokon végzett szimulációs műveleteket és az előrejelzések/egyéb vizsgálatok eredményeit validálni lehessen, valamint az egyes vizsgálatokhoz célmutatókat definiáljunk.

7.2. ESETTANULMÁNY: VÁROSÜZEMELTETÉSI BEJELENTÉSEK KIÉRTÉKELÉSE

Az alábbiakban példát mutatunk arra, hogyan lehet az adatokat abból a célból felhasználni, hogy magas szintű, kvalitatív modellt alkossunk egy terület működéséről. Példánkban a már említett 311-es szám San Francisco városából származó statisztikáit vettük alapul. Az adatkészlet 2008–2017 közti bejegyzéseket tartalmaz, összesen több, mint 2,5 millió sorból áll. Az elemzéshez a Microsoft Power BI eszközt használtuk, de más önkiszolgáló üzleti intelligencia eszközzel is elérhető hasonló eredmény. A továbbiakban látható ábrák szemléltetik egyben a csatolt kijelölés (linked highlighting) és a lefűrés (drill-down) nézet – és modellkezelési műveletek használatát is.

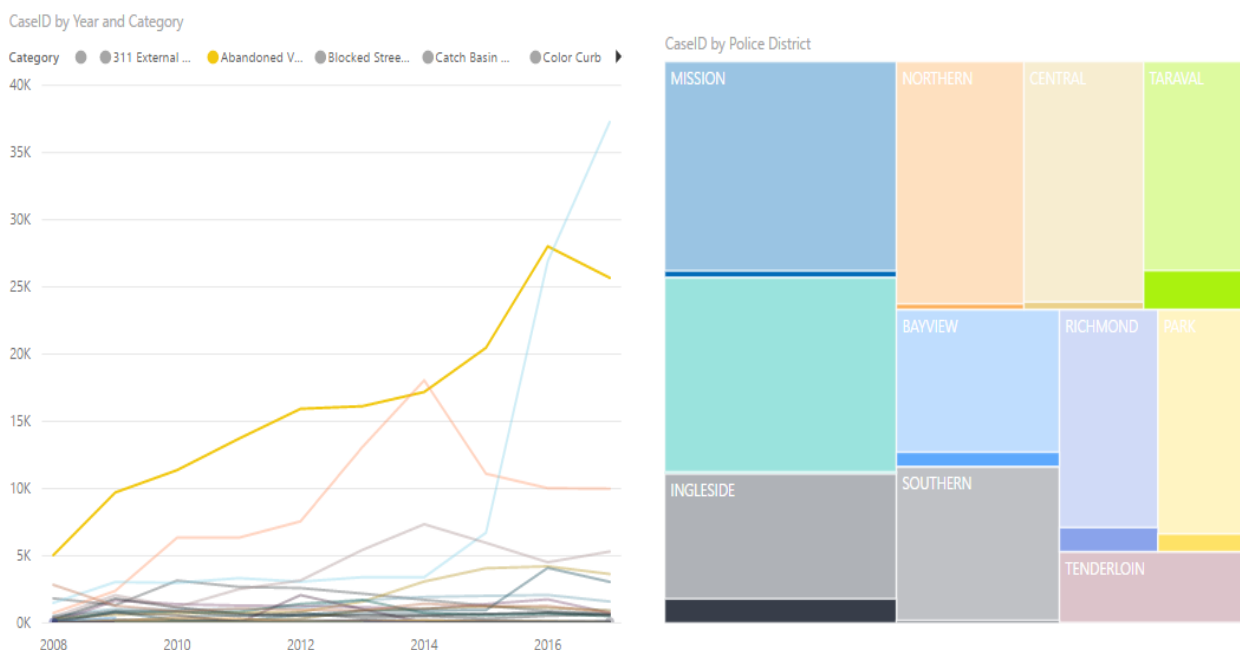


13. ábra: Incidensek eloszlása típusuk szerint

Forrás: a szerző saját szerkesztése

A fenti besorolás lehet egy adattípus szerinti osztályozás finomításának alapja, illetve az ontológia tapasztalati adatok alapján történő súlyozása.

Eddig a rendszer egy statikus képét néztük, az alábbiakban figyelembe vesszük az időbeli változások hatásait is. Időbeli változás figyelembevétele:



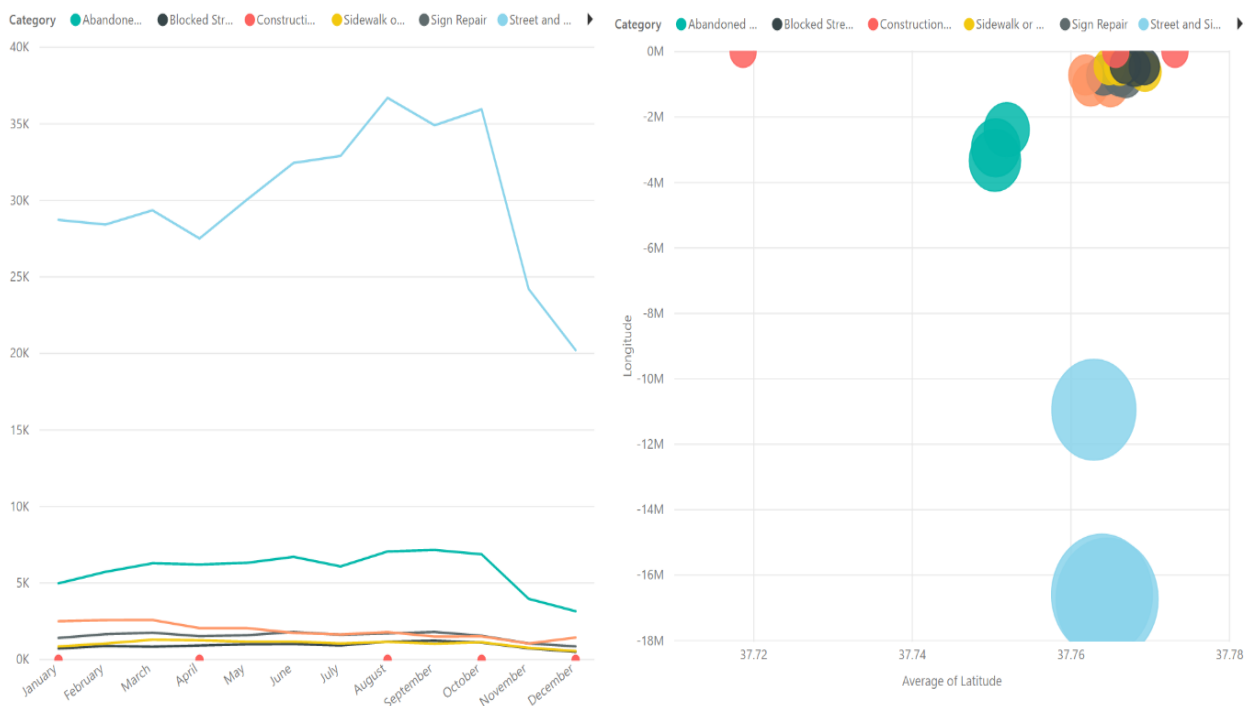
14. ábra: Események eloszlásának idő és hely szerinti változása

Forrás: a szerző saját szerkesztése

A 14. ábra az egyes kategóriába eső bejelentések számának alakulását és az egyes területi egységeken belül a bal oldali ábrán kiválasztott kategória (elhagyott járművek) arányát mutatja.

- Az idősoros elemzés, idősorok korrelációja és előrejelzése nem képezi ennek a tanulmánynak a fókuszát, azonban kiemelendő, hogy a fentihez hasonló idősorok alkalmasak lehetnek magas szintű, kvalitatív empirikus modellek megalkotására. A fenti ábrán például az elhagyott járművek bejelentési gyakoriságának drasztikus növekedése és az egyes körzeteken belül az ilyen bejelentések aránya látszik. Ezek alapján intézkedések tervezhetők, illetve a körzetek kategorizálhatók lehetnek.
- Ugyanakkor az egyes kategóriák fontossága és a lényeges időbeli változások a modell validálására felhasználhatók, valamint alkalmasak lehetnek arra is, hogy detektáljuk a rendszer viselkedésének lényeges változásait (például azért van-e több elhagyott jármű, mert változtak a parkolási szabályok, vagy a bejelentés lett könnyebb).
- Megvizsgálandó, mi lehet az oka annak, hogy egyes területeken lényegében nincsenek elhagyott járművekre vonatkozó problémák, milyen összefüggésben van ez az adott terület egyéb rendelkezéseivel, sajátosságaival, infrastrukturális kiépítettségével.

A fenti módszer egyben alkalmas arra is, hogy az összekötött infrastruktúrák lehetséges szűk keresztmetszeteit vizsgálja, ezáltal segítve olyan modellek hiteles megalkotását, amelyek várostervezési/üzemeltetési szempontból segítséget jelentenek.

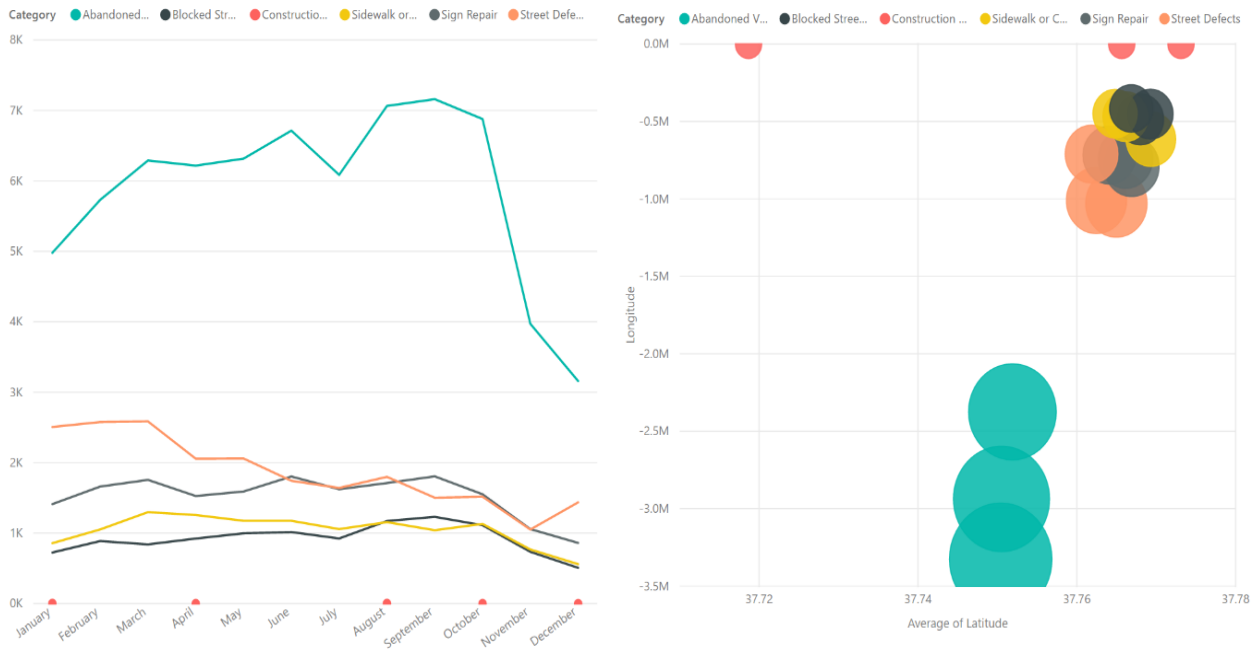


15. ábra: Városi incidensek időbeli eloszlása, egy kiválasztott terület incidensei

Forrás: a szerző saját szerkesztése

A következőkben példát adunk arra, hogyan lehet egy modell elemeinek értékkészletét finomítani, illetve kategorizálásra felhasználni az adatokat. A 15. ábra mutatja egy kiválasztott terület közlekedési bejelentéseit 5 éves időszakra, havonkénti bontásban, illetve jobb oldalon földrajzi koordináták szerinti eloszlásban. Kiemelkednek a tisztítással összefüggő incidensek, amelyeket érdemes eltávolítani, egyben azonosíthatók azok a területek, ahol ezek a leggyakoribbak. Ezek után azonosíthatók azok a területek, ahol a leggyakrabban előfordulnak a nem az utcák pillanatnyi állapotával, hanem az infrastruktúra jellegű hibaokkal összefüggő problémák, valamint felvethető ezeknek egy közös azonosító kategória létrehozása is (absztrakció).

A 16. ábra szerint egyértelműen azonosítható az a földrajzi terület, ahol ezek a problémák kiemelkedően jelentkeznek, valamint azonosítható például az is, hogy az elhagyott járművek száma egyfelől kiemelkedően magas, másfelől megugrik a nyári hónapokban. Az ilyen jellegű, a számszerű adatokból származó kvalitatív megfigyelésekből építhető olyan döntéstámogató modell, mint például az adott területeken történő, időszakosan gyakoribb ellenőrzés, vagy az adott terület kameraképeinek célzott feldolgozása adott időszakban.



16. ábra: A közlekedési infrastruktúra hiányosságaira utaló incidensek eloszlása idő és területenkénti bontás szerint

Forrás: a szerző saját szerkesztése

A hierarchia szintjén egyet lejjebb ereszkedve (kihasználva, hogy az absztrakciós/finomítási műveletek hierarchikus adattípus, így a dátum esetén támogatottak), vizsgálható az egyes incidenstípusok időbeli eloszlása, akár nap pontossággal.

Mire használható egy ilyen kezdeti modell?

- Megadható, milyen hiányosságok kiküszöbölésére érdemes például a lakosság fele nyitott okos szolgáltatásokat indítani. Az előző példánál maradva, érdemes a 311 bejelentő alkalmazásnál kiemelni a gyakran szereplő incidenstípusokat, ezáltal használhatóbbá téve az alkalmazást.
- Osztályozhatók az infrastruktúra bizonyos felhasználási/működési területei kihasználtságuk, hibaarányuk, használati jellegük stb. szerint. Ez felhasználható arra, hogy üzemeltetési politika szinten különböző beavatkozások legyenek tervezhetők.
- Az incidenseeloszlás ismerete felhasználható kapacitástervezési lépésekben is. Ez jelenti egyfelől a bejelentési szolgáltatás kapacitásának tervezését (például bizonyos típusú gyakori kérések kiszolgálásának átszervezését), másfelől inputot szolgáltathat az infrastruktúra szintű ütemezett karbantartások tervezéséhez is. A fenti, kezdeti modell mutatja például, hogy mely részeken lehet az infrastruktúrában szűk keresztmetszet azáltal, hogy sok közlekedési akadályt jelentenek be.

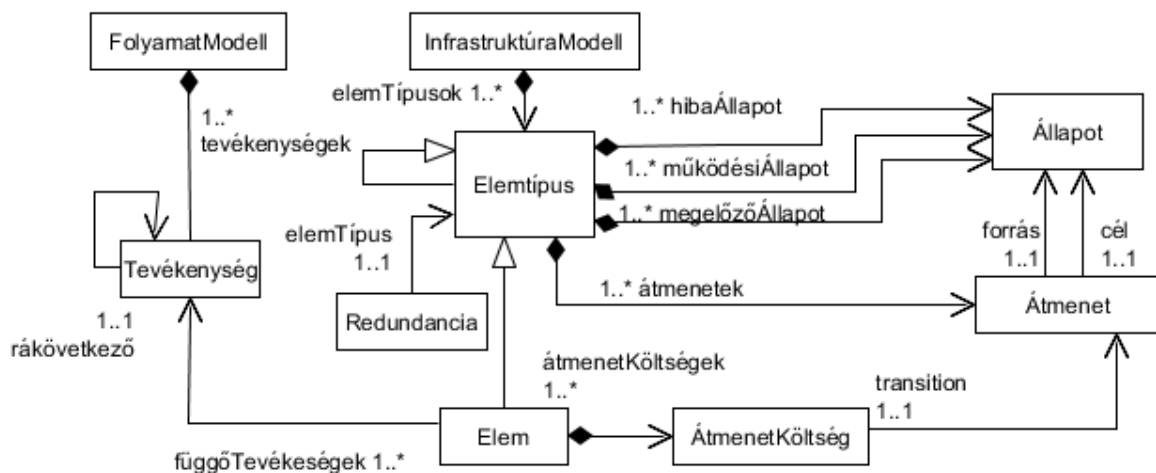
8. MODELLEZÉSI ESETTANULMÁNYOK

Az alábbiakban esettanulmányokat mutatunk be kritikus infrastruktúra modellek felépítésére, illetve használatára.

8.1. KRITIKUS INFRASTRUKTÚRÁK OPTIMALIZÁLT REKONFIGURÁCIÓJA

Kritikus infrastruktúrák üzemeltetése kapcsán az egyik fő kérdés az infrastruktúra és a ráépülő szolgáltatások változtatásainak tervezése. Ennek tervezéséhez fel kell mérni a rendszert alkotó főbb elemeket, a rendszer állapotait, az ezek közti átmeneteket, valamint (amennyiben ez meghatározható) az átmenetek költségét. A ténylegesen lehetséges átmenetek és azok költségének meghatározása nagyban segíthető a rendelkezésre álló adatok módszeres feldolgozásával.

A TÁMOP-4.2.2.C-11/1/KONV-2012-0004 által támogatott munka alapján az alábbiakban példát mutatunk egy általánosított kritikus infrastruktúra leíró modellre, amelyet kifejezetten a rekonfiguráció modellezésének célja vezérelt [33].

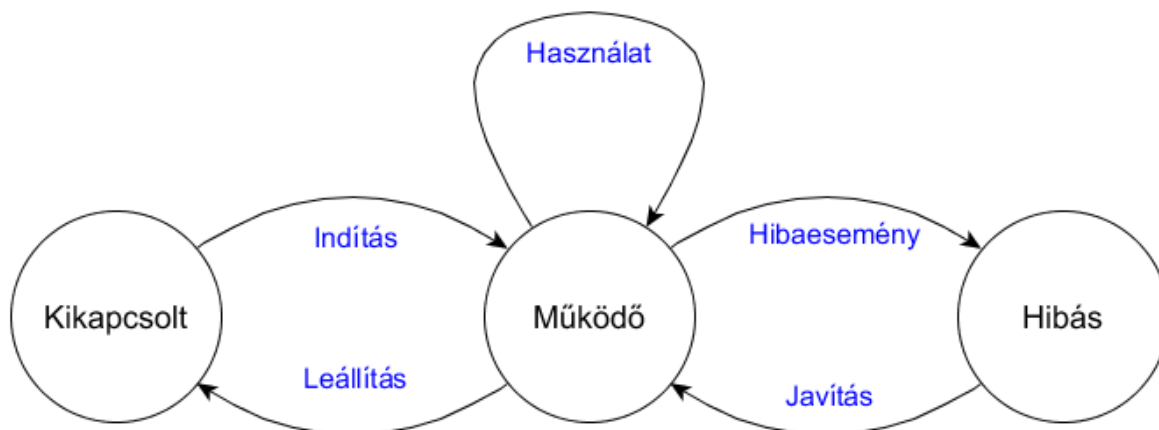


17. ábra: Példa metamodell kritikus infrastruktúrák változásainak leírásához

Forrás: a szerző saját szerkesztése [33] alapján

A fenti ábra a kritikus infrastruktúrák leírásának egy lehetséges metamodelljét adja meg, követve a szoftvermodellezés során használt jelölési konvenciókat (szavak írásmódja stb.). A metamodell tartalmazza az értelmezett elemeket és a köztük lehetséges kapcsolatokat. Érdekes megfigyelni, hogy a statikus és dinamikus vonatkozásokat együtt is ábrázolhatjuk, tehát a fenti modell (amely jellegét tekintve követi a TOGAF elemeit, bár nem felel meg annak egy az egyben) egyszerre alkalmas topológia és viselkedés jellegű vizsgálatok támogatására is.

Természetesen a fenti elemeket, elsősorban a rendszer felépítésének elemeit, nem elsődlegesen a fenti eszközben érdemes definiálni, hanem más, szakterületspecifikus eszközökből lehet a fenti eszköz elemkészletét ismerő modellező rendszerbe importálni.



18. ábra: Rekonfigurációs állapotmodell

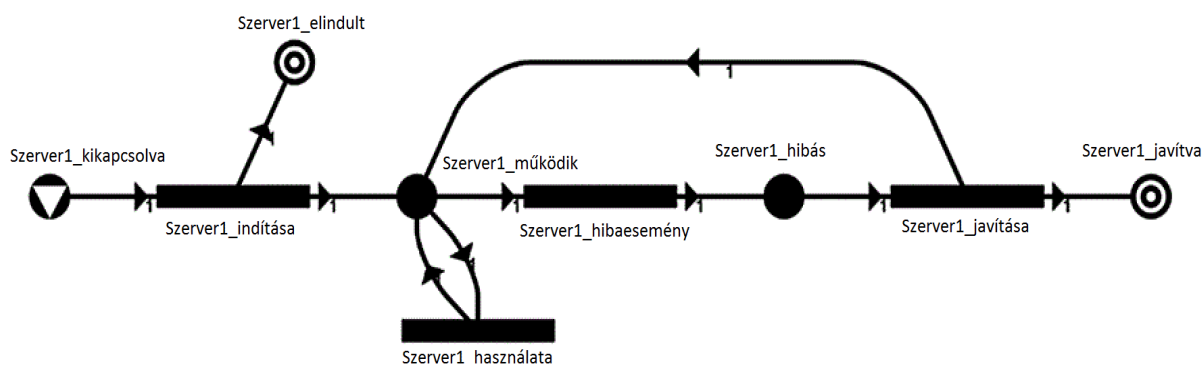
Forrás: a szerző saját szerkesztése [35] alapján

A fenti ábra a rendszer egy komponensének állapotalapú, dinamikus modelljét mutatja. A modellben az állapotok közt címkézett átmenetek vannak, ahol a címke megfelel annak a tevékenységnek vagy eseménynek, amely az átmenetet kiváltja. Fontos megjegyezni, hogy ez magas szintű tervezési modell, amely a rendszerkomponens állapotait összevonja, és csak a megbízhatósági elemzés szempontjából érdekes aspektusokra koncentrálnak. Hasonlóan az egyes átmenetek is megfelelhetnek több valódi eseménynek (például értelemszerűen mind a javításnak, mind a meghibásodásnak több fajtája lehet), a modell szempontjából egységesen kezeljük ezeket. További megkötés, hogy ezekben a modellekben tipikusan független meghibásodásokra és „örökifjú” viselkedésre készülünk, azaz feltételezzük, hogy egy adott hiba bekövetkezése és annak javítása egyfelől nincs hatással más hibák előfordulására, másfelől nem befolyásolja az adott hiba későbbi előfordulási gyakoriságát.

A fenti állapotgépből úgynevezett P-gráf modell készíthető [34] [35]. A P-gráf egy olyan matematikai formalizmus, mely speciális páros gráfokon alapszik. Eredeti célja gyártási folyamatok modellezése és optimalizálása, azonban használható optimális rekonfigurációs folyamatok modellezésére is.

A P-gráfok elemei megfelelnek egyfelől a gyártási lépéseknek (téglalappal jelölve), másfelől az anyagoknak, illetve termékeknek (kör alakú ikonokkal jelölve). Az egyes gyártási lépésekhez megadhatók kimenetek, illetve bemenetek, valamint költségek is. Az infrastruktúra-rekonfiguráció esetében ez megfelel az egyes rekonfigurációs lépések elő-, illetve utófeltételeinek, valamint a rekonfiguráció valamilyen, például időben mért költségének. Példa lehet egy tartalék erőforrás elindítása mint művelet, amelynek egyfelől bemenete az elérhető hidegtartalék (kikapcsolt) erőforrás, másfelől kimenete a már működő, bekapcsolt tartalék. Technikai okokból bemenetnek szükséges megadni a „nem működő” eredeti erőforrást is, hiszen így csak azok a folyamatok lesznek modellezve, ahol ténylegesen szükséges a tartalék bekapcsolása.

Amennyiben adottak a különböző át-, illetve bekapcsolási stratégiák, a P-gráf és a ráépülő, hatékony keresési módszerek alkalmasak arra, hogy úgynevezett megoldásstruktúrákat állítsanak elő, amelyek leírják a legjobb gyártási receptúrát, esetünkben a leghatékonyabb rekonfigurációs javaslatot adja vissza.



19. ábra: Infrastruktúraelem javításának P-gráf modellje

Forrás: a szerző saját szerkesztése [35] alapján

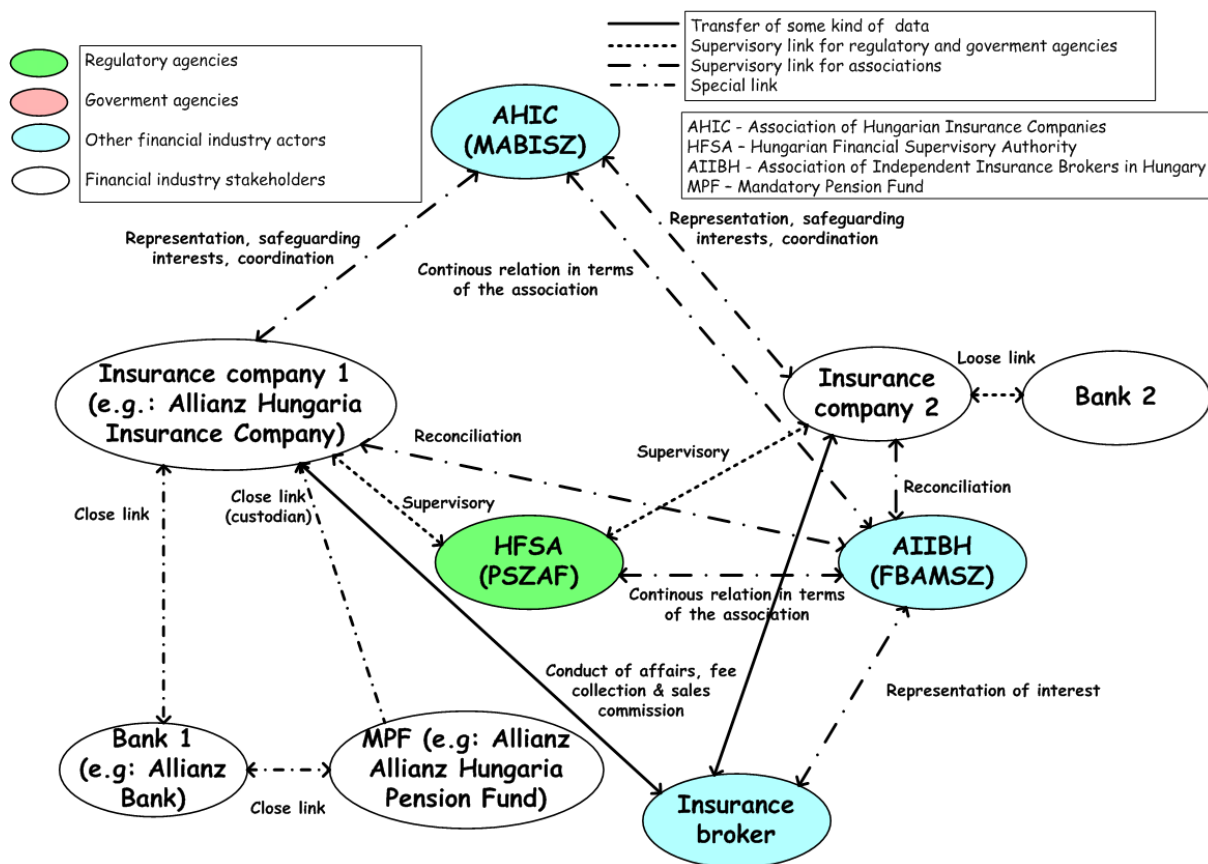
A fenti ábra P-gráf alapon ábrázolja egy szerver mint infrastruktúra-elem lehetséges használati modelljét, ahol azt feltételezzük, hogy javítható komponensről van szó, követve az előzetesen definiált állapot alapú viselkedést. Az érdeklődő olvasónak a teljes esettanulmány P-gráf modelljéhez ajánljuk a [33] cikket.

A módszer előnye, hogy automatizáltan állítja elő a matematikai modelleket, tehát a P-gráf előáll a mérnöki folyamat és erőforrásmodell alapján. Ez jelentős fejlesztési idő-megtakarítást jelent, valamint így nem szükséges a P-gráf-módszertan részleteit ismerni a modell megalkotójának.

További előnye, hogy átlátható módon (a folyamat- és rendszermodellre visszavetítve) adja meg a javasolt, a feltételeket kielégítő optimális megoldást.

8.2. PÉNZÜGYI INFRASTRUKTÚRÁK MAGAS SZINTŰ MODELLEZÉSE

Az alábbiakban a CoMiFin (Communication Middleware for Monitoring Financial Critical Infrastructures), kiváló (excellent) minősítéssel zárult EU kutatási projektben a 2010-es magyar pénzügyi infrastruktúráról készült modellt tekintjük át. Bár azóta több változás történt az infrastruktúrában (ilyen volt például a PSZÁF feladatkörének a Magyar Nemzeti Bankhoz rendelése), a modell alapvetően érvényesen mutatja be a pénzügyi infrastruktúra főbb szereplőit és a köztük fennálló függőségeket.



20. ábra: A pénzügyi kritikus infrastruktúra szereplőinek magas szintű modellje [36]

Mire használható egy ilyen modell?

A modell segítségével vizsgálható az egy-egy változás által közvetlenül és közvetett módon érintett szereplők halmaza. Ilyen lehet például a riportlási kötelezettségek vagy felügyeleti jogkörök változása.

A CoMiFin projekt egyik fő eredménye a kontrollált információ megosztást lehetővé tevő úgynevezett Semantic Room megközelítés volt. Ez lehetővé tette, hogy különböző szereplők (akár egy bank különböző irodái, akár egy ország kereskedelmi bankjai és felügyelete, akár maguk a pénzügyi felügyelet országok közt) megoszassanak egymással alapvetően támadási scenáriókra és a rendszerek igénybevételére utaló információkat. Ezek az információk tipikusan ad hoc módon, adott esetben több nap késleltetéssel terjednek országok közt, pedig a gyors felderítéssel megelőzhetőek lennének a rohamosan terjedő elektronikus támadások, illetve csökkenthető/jobban behatárolható lenne az ezek által okozott kár. (Mindez nem keverendő össze a kibervédelemmel, az internethálózati forgalom biztonságával foglalkozó CERT – Computer Emergency Response Team működésével.)

A modell segítségével tervezhető és ellenőrizhető azon szereplők köre, akik egy információmegosztási körben érintettek, tehát egy Semantic Room tagjai lehetnek. Bár a fenti ábra csak a pénzügyi infrastruktúrára koncentrál, természetesen hasonló felelősségi körök definiálhatók más rendszerekben is (például energiaszolgáltatók, közgazgatás stb.).

A modell alkalmas lehet arra is, hogy „okos” szolgáltatásokat tervezzünk akár a lakosság, akár a pénzügyi szektor szereplői számára. Ennek kiemelt jelentősége lesz 2018-ban, amikor a PSD2 (Revised Payment Service Directive), amely többek közt számlainformációk és fizetési tranzakciók API felületen történő megnyitását írja elő a bankok

száma 2018. január 31-től. Ez egyben lehetőséget jelent arra is, hogy akár a magánszemélyek/cégek kiadásait a fizetendő közterhekkkel vagy a városi szolgáltatásokkal (például BKK-bérlet) összehangoló, figyelemfelkeltő, a pénzügyi tudatosságot növelő alkalmazások szülessenek. A fenti modell alkalmas lehet arra is, hogy ilyen alkalmazások és szolgáltatások potenciális információforrásait és -fogyasztóit azonosítsa.

A fenti modell logikai kapcsolatokat definiál, míg a fizikai összeköttetést a GIRO hálózat biztosítja. Éves forgalma 300 millió tranzakció fölött van, így Magyarország egyik legfontosabb kritikus infrastruktúrájának tekinthető.

Az elmúlt években jelent meg és terjedt el a letagadhatatlanságot biztosító blockchain technológia, amely alkalmas megoldást kínálhat hasonló problémák informatikai megvalósítására.

8.3. PÉLDA MAGYAR INFRASTRUKTÚRA-MODELLRE: EGYSÉGES FORGALMI MODELL

A városi és elővárosi tömegközlekedés forgalmi modelljét alkotta meg a Budapesti Közlekedési Központ (továbbiakban BKK) a KÖZOP-5.5.0-09-11-2011-0025 projektben. A modell önmagán túlmutató jelentőséggel bír a tekintetben, hogy moduláris kiépítése miatt alkalmas lehet más infrastruktúrákkal történő összeköttetések leírására [37].

Az Egységes Forgalmi Modell (EFM) megalkotásának fő motivációja volt, hogy a közlekedési rendszer változásainak hatása felmérhető és tervezhető legyen. Ilyenek lehetnek például új szolgáltatások/infrastruktúra-elemek, a megszokottól eltérő események, megváltozott utazási szokások stb. A modell különösen érdekes példája az összekapcsolt infrastruktúrák (ideértve a városi közlekedést mint kritikus infrastruktúrát) leírásának, amennyiben a teljes közlekedésfejlesztésnek egységes alapot ad, ugyanakkor adatcsereformátumát tekintve felkészült a különböző nyilvántartó rendszerekkel történő együttműködésre.

Az EFM főbb elemei

- A közlekedési **kínálati modell** a jelenleg elérhető vonalakat tartalmazza kapacitásaikkal együtt.
- A közlekedési **igénymodell**, vagyis a háztartások megkérdezésével, a forgalomszámálással, a kamerák és detektorok jelei alapján és egyéb vizsgálatokkal begyűjtött adatok alapján előállított modell arról, milyen célokból és milyen módon közlekednek Budapest lakosai.
- A **területi modell**, vagyis a földrajzi/fizikai sajátosságokat is tartalmazó, a közlekedési infrastruktúra alapjául szolgáló modell.

Az EFM gráf formában tárolva mintegy 10 000 csomópontból és 30 000 szakaszból (élből) áll. Alkalmas „what-if” (mi van, ha?) vizsgálatok támogatására, valamint a multimodális közlekedés átfogó tervezésére.

Kiemelendő, hogy az okosváros-konceptió fontos eleme lehet, amennyiben (igaz, egyelőre periodikus frissítés mellett) számtalan valós mérésből származó adatot tartalmaz. Emellett deklarált célja az üzleti szféra és a városüzemeltetés közti kapcsolat mélyítése, valamint az adat alapú várostervezés és -üzemeltetés támogatása (például akár az energiaárak változásának hatásvizsgálata mentén).

Az okosváros-konceptió fontos eleme lehet az igénymodell pontosítása az alábbiak mentén:

A forgalom keltésének/vonzásának leírása alapvetően azt a modellezési lépést takarja, amikor egy-egy területi egységet közelítünk az onnan induló, illetve oda érkező forgalom meghatározásával. Ilyen lehet például az anonimizált mobilforgalmi adatok hozzáillesztése a modellhez annak érdekében, hogy a forgalom pontosabban becsülhető legyen. Kihasználható lenne továbbá a közösségi alkalmazások (például Waze) historikus adatgyűjtése arról, hogyan alakulnak a forgalmi igények adott időszakokban/területeken. Utóbbi egyben felhasználható lenne az útvonalválasztások/-váltások pontosabb közelítésére is.

9. KRITIKUS INFRASTRUKTÚRA MODELLEK ALKALMAZÁSI LEHETŐSÉGEI

Az alábbiakban áttekintjük az infrastruktúra és szolgáltatásmodellek alkalmazási lehetőségeit.

9.1. KOCKÁZATMODELLEZÉS ÉS KATASZTRÓFAVÉDELEM

A kockázatmodellezés szakirodalmának részletes ismertetése nem képezi jelen monográfia tárgyát, azonban érdemes kiemelni, hogy a kockázati modellek főbb paramétereinek meghatározása nagyban segíthető mérnöki modellek alkalmazásával.

Az infrastruktúrák kockázatmodellezésének sikeressége alapvetően függ attól, hogy a hatás erőssége és az egyes események hatásának továbbterjedése mennyire hihetően és szisztematikusan van modellezve.

9.2. INFRASTRUKTÚRA-FEJLESZTÉSEK TÁMOGATÁSA SZIMULÁCIÓS HATÁSVIZSGÁLATOKKAL

Az infrastruktúra-fejlesztések tervezésének egyik kulcskérdése az alternatívák kiértékelése. Ennek eszközéül szolgál a szimuláció, amely alkalmas lehet egy lokális változás hatása továbbterjedésének, illetve a rendszer külső körülményeiben bekövetkezett változás hatásának lekövetésére.

Speciális esete a hatásvizsgálatnak a hibaelemzés (például Error Propagation Analysis, Failure Modes Effects and Cause Analysis stb.), amikor kifejezetten a rendszerben történő hibaterjedést vizsgáljuk terjedési szabályok, bemenetek és a rendszer egyes elemeinek az állapotának figyelembevételével.

9.3. ADAT- ÉS SZOLGÁLTATÁSKATALÓGUS LÉTREHOZÁSA

Ahogy a bemutatott külföldi példák is szemléltetik, a sikeres okosváros-szolgáltatások fontos feltétele a nyilvánosan elérhető, egységes leírómodell (metaadatokat) használó adat- és szolgáltatás/alkalmazáskatalógus létrehozása. Ehhez legalább taxonómia jelleggel ajánlott egy kategóriamodell definiálása.

9.4. INTEGRÁCIÓ TERVEZÉSE

A modellek alkalmasak lehetnek arra, hogy a már bemutatott függőségek figyelembevételével infrastruktúra-szolgáltatások integrációját megtervezzük.

Amennyiben a szolgáltatások minőségi paramétereit (elvárt/nyújtott paramétereket) leírjuk, a kompozit szolgáltatásokra minőségi értékek is becsülhetők (lásd például [38]), így az okos város összetett folyamatainak teljesítményére, szolgáltatásbiztonságára és teljesítőképességére adatokkal és számításokkal alátámasztott becslés adható.

IRODALOMJEGYZÉK

- [1.] RINALDI, Steven M. – PEERENBOOM, James P. – KELLY Terrence K. (2001): Identifying, understanding, and analyzing critical infrastructure interdependencies. *IEEE Control Systems*, 21(6), 11–25.
- [2.] PEDERSON, Peter – DUDENHOEFFER, Danile – HARTLEY, Steven – PERMANN, May (2006): Critical infrastructure interdependency modeling: a survey of US and international research. *Idaho National Laboratory*, 25:27.
- [3.] MUHI Lajos (2007): *A Magyar Köztársaság kritikus infrastruktúráinak védelme*. PhD-értekezés, Zrínyi Miklós Nemzetvédelmi Egyetem.
- [4.] SETOLA, Roberto – ROSATO, Vittorio – KYRIAKIDES, Elias – ROME, Erich (2017): *Managing the Complexity of Critical Infrastructures: A Modelling and Simulation Approach*. Springer.
- [5.] OUYANG, Min (2014): Review on modeling and simulation of interdependent critical infrastructure systems. *Reliability Engineering & System Safety*, 121: 43–60.
- [6.] HAIMES, Yacov, Y. – JIANG, Pu (2001): Leontief-based model of risk in complex interconnected infrastructures. *Journal of Infrastructure systems* 7, no. 1: 1–12.
- [7.] Dd MIN, Hyeung-Sik J. – BEYELER, Walter – BROWN, Theresa – SON, Young Jun – JONES, Albert T., (2007): Toward modeling and simulation of critical national infrastructure interdependencies. *IIE Transactions* 39, no. 1: 57–71.
- [8.] ALBERT Réka – BARABÁSI Albert László (2002): Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74 (1): 47–97.
- [9.] ERDŐS Pál – RÉNYI Alfréd. (1959). On Random Graphs. I. *Publicationes Mathematicae*, 6: 290–297.
- [10.] CACCIOLI, Fabio – LIVAN, Giacomo – ASTE, Tomaso (2016): Scalability and Egalitarianism in Peer-to-Peer Networks. In *Banking Beyond Banks and Money*, pp. 197–212. Springer International Publishing.
- [11.] ISO/IEC 25010 szabvány. Elérve: 2017. 11. 23. <http://iso25000.com/index.php/en/iso-25000-standards/iso-25010/60-compatibility>
- [12.] KOCSIS Imre – SALÁNKI Ágnes – PATARICZA András (2017): *Measurement-Based Identification of Infrastructures for Trustworthy Cyber-Physical Systems*. The 14th Chapter of Trustworthy Cyber-Physical Systems Engineering, pp. 369–394. CRC Press, Taylor & Francis Group.
- [13.] MOHAMED, Eid – ROSATO, Vittorio (2016): Critical Infrastructure Disruption Scenarios Analyses via Simulation. In *Managing the Complexity of Critical Infrastructures*, pp. 43–61. Springer International Publishing.
- [14.] SMART CITY ARTIFACTS. (Elérve: 2017. 11. 15.) <http://ci.emse.fr/opensensingcity/ns/ontologies/>
- [15.] AUBIGNY, Matthieu – HARPE, Carlo – CASTRUCCI, Marco (2011): Risk Ontology and Service Quality Descriptor Shared among Interdependent Critical Infrastructures. In Xenakis C. – Wolthusen S. eds: *Critical Information Infrastructures Security. CRITIS 2010. Lecture Notes in Computer Science*, vol. 6712. Springer, Berlin, Heidelberg.
- [16.] DAVID, István – GÖNCZY László (2014): *Ontology-Supported Design of Domain-Specific Languages: A Complex Event Processing Case Study*. The 6th Chapter of *Advances and Applications in Model-Driven Engineering* pp. 106–133. DOI: 10.4018/978-1-4666-4494-6.ch006. IGI Global, 2014.
- [17.] KELLER, Alexander – HEIKO, Ludwig (2003): The WSLA framework: Specifying and monitoring service level agreements for web services. *Journal of Network and Systems Management*, 11, no. 1, 57–81.

- [18.] Web Services Reliable Messaging. OASIS consortium, 2009. (Elérve: 2017. 11. 21.). <http://docs.oasis-open.org/ws-rx/wsrml/200702>
- [19.] Web Services Security szabványok, OASIS consortium (2007). (Elérve: 2017. 11. 21.) https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss
- [20.] STEPHEN, Gilmore – GÖNCZY László – KOCH Nóra – MAYER, Philip – TRIBASTONE, Mirco – VARRÓ Dániel (2011): Non-functional properties in the model-driven development of service-oriented systems. *Software and Systems Modeling* 10, no. 3, 287–311.
- [21.] Business Process Modelling and Notation (BPMN). Object Management Group. 2013. (Elérve: 2017. 11. 11.) <http://schema.omg.org/spec/BPMN/index.htm>
- [22.] RICCARDO, Cognini – CORRADINI, Flavio – POLINI, Andrea – RE, Barbara (2014): Modelling process intensive scenarios for the smart city. In *International Conference on Electronic Government*, pp. 147–158. Springer, Berlin, Heidelberg.
- [23.] Business Process Incubator. (Elérve: 2017. 11. 11.) <https://www.businessprocessincubator.com/>
- [24.] Web Services Business Process Execution Language (WS-BPEL), V2.0. OASIS Standard. (Elérve: 2017. 11. 02.) <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html>
- [25.] NANDI, Prabir et al. Data4BPM, Part 1: *Introducing Business Entities and the Business Entity Definition Language (BEDL)*. A first class representation of data for BPM applications. IBM Whitepaper, 2010.
- [26.] Decision Model and Notation (OMG szabvány). Elérve 2017. 11. 11. <https://www.omg.org/dmn/>
- [27.] The Rule Markup Initiative, 2013. (Elérve: 2017. 11. 10.) <http://ruleml.org/index.htm>
- [28.] The Open Group Service Integration Maturity Model (OSIMM) Version 2 – The Model, version 2. Open Group Consortium, 2016. (Elérve: 2017. 11. 10.) <http://www.opengroup.org/soa/source-book/osimmv2/p2.htm>
- [29.] ANTAL Péter – ANTOS András – HORVATH Gábor – HULLAM Gábor – KOCSIS Imre – MARX Péter – MILLINGHOFFER András – PATARICZA András – SALÁNKI, Ágnes (2014): *Intelligens adatelemzés*. Typotex.
- [30.] INSELBERG, Alfred – DIMSDALE, Bernard (1987): Parallel coordinates for visualizing multi-dimensional geometry. In *Computer Graphics*, pp. 25–44. Springer, Tokyo.
- [31.] KOCSIS Imre (2020). *IT rendszerek és kritikus szolgáltatások teljesítménye*. Kismonográfia. Okosváros-technológiák 16. kötet. NKE-KTI, Budapest.
- [32.] RAUNIO, Mika – NORDLING, Nadja – KETOLA, Taina – SAARINEN, Jukka P. – HEINIKANGAS, Annina (2016): *The Open Innovation Platforms Handbook*. 2016. (Elérve: 2017. 11. 28.) https://avoimetinnovaatioalustat.files.wordpress.com/2016/02/kc3a4sikirja_eng.pdf
- [33.] VARRÓ-GYAPAY Szilvia – MAGYAR Dániel László – KOCSIS-MAGYAR Melinda – TASI Katalin – HOANGTHANH DINH Attila – BAUSZ Ágota – GÖNCZY László (2015): *Optimization of reconfiguration mechanisms in Critical Infrastructures*. In Proceedings of the 2nd International Workshop on Reliability and Security Aspects for Critical Infrastructures (ReSA4CI), Delft, The Netherlands, September 22, 2015.
- [34.] KLEMES, Jiri – FRIEDLER FERENC – BULATOV, IGOR – VARBANOV, PETAR (2010): *Sustainability in the Process Industry: Integration and Optimization*. *Green Manufacturing & Systems Engineering*. McGrawHill.
- [35.] BERTÓK Botond – KOVÁCS Zoltán (2011): *Gyártórendszerek modellezése*. Pannon Egyetem, Typotex.
- [36.] Deliverable D2.1 *Identification and classification of vulnerabilities of interconnected infrastructures and solutions*. State of the art. Public Deliverable of COMIFIN FP7 (225407). 2009.

- [37.] MÁTRAI Tamás – ÁBEL Melinda – KERÉNYI László Sándor (2015): *How can a transport model be integrated to the strategic transport planning approach: A case study from Budapest*. IEEE 2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS).
- [38.] <http://www.citysdk.eu/citysdk-toolkit/components-of-the-toolkit/discovery-service/> CitySDK Discovery Service.
- [39.] GÖNCZY László – CHIARADONNA, Silvano – DI GIANDOMENICO, Felicita – PATARICZA András – BONDAVALLI, Andrea – BARTHA Tamás (2006): Dependability evaluation of web service-based processes. In *European Performance Engineering Workshop*, pp. 166–180. Springer, Berlin, Heidelberg.

A Nemzeti Közszerológáti Egyetem kiadványa



Kiadó:

Nemzeti Közszerológáti Egyetem;
Közgazgatási Továbbképzési Intézet

www.uni-nke.hu

Felelős kiadó:

Prof. Dr. Kis Norbert rektorhelyettes

Címe: 1083 Budapest, Üllői út 82.

Olvasószerkesztő:

Dorogi Katalin

Tördelőszerkesztő:

Vöröss Ferenc

ISBN 978-963-498-395-8 (PDF)