

# FOLYAMATLEÍRÓ MODELLEK ÉS AUTOMATIKUS FELTÁRÁSUK

BAKSÁNÉ VARGA ERIKA

*Egy vállalat működését az üzleti folyamatai határozzák meg. Ezek a munkafolyamatok olyan feladatok sorozatából állnak, amelyeket az alkalmazottak rendszeresen végeznek konkrét célok elérése érdekében. Ezeknek a folyamatoknak a formális leírása kulcsfontosságú tényező a vállalat működésének hatékonyabbá tétele szempontjából. A számítógépes munkavégzés során elvégzett tevékenységeket (bekövetkezett eseményeket) eseménynaplók tárolják. Mivel az eseménynaplót a rendszer automatikusan rögzíti, első körben feltételezzük, hogy valós adatokat tartalmaz. Jelen kutatás célja vizsgálni azokat a módszereket, amelyekkel a napló adataiból kikövetkeztethető az az általános folyamatmodell, amelyre a napló összes eseményeseménye illeszkedik. A vizsgálatot azért végezzük, hogy kiválasszunk benchmark eljárásokat a saját módszerünk értékeléséhez.*

## 1. Bevezetés

A kutatás első lépéseként áttekintem a leggyakrabban alkalmazott folyamatleíró nyelveket és azokat az algoritmusokat, amelyek a számítógépes munkavégzés során rögzített eseménynaplókból ilyen modelleket képesek automatikusan előállítani.

A grafikus üzleti folyamatmodellekben (Business Process Modeling, BPM) az a közös, hogy a folyamatokat tevékenységek sorozataként írják le, ahol a tevékenységek sorrendje egymásra épülést, időbeli függőséget jelöl. Ezt egy gráfban ábrázolják, amely csomópontokból és a csomópontok közötti irányított élekből épül fel, jelezve a folyamatok kronológiai sorrendjét. Csoportosításuk szerint vannak

- hagyományos folyamatmodellező nyelvek (pl. Petri háló, IDEF, EPC, RAD),
- munkafolyamat-modellező nyelvek (pl. WPD, BPMN), és
- objektumorientált nyelvek (pl. UML).

A legrégebbi és legelterjedtebb modell a Petri-háló, amelynek létezik néhány magasabb szintű kiterjesztése, míg a legkifejezőbb nyelv a BPMN. Ezekhez a modellekhez szabványos XML-alapú adatcsere-formátumokat is kidolgoztak. A BPMN-szabvány magában foglalja az XPDL nyelvet (XML Process Definition Language), a Petri hálók pedig automatikusan feldolgozhatók a PNML nyelv (Petri Net Markup Language) segítségével.

A Workflow Patterns Initiative szisztematikus elemzést végzett, hogy milyen jellemző szerkezeteket használnak a BPM-nyelvek. Ennek eredményeként létrehoztak egy sablon gyűjteményt, és kidolgozták a YAWL nyelvet, amellyel minden azonosított minta leírható. Ezek a minták a munkafolyamatot többféle nézőpontból írják le, például vannak vezérlési minták, adatminták vagy erőforrásminták.

Vizsgálataink során mi a vezérlő szerkezetekre összpontosítunk. Ebben a csoportban 43 mintázat található 8 osztályba sorolva, melyek közül a legfontosabbak:

- szekvenciális végrehajtása a tevékenységeknek,
- XOR, azaz kizárólagos választás több tevékenység közül, amelynek be kell fejeződnie a következő tevékenység indítása előtt,
- OR, azaz egy vagy több tevékenység kiválasztása az opciók közül. Az összes kiválasztott tevékenységnek be kell fejeződnie ahhoz, hogy a következő tevékenység indulhasson.
- AND, azaz az összes tevékenység párhuzamos indítása, majd ezek szinkronizálása.

A fent említett modellek speciális jelölést alkalmaznak a vezérlés menetét befolyásoló elemekre. Ezek az elemek összekapcsolják a tevékenységeket a grafikus folyamatmodellgráfban, de az eseménynaplóban – ami a folyamatmodell egy konkrét, megvalósult példánya – nincs jelük. Ezért ezek feltárása külön módszereket igényel.

## 2. Kutatási eredmények összesítése

Az egyik legszélesebb körben tanulmányozott folyamatbányászati művelet az automatikus folyamatfeltárás. A folyamatfeltáró módszerek bemenete egy eseménynapló, kimenetként pedig egy üzleti folyamatmodellt állítanak elő, ami az eseménynaplóban rögzített vagy az alapján feltételezett tevékenységek közötti kapcsolatokat ábrázolja a vezérlésfolyam szempontjából. Az eljárás során előállított modellnek az alábbi elvárásokat kell teljesítenie:

- az eseménynaplóban található összes esetet lefedi (maximálisan illeszkedik a bemenetre),
- nem generál olyan eseteket, amelyekhez hasonló az eseménynapló nem tartalmaz (pontos),
- minden olyan eset előállítható belőle, ami az eseménynaplóban található esetekhez hasonló (általános), és
- a lehető legegyszerűbb (a komplexitása minimális).

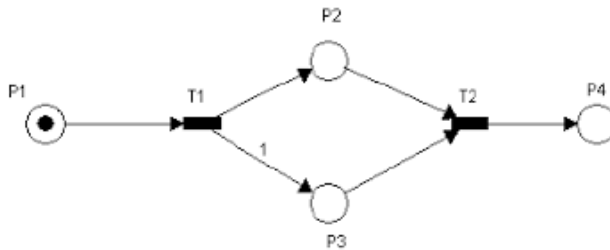
Vizsgált folyamatmodellező nyelvek:

Petri-háló. A legelterjedtebb folyamatmodellezési eszköz. A modellben a csomópontoknak két típusa van:

- normálállapot-hely (place),
- állapotátmenet (transition).

Kapcsolat csak normál állapot és állapot átmenet csomópontokat köthet össze. Az átmeneteknél így beszélhetünk bemenő és kimenő helyekről.

A normál állapotok és átmenetek mellett státuszjelölő elemek (tokenek) is vannak a modellben. A tokenek a létezésükkel jelzik a kapcsolódó állapot teljesülését. A modellben sajátos feltétel kell az átmenet megvalósulásához. Egy átmenet csak akkor történik meg (tüzelés), ha minden bemenő helyen van token. Az átmenet megvalósulása során a tokenek átkerülnek a kimenő helyekre. Itt nem feltétlenül érvényesül a tokenmegmaradás törvénye. Az élekhez rendelhetünk kapacitásértéket is, mely megadja az átmenet során érintett tokenek darabszámát.



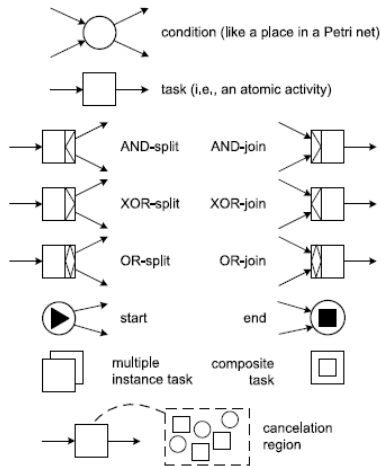
Az alap Petri-hálóhoz többféle továbbfejlesztést dolgoztak ki, amelyek további szempontokkal bővítik ki az alapmodellt. A fontosabb bővítési irányok:

- színezett Petri-háló, amikor a tokenek további jellemzőket is tartalmazhatnak,
- hierarchikus Petri-háló,
- időbeliséget, ütemezést tartalmazó Petri-háló.

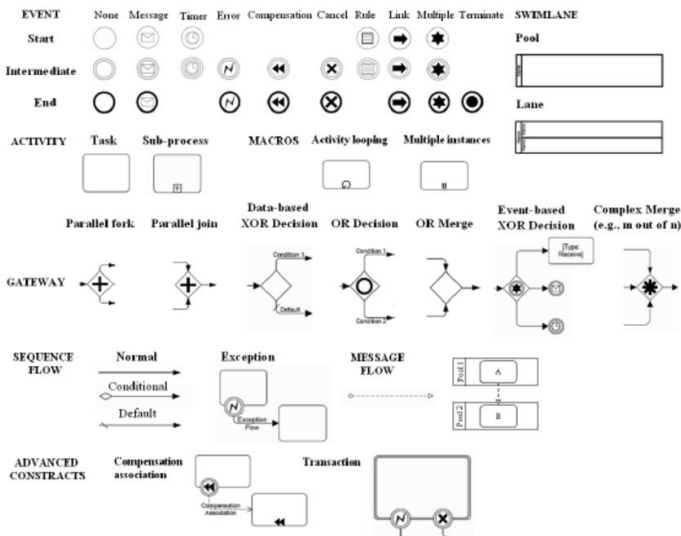
A Petri-háló speciális esete a Workflow háló, amelyben az alábbi két megkötés él:

- Egyetlen induló (start) és egyetlen cél (záró) csomópont létezik.
- Minden csomópont rajta van egy start-cél útvonalon.

YAWL (Yet Another Workflow Language). Workflow szemléletű modellezési eszköz. A nyelv több tipikus workflow mintát biztosít építőköként. A minták több különböző szempontra is kiterjednek, így vannak többek között vezérlésiem-minták, adatstruktúra-minták, műveleti minták és hibajelenség-minták. A nyelv hatékonyságát ezen minták biztosítják, hiszen a mintákat felhasználva gyorsabban lehet komplex rendszereket felépíteni. A modell vezérlő elemei az alábbi ábrán láthatók.

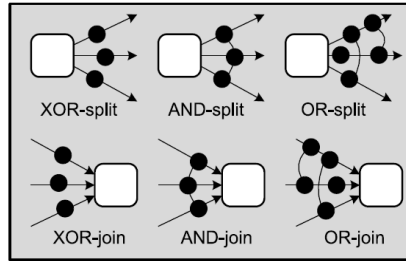


BPMN (Business Process Model Notation). A YAWL formalizmushoz hasonló modellezési nyelv, mely az iparban igen széles körben elterjedt, elsősorban a funkcionális gazdagsága miatt. Jelölésrendszerét az alábbi ábra szemlélteti.

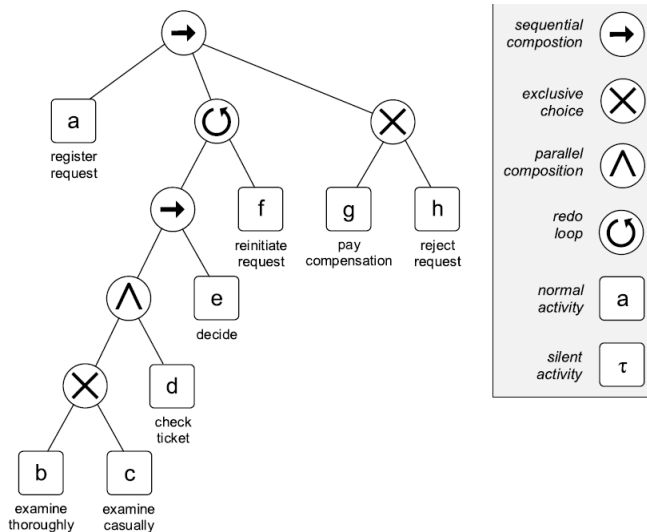


Causal Nets. Ez a modell a folyamatot olyan gráfként reprezentálja, ahol a csomópontok az eseménynaplóban is megjelenő tevékenységek és az élek a tevékenységek

közötti okozati összefüggést képviselik. A modellben a tevékenységekhez hozzárendelik a bejövő függőségek és a kimenő függőségek halmazát. Az okozati kapcsolatoknál azonban összetett összefüggések is megjelenhetnek, ezért a modellben új kapcsolati formák is megjelennek, melyek a szokásos konjunkció és diszjunkció operátorokon alapulnak.



Folyamatfa. A folyamatok leírásánál az általános gráfalapú módszerek nem feltétlenül adnak érvényes leírást, nem tudják igazán kontrollálni a magasabb szinten megjelenő formákat. A folyamatfa-alapú leírás ezzel szemben egy strukturális, tartalmazásalapú hierarchikusmodell-leírás, ami egy kevésbé rugalmas, de ellenőrzöttebb modellt eredményez. A fában a belső csomópontok strukturális egységek, míg a levelek elemi műveleteket jelölnek. Az alábbi ábrán látható folyamatfa Wil van der Aalst professzor *Process Mining* című könyvében található.



### 2.1. Elvégzett kísérletek bemutatása

A vizsgálat alá vont eljárások kiválasztásának szempontjai:

- 1) 10 évnél nem régebbi módszer,
- 2) valós eseménynaplón tesztelt eljárás,
- 3) a feltárt folyamatmodell procedurális típusú, azaz az események végrehajtási sorrendjét (control flow) határozza meg.

Ez alapján 23 folyamatfeltáró eljárást vizsgáltam meg 2 szempont szerint:

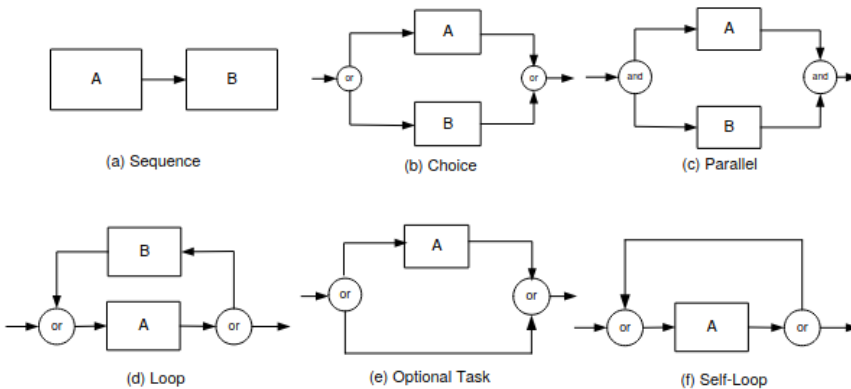
- 1) a feltárt modell leíró nyelve (pl. BPMN, Petri-háló stb.), és
- 2) a feltárt vezérlőelemek (AND, OR, XOR).

### 2.2. Kiértékeléshez használt referencia folyamatok és értékek bemutatása

Az eseménynaplókból procedurális folyamatmodell feltáró létező eljárások:

HK, Inductive Miner, Process Skeletonization, Evolutionary Tree Miner, Aim, Competition Miner, Directed Acyclic Graphs, CN Mining, alpha\$, Maximal Pattern Mining, DGEM, RegPFA, BPMN Miner, CSM Miner, TAU Miner, PGminer, ProM-D, Proximity Miner, Heuristics Miner, Split miner, Fodina, Stage miner, Decomposed Process Miner.

Ezeknek az eljárásoknak a vizsgálata során azt tartottam szem előtt, hogy a leggyakoribb folyamatvezérlési minták közül, melyeket képesek feltárni:



### 2.3. Kiértékelések eredményeinek bemutatása

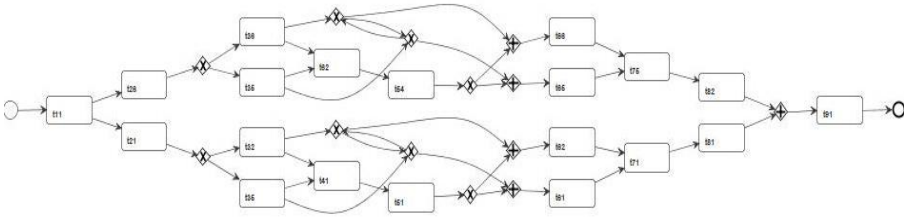
A vizsgálati eredményeket az alábbi táblázatban foglalom össze.

Eljárás neve	A feltárt modell leíró nyelve	Feltárt vezérlőelemek			
		AND	XOR	OR	Ciklus
Aim	Petri-háló	+	+		+
alpha\$	Petri-háló	+	+		+
<b>BPMN Miner</b>	<b>BPMN</b>	+	+	+	+
CN Mining	Causal net (ok-okozati háló)	+	+		+
Competition Miner	BPMN	+	+		+
CSM Miner	Állapotgép	+	+		+
<b>Decomposed Process Miner</b>	<b>Petri-háló</b>	+	+	+	+
DGEM	BPMN	+	+		+
Directed Acyclic Graphs	Írányított gráf		+		
<b>Evolutionary Tree Miner</b>	<b>Folyamatleíró fa</b>	+	+	+	+
Fodina	BPMN	+	+		+
Heuristics Miner	BPMN	+	+		+
HK	Petri-háló	+	+		+
<b>Inductive Miner</b>	<b>Folyamatleíró fa</b>	+	+	+	+
Maximal Pattern Mining	Causal net (ok-okozati háló)	+	+		+
PG miner	Részben rendezett gráf	+	+		
Process Skeletonization	Írányított gráf		+		+
ProM-D	Petri-háló	+	+		+
Proximity Miner	Causal net (ok-okozati háló)	+	+		+
RegPFA	Petri-háló	+	+		+
Split miner	BPMN	+	+		+
Stage miner	Causal net (ok-okozati háló)	+	+		+
TAU Miner	Petri-háló	+	+		+

## 2.4. Eredményeket szemléltető képernyőképek

A vizsgált folyamatmodell-leíró formalizmusok grafikus nyelvek. A folyamatfeltáró algoritmusok ezek közül csak a folyamatleíró fákat és a gráfokat képesek előállítani. A BPMN és Petri-háló modellek csak konverzió után vizualizálhatók. Ezek esetében a modellt feltáró eljárások kimenete egy számítógéppel olvasható, XML-alapú formátum. A BPMN esetén egy XPD- (XML-based Process Definition Language), Petri-háló esetén pedig egy PNML- (Petri Net Markup Language) fájl.

A PNML folyamatmodellfájlokat a ProM szoftverrel lehet megjeleníteni ([www.promtools.org](http://www.promtools.org)). A PNML egy XML-alapú, számítógéppel olvasható (machine readable) formátum. A ProM szoftverben BPMN grafikus formátumra konvertáltam, majd exportáltam jpg, png, pdf, eps stb. formátumba és az alábbi eredményt kaptam:



A BPMN-gráfban a téglalapok az események (tevékenységek) és a rombuszszal jelölt elemek az átjárók (gateway). Az x-szel jelölt átjáró azt jelenti, hogy az adott ponton szétválik a folyamat vezérlése egymást kizáró ágakra (exclusive gateway), azaz a folyamatpéldányokban csak az egyik útvonal valósul meg az ebből a pontból kiinduló ágak közül (OR-split). A döntés egy megadott feltétel kiértékelésétől függ. A +-szal jelölt átjáró az a pont, ahonnan akkor lehet továbblépni, ha a megelőző ágak közül az aktív befejeződött (OR-join). Ezek az átjárók a modell feltárása során kerülnek meghatározásra, a naplóállományokban nincs nyomuk.

## 3. Összegzés

A vizsgálat során áttekintettem a legelterjedtebb folyamatmodellező nyelveket, és az automatikus feltárásukat megvalósító módszereket azzal a céllal, hogy kiválasszam azokat az eljárásokat, amelyekkel össze fogjuk hasonlítani a saját algoritmusunk teljesítményét.

A vizsgálat eredményeit összegző táblázatból látható, hogy első körben 4 benchmark módszert választottam ki: a BPMN Miner, a Decomposed Process



Miner, az Evolutionary Tree Miner és az Inductive Miner eljárásokat, mert ezek adják vissza legpontosabban a folyamat vezérlési menetét (control-flow). Mivel azonban az OR-joint is tartalmazó BPMN modellek és Petri-hálók esetén a pontosságot és az illeszkedés mértékét nem tudjuk mérni (nem találtam hozzá módszert), ezért a folyamatleíró fát visszaadót eljárásokat fogjuk referenciaként használni: az Evolutionary Tree Minert és az Inductive Minert.