

# BÖRCS ATTILA

---

## Komponensek kompatibilitásának modellezése és verifikációja

Feldolgozott cikkek:

**Modelling and Verification of Compatibility of Component Composition**  
by Donald C. Craig , Wlodek M. Zuberek

**A generic framework for n-protocol compatibility checking** by Gwen  
Salaün, Meriem Ouederni, Francisco Durán

# Tartalom

- Bevezetés, probléma felvetés
- Komponensek modellezése Petri hálókkal
- Komponensek összecsatolása és kompatibilitása
- Alkalmazási lehetőségek
- Probléma kiterjesztés: Egy generikus megoldás n-protokoll kompatibilitás vizsgálatra

# Tartalom

- Bevezetés, probléma felvetés
- Komponensek modellezése Petri hálókkal
- Komponensek összecsatolása és kompatibilitása
- Alkalmazási lehetőségek
- Probléma kiterjesztés: Egy generikus megoldás n-protokoll kompatibilitás vizsgálatra

# Bevezetés és probléma felvetés

- Komplex szoftver rendszerek modul kompatibilitás vizsgálata nehézkes
- Nehézségek:
  - Előgyártott szoftver elemek, szubrutinok
  - Különböző interfészek összehangolása nehézkes
- Lehetséges megoldás:
  - Statikus kompatibilitás vizsgálat (függvény visszatérési érték, fordítási idő, függvény argumentumok típusai...)
  - Hátrány: Nem hatékony dinamikus és viselkedési kompatibilitás vizsgálatára két szoftver modul között

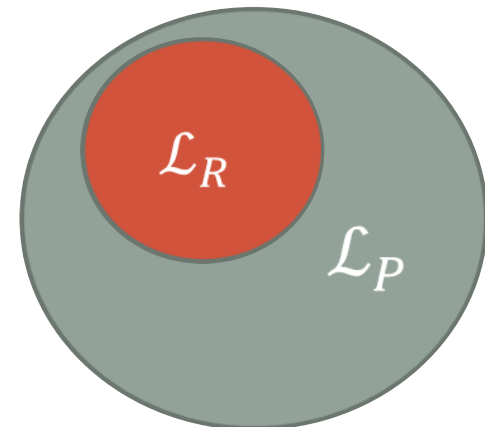
# Tartalom

- Bevezetés, probléma felvetés
- **Komponensek modellezése Petri hálókkal**
- Komponensek összecsatolása és kompatibilitása
- Alkalmazási lehetőségek
- Probléma kiterjesztés: Egy generikus megoldás n-protokoll kompatibilitás vizsgálatra

# Komponensek modellezése Petri hálókkal

## Kontribúció

- Komponens kölcsönhatások modellezése
- Komponens interfészek megvalósítása Petri hálókkal
- Interfész kompatibilitásának vizsgálata:
  - Összekötött, deadlock mentes
  - Jelölje  $\mathcal{L}_R$  a kérések sorozatainak egy halmazát
  - Jelölje  $\mathcal{L}_P$  a szolgáltatások sorozatainak halmazát
  - $\mathcal{L}_R \subseteq \mathcal{L}_P$  esetén a két komponens kompatibilis



# Komponensek modellezése Petri hálókkal

- **Komponens:** logikai egység, jól definiált interfészekkel.
  - Szolgáltatásokat ad a környezetének
  - Szolgáltatásokat fogad a környezetétől
  - ☹ Belső működést nem modellezzük
  - ☺ Komponensek egy rendszerét modellezzük
- **Interfész modell**
  - Címkézett Petri háló:  $\mathcal{M}_i = (P_i, T_i, S_i, A_i, \ell_i, m_i)$ , ahol
    - $P_i$ : Helyek, vagy állapotok
    - $T_i$ : Átmenetek
    - $A_i \subseteq P_i \times T_i \cup T_i \times P_i$ : Lehetséges élek, pályák
    - $S_i$ : szolgáltatások halmaza
    - $\ell_i: T_i \rightarrow S_i \cup \{\varepsilon\}$ : címkézett átmenetek
    - $m_i: P_i \rightarrow \{0,1,2 \dots\}$ : kezdeti tüzelő függvény

# Komponensek modellezése Petri hálókkal

- Két fajta interfész: provider (p) szolgáltató, requester (r) kérő
- Interfészek kétértelműségének elkerülése:
  - p-interfészek összes szolgáltatásának egy címkézett reprezentációja létezhet:

$$\forall t_i, t_j \in T : \ell(t_i) = \ell(t_j) \rightarrow t_i = t_j$$

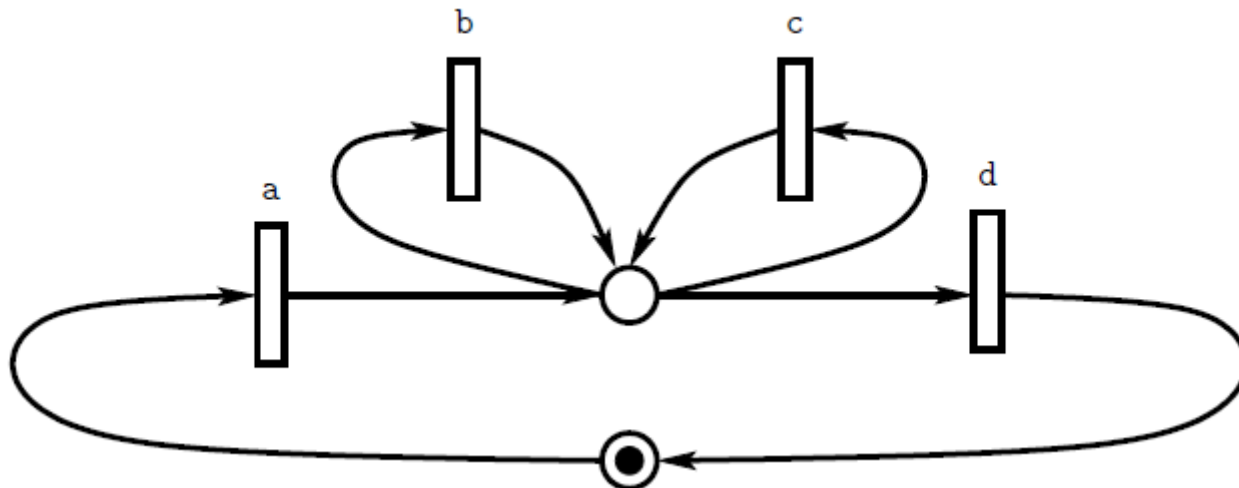
- **Interfész leíró nyelv:**

- Célja:
  - Interfészek közötti argumentum továbbítás
  - Tartalom független (egyszerű adatstruktúrák, stack...)
- Legyen  $\delta = t_{i_1} t_{i_2} \dots t_{i_k}$  egy kezdeti tüzelési szekvencia
- Legyen  $\mathcal{F}(\mathcal{M}_i)$  kezdeti tüzelési szekvenciák halmaza



# Komponensek modellezése Petri hálókkal

- $\mathcal{M}_i$  leíró nyelve:  $\mathcal{L}(\mathcal{M}_i)$  az összes string halmaza  $S_i$  szolgáltatásokon, ami
  - Elérhető egy teljes kezedi tüzelési szekvencia címkézésével  
 $\mathcal{L}(\mathcal{M}_i) = \{\ell(\delta) \mid \delta \in \mathcal{F}(\mathcal{M}_i) \wedge \ell(\delta) \text{ műv. teljes sorozata}\}$
- Műveletek teljes sorozata példa:  $(a(b|c)^*d)^*$

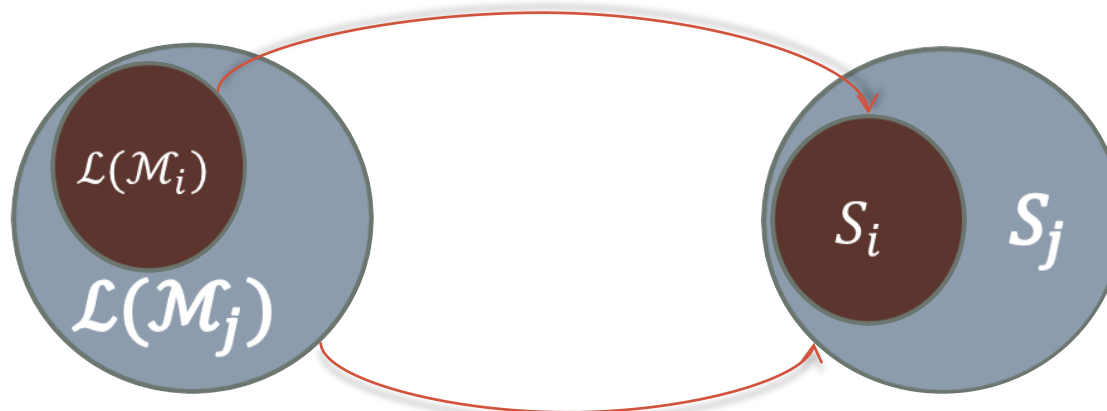


# Tartalom

- Bevezetés, probléma felvetés
- Komponensek modellezése Petri hálókkal
- **Komponensek összecsatolása és kompatibilitása**
- Alkalmazási lehetőségek
- Probléma kiterjesztés: Egy generikus megoldás n-protokoll kompatibilitás vizsgálatra

# Komponensek összecsatolása és kompatibilitása

- Két komponens kompatibilis ha interfészen keresztül,
  - p-komponens által kínált összes szolgáltatás elérhető a r-komponens által
  - r-komponens olyan kéréseket eszközöl, amit a p-komponens el tud látni
- **Kompatibilis interfész:**
- $\mathcal{M}_i$  r-interfész modell és  $\mathcal{M}_j$  p-interfész modell kompatibilis  $\Leftrightarrow \mathcal{L}(\mathcal{M}_i) \subseteq \mathcal{L}(\mathcal{M}_j)$  és  $S_i \subseteq S_j$



# Komponensek összecsatolása és kompatibilitása

## • Interfészek összekapcsolása

- r-interfész modell  $\mathcal{M}_i$  (kérő) és p-interfész modell (szolgáltató)  $\mathcal{M}_j$  összevonása egy közös  $\mathcal{M}_{ij} = (P_{ij}, T_{ij}, S_{ij}, A_{ij}, \ell_{ij}, m_{ij})$  Petri hálóvá:  $\mathcal{M}_i \triangleright \mathcal{M}_j$
- Követelmény 1): Helyek és átmentek diszjunkt halmazt alkossanak:  $P_i \cap P_j = T_i \cap T_j = \emptyset$
- Érvényes az összekapcsolás ha r-interfész és p-interfész között átmentnek nincs üres címkéje:

$$\hat{T}_i = \{t \in T_i : \ell_i(t) \neq \varepsilon\}$$

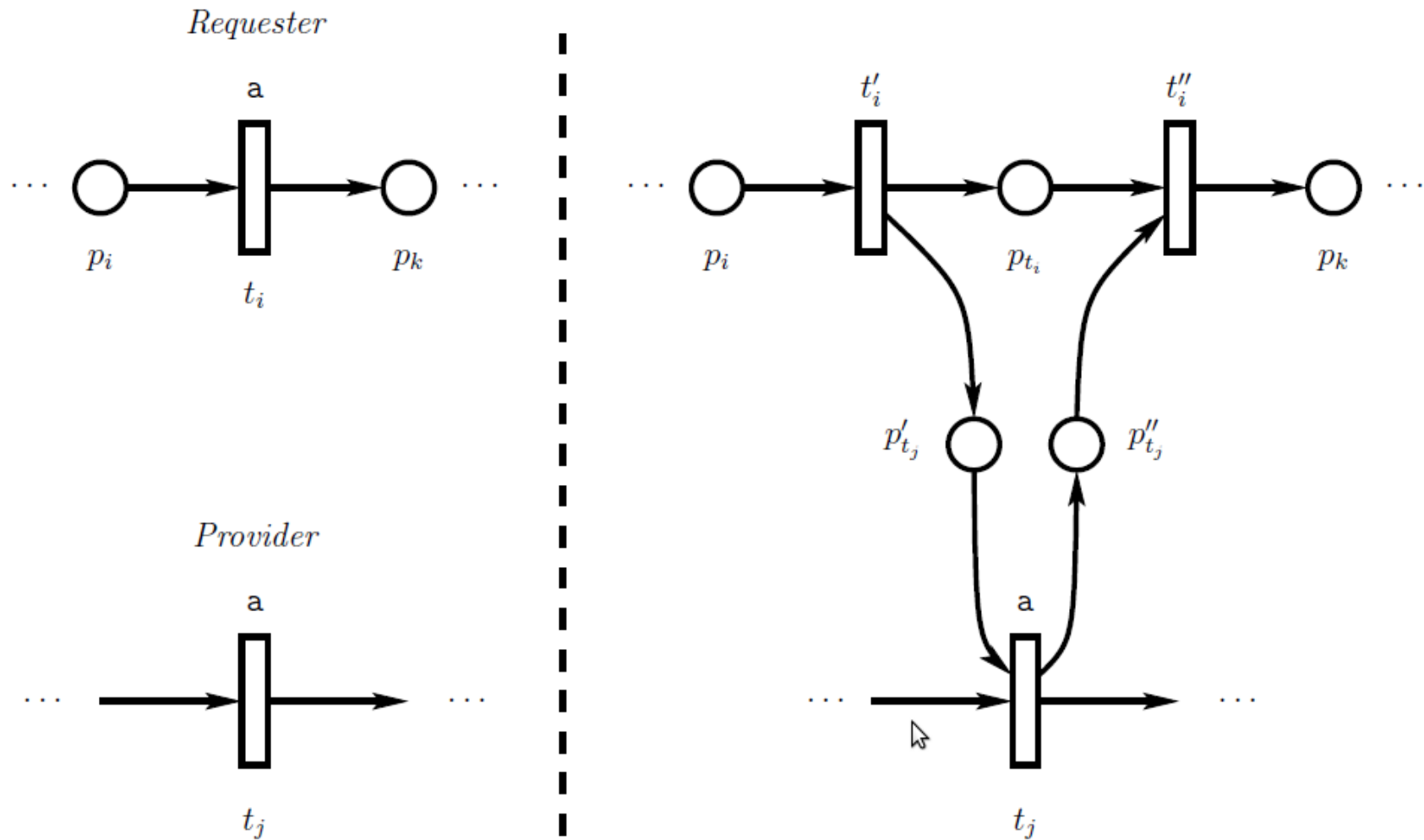
$$\hat{T}_j = \{t \in T_j : \ell_j(t) \neq \varepsilon\}$$

- Követelmény 2): Szinkronizációs pont:

$$P_{ij} = P_i \cup P_j \cup \{P_{t_i} : t_i \in \hat{T}_i\} \cup \{P'_{t_j}, P''_{t_j} : t_j \in \hat{T}_j\}$$

Megakadályozza további helyek hozzáadását, ha több kérés érkezik egy szolgáltató interfészhez

# Komponensek összecsatolása és kompatibilitása

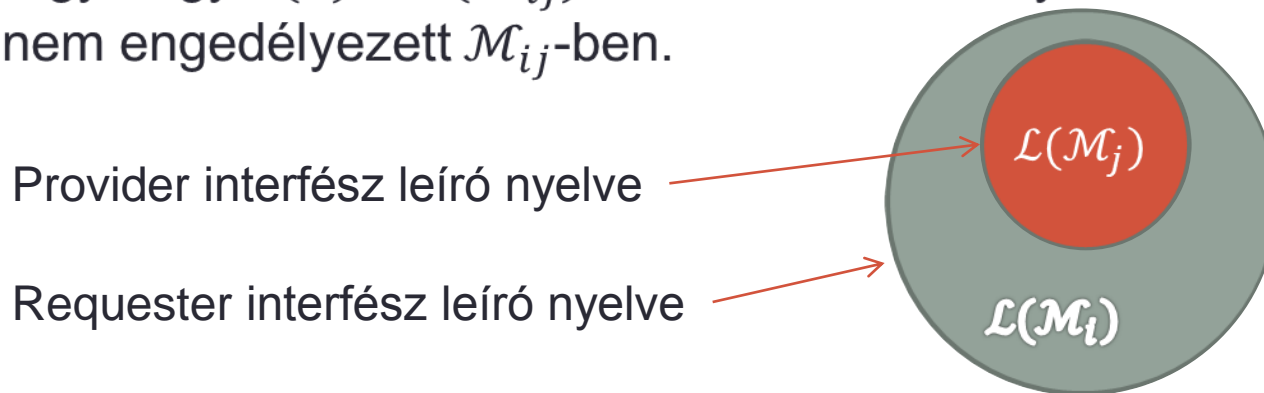


# Komponensek összecsatolása és kompatibilitása

- Követemény 3): Deadlock mentesség
  - Állítás: r-interfész modell  $\mathcal{M}_i$ (kérő) és p-interfész modell (szolgáltató)  $\mathcal{M}_j$  inkompatibilis ha  $\mathcal{M}_{ij} = \mathcal{M}_i \triangleright \mathcal{M}_j$  deadlock-ot tartalmaz
  - Bizonyítás:

$\mathcal{M}_i$  inkompatibilis  $\mathcal{M}_j$ -vel, ha  $\mathcal{L}(\mathcal{M}_i) \supset \mathcal{L}(\mathcal{M}_j) \rightarrow \mathcal{M}_{ij}$  deadlock-ot tartalmaz

Ha  $\mathcal{L}(\mathcal{M}_i) \supset \mathcal{L}(\mathcal{M}_j)$  estén létezik  $\hat{\delta}$  kezdeti tüzelés szekvencia  $\mathcal{M}_i$ -ben, úgy hogy  $\ell(\hat{\delta}) \notin \mathcal{L}(\mathcal{M}_{ij})$ -nek  $\rightarrow$  tartalmaz olyan kezdeti tüzelést ami nem engedélyezett  $\mathcal{M}_{ij}$ -ben.

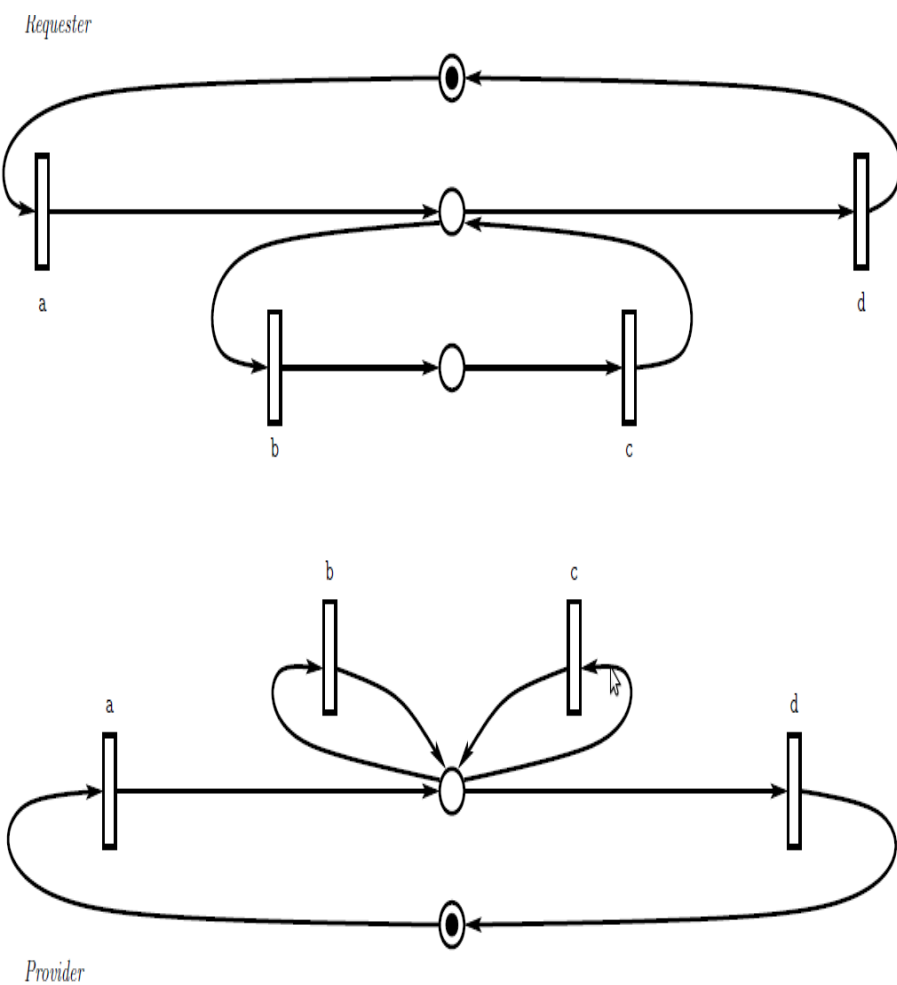


# Tartalom

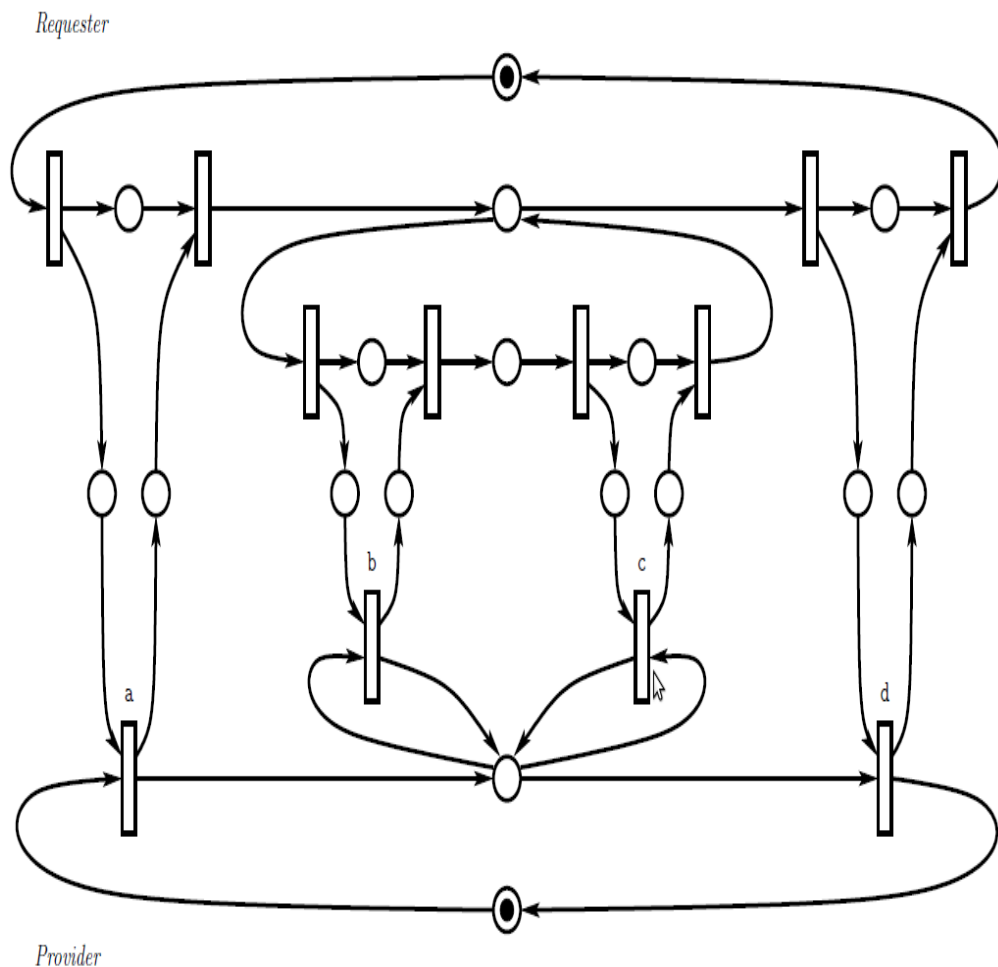
- Bevezetés, probléma felvetés
- Komponensek modellezése Petri hálókkal
- Komponensek összecsatolása és kompatibilitása
- **Alkalmazási lehetőségek**
- Probléma kiterjesztés: Egy generikus megoldás n-protokoll kompatibilitás vizsgálatra

# Alkalmazási lehetőségek

Gyakorlati példa: Adatbázis kérés-kiszolgálás komponens csatolás:  $(a(bc)^*d)^*$



Eljárás előtt



Eljárás után



# Tartalom

- Bevezetés, probléma felvetés
- Komponensek modellezése Petri hálókkal
- Komponensek összecsatolása és kompatibilitása
- Alkalmazási lehetőségek
- **Probléma kiterjesztés: Egy generikus megoldás n-protokoll kompatibilitás vizsgálatra**

# Egy generikus megoldás n-protokoll kompatibilitás vizsgálatra

- Miket nem kezel az előző modell?
  - Érték átadást nem megfigyelhető és belső ( $\mathcal{T}$ ) komponens állapokban
  - Kompatibilitás vizsgálatot tetszőleges  $n$  ( $n \geq 2$ ) komponens között
  - Automatikus kompatibilitás vizsgálatot a komponensek között
- Mivel lehet orvosolni ezeket a hiányosságokat?
  - Symbolic Transition System (STS) alapú működéssel
  - Globális állapotok bevezetésével
  - Kompatibilitási mértékek bevezetésével:
    - Bidirectional Complementarity (BC)
    - Unspecified receptions (UR)
    - Deadlock freeness (DF)
    - One Path (OP)
    - Unidirectional Complementarity (UC)