

MYKYTA BOBROV (224747)  
PAWEŁ MAZUR (184197)

8 CZERWCA 2015

---

LOGIKA TEMPORALNA I AUTOMATY CZASOWE

# MODELOWA WERYFIKACJA SYSTEMÓW W UPPAAL

---

*Prowadzący:*

DR INŻ. PAWEŁ GŁUCHOWSKI

*Grupa:*

PN 9:15

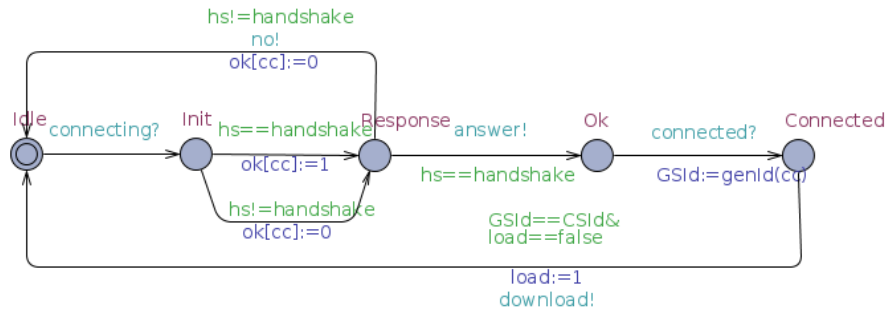
# 1 Zadanie 1

Na potrzeby realizacji pierwszej części zadania został utworzony szablon serwera (*Sever*), klienta (*Client*) oraz sesji (*Session*). Szablony te zostały zaprezentowane kolejno na rys. 1-3.

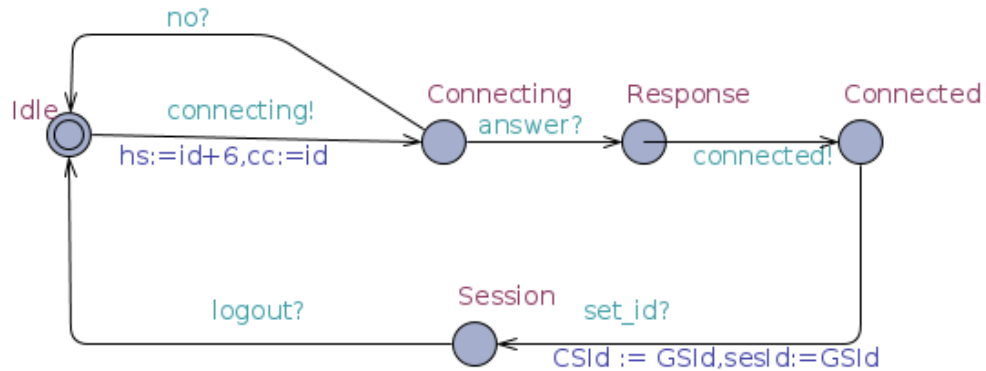
Klient łączący się do serwera przesyła swój identyfikator (*hs*), który po zatwierdzeniu przez serwer otrzymuje od niego sygnał (*answer*), następnie po podłączeniu (*connected*) tworzona jest sesja klienta (*set\_id*).

Po otwarciu sesji (*Session.Open*) możliwe jest przeście do stanu pobierania (*Session.Download*), który może trwać nie więcej niż 10000 sekund, po czym następuje wylogowanie klienta (przechodzi w stan *Client.Idle*) i zamknięcie sesji (*Session.Closed*).

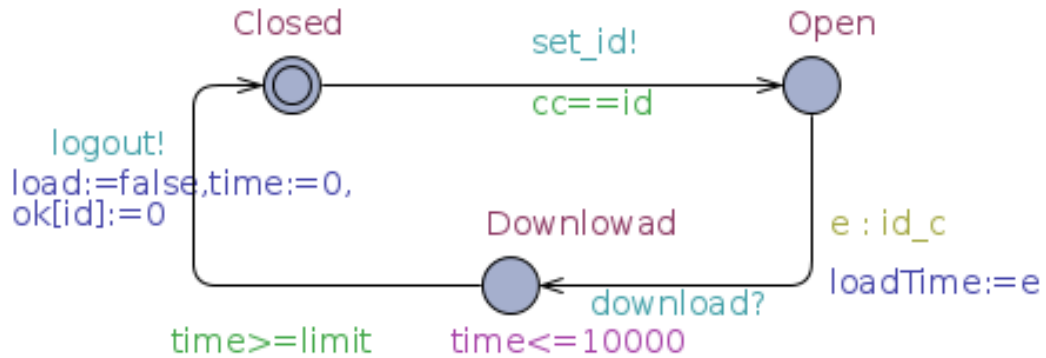
W przypadku podania przez klienta nieprawidłowego identyfikatora *hs*, połączenie zostaje zablokowane i klient oraz serwer wracają do stanu *Idle* (*Server.Idle*, *Client.Idle*).



Rysunek 1: Szablon serwera



Rysunek 2: Szablon klienta



Rysunek 3: Szablon sesji

## 2 Zadanie 2

W celu weryfikacji zbudowanego modelu zostały przygotowane i zweryfikowane następujące własności:

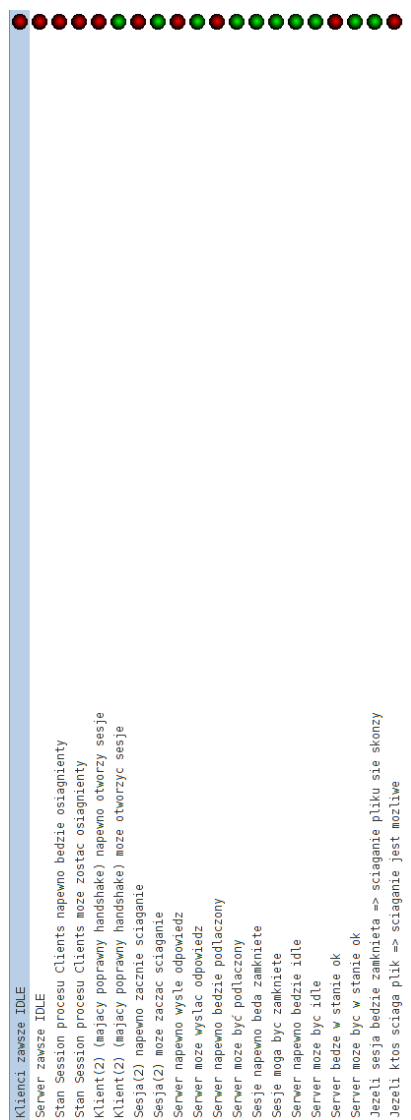
### 1. Bezpieczeństwo

- (a) Serwer będzie zawsze w stanie oczekiwania (FALSE)  
 $A \Box \text{Server.Idle}$
- (b) Klienci będą zawsze w stanie oczekiwania (FALSE)  
 $A \Box \text{forall}(i : \text{int}[0, 2]) \text{Client}(i).Idle$

### 2. Osiągalność

- (a) Stan Session procesu Clients napewno będzie osiągnięty (FALSE)  
 $A \langle \rangle \text{forall}(i : \text{int}[0, 2]) \text{Client}(i).Session$
- (b) Stan Session procesu Clients może zostać osiągnięty (FALSE)  
 $E \langle \rangle \text{forall}(i : \text{int}[0, 2]) \text{Client}(i).Session$
- (c) Klient(2) (mający poprawny handshake) na pewno otworzy sesje (FALSE)  
 $A \langle \rangle \text{Client}(2).Session$
- (d) Klient(2) (mający poprawny handshake) może otworzyć sesje (TRUE)  
 $E \langle \rangle \text{Client}(2).Session$
- (e) Sesja(2) na pewno zacznie ściągni (FALSE)  
 $A \langle \rangle \text{Session}(2).Download$
- (f) Sesja(2) może zacząć ściągnięcie (TRUE)  
 $E \langle \rangle \text{Session}(2).Download$
- (g) Serwer napewno wyśle odpowiedź (FALSE)  
 $A \langle \rangle \text{Server.Response}$
- (h) Serwer może wysłać odpowiedź (TRUE)  
 $E \langle \rangle \text{Server.Response}$

- (i) Serwer na pewno będzie podłączony (FALSE)  
 $A \nleftrightarrow Server.Connected$
  - (j) Serwer może być podłączony (TRUE)  
 $E \nleftrightarrow Server.Connected$
  - (k) Sesje na pewno będą zamknięte (TRUE)  
 $A \nleftrightarrow forall(i : int[0, 2])Session(i).Closed$
  - (l) Sesje mogą być zamknięte (TRUE)  
 $E \nleftrightarrow forall(i : int[0, 2])Session(i).Closed$
  - (m) Serwer na pewno będzie w stanie idle (TRUE)  
 $A \nleftrightarrow Server.Idle$
  - (n) Serwer może być w stanie idle (TRUE)  
 $E \nleftrightarrow Server.Idle$
  - (o) Serwer na pewno będzie w stanie ok (FALSE)  
 $A \nleftrightarrow Server.Ok$
  - (p) Serwer może być w stanie ok (TRUE)  
 $E \nleftrightarrow Server.Ok$
3. (a) Jeśli sesja będzie zamknięta to ściąganie pliku się skończy (TRUE)  
 $forall(i : int[0, 2])Session(i).Closed \rightarrow load == false$
- (b) Jeśli ktoś ściąga plik to ściąganie jest możliwe (FALSE)  
 $load! = true \rightarrow forall(i : int[0, 2])Session(i).Downloaded$



Rysunek 4: Wynik działania weryfikatora w programie UPPALL