

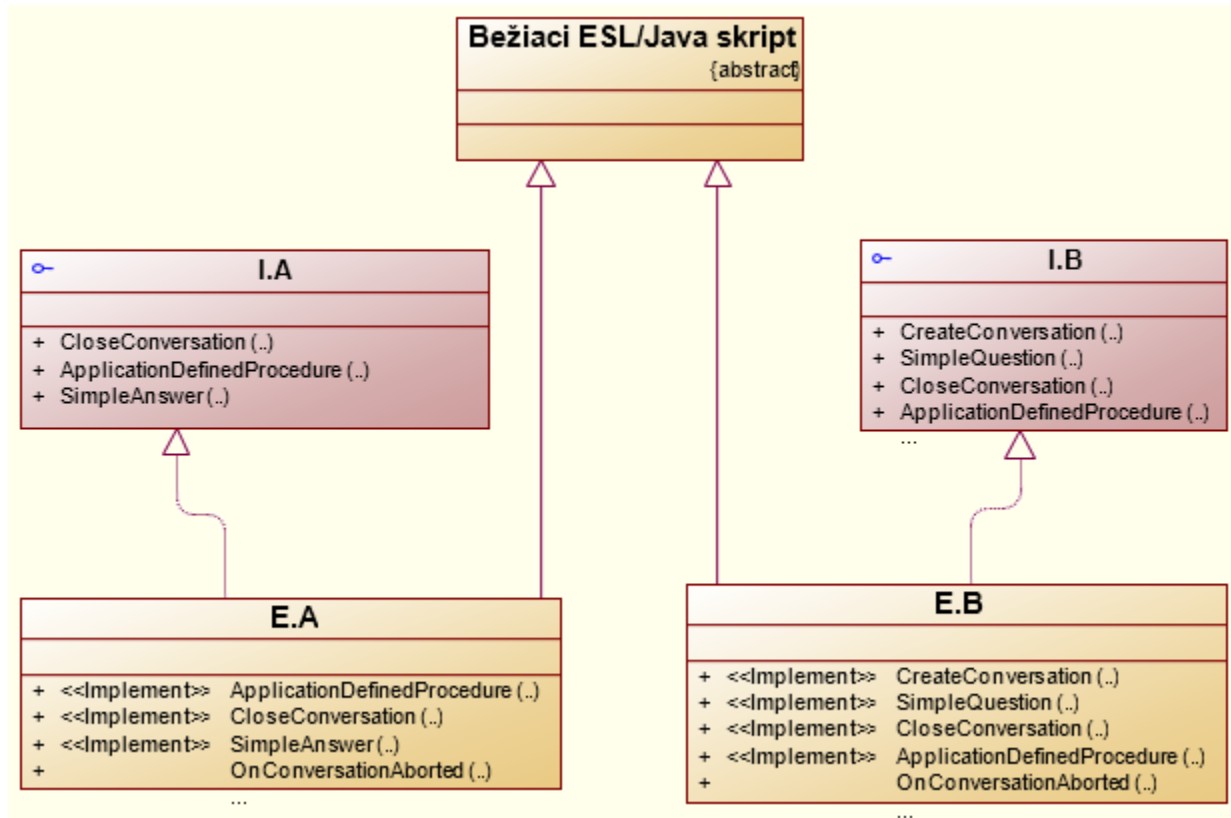
# Používanie konverzácií

Mechanizmus konverzácií vysvetlíme na ich použití medzi ESL skriptami.

Predpokladáme, že existujú inštancie dvoch skriptov typu server **E.A** a **E.B** (za server skript je pokladaná aj otvorená schéma). Tieto bežia na dvoch event handleroch A.EVH a B.EVH v inšanciách 0 (týmto je každý bežiaci skript jednoznačne adresovateľný podľa schémy *proces;event;inšancia eventu*).

Za účelom obsluhy konverzácií sú vytvorené objekty typu *ESL Rozhranie* I.A a I.B, ktoré implementujú eventy E.A a E.B (použitie objektov typu *ESL Rozhranie* nie je podmienkou).

Stav je znázornený na nasledovnom obrázku:



Konfigurácia jednotlivých rozhraní je nasledovná:

```
; procedura vyuzivajuca existujucu konverzaciu
RPC PROCEDURE [_hTC, TC_C] ApplicationDefinedProcedure(IN INT _parameters)

; obsluha zatvorenia konverzacie
RPC PROCEDURE [_hTC, TC_E] SimpleAnswer(IN INT _parameters)

; obsluha zatvorenia konverzacie
RPC PROCEDURE [_hTC, TC_E] CloseConversation(IN INT _parameters)
```

ESL skript implementujúci rozhranie I.A by mal obsahovať implementáciu ERROR procedúry.

```
; volana aplikacnym serverom pri prerusení transakcie
RPC PROCEDURE [_hTC, ERROR] OnConversationAborted
```

```

; procedura pre vytvorenie trvajucej konverzacie (CALL TC_B)
RPC PROCEDURE [_hTC, TC_B] CreateConversation(IN INT _parameters)

; jedna otazka (moznych viac odpovedi) (CALL TC_BE)
RPC PROCEDURE [_hTC, TC_BE] SimpleQuestion(IN INT _parameters)

; procedura pre zatvorenie konverzacie (CALL TC_E)
RPC PROCEDURE [_hTC, TC_E] CloseConversation(IN INT _parameters)

; procedura vyuzivajuca existujucu konverzaciu
RPC PROCEDURE [_hTC, TC_C] ApplicationDefinedProcedure(IN INT _parameters)

```

ESL skript implementujúci interface I.B by mal obsahova implementáciu ERROR procedúry.

```

; volana aplikacnym serverom pri prerusení konverzacie
RPC PROCEDURE [_hTC, ERROR] OnConversationAborted

```

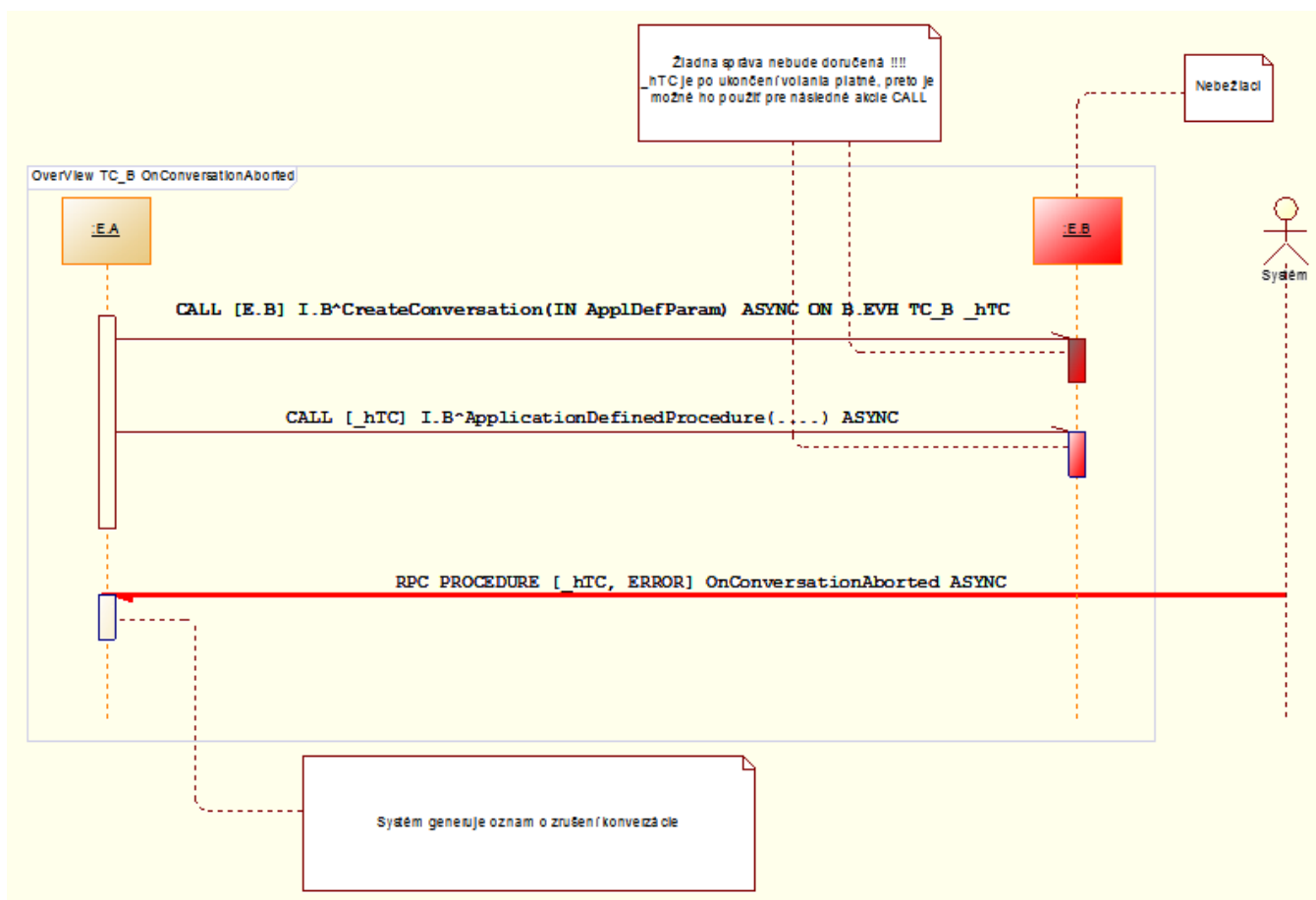
## Vytvorenie konverzácie (TC\_B)

O konverzácii, ktorá je vytvorená volaním **CALL ... TC\_B** predpokladáme, že bude mať dlhšie trvanie, resp. je potrebná pre výmenu zložitejšej sekvencie správ. Konverzácia vzniká tak, že ju niektorý skript inicializuje vzhľadom k druhému skriptu. V našom príklade bude konverzáciu inicializovať skript E.A vzhľadom k skriptu E.B nasledovne:

```
CALL [E.B] I.B^CreateTransaction(IN ApplDefParam) ON B.EVH TC_B _hTC
```

príom `_hTC` je deklarovaná premenná typu *INT*. Akcia zabezpečí vytvorenie konverzácie, ktorá bude naalej identifikovaná hodnotou `_hTC`, a asynchrónne zaslanie požiadavky na vykonanie RPC procedúry *I.B^CreateTransaction* do skriptu E.B. Treba si uvedomiť, že ak akcia CALL bude vykonaná úspešne, konverzácia vznikne aj napriek tomu, že adresovaný skript E.EV:E.B;0 je z nejakého dôvodu neaktívny (nie je možné vykonať volanú procedúru). Tento stav je pokladaný za chybu konverzácie a systém zabezpečí volanie procedúry *[\_hTrans, ERROR]* s príslušným `_hTC`.

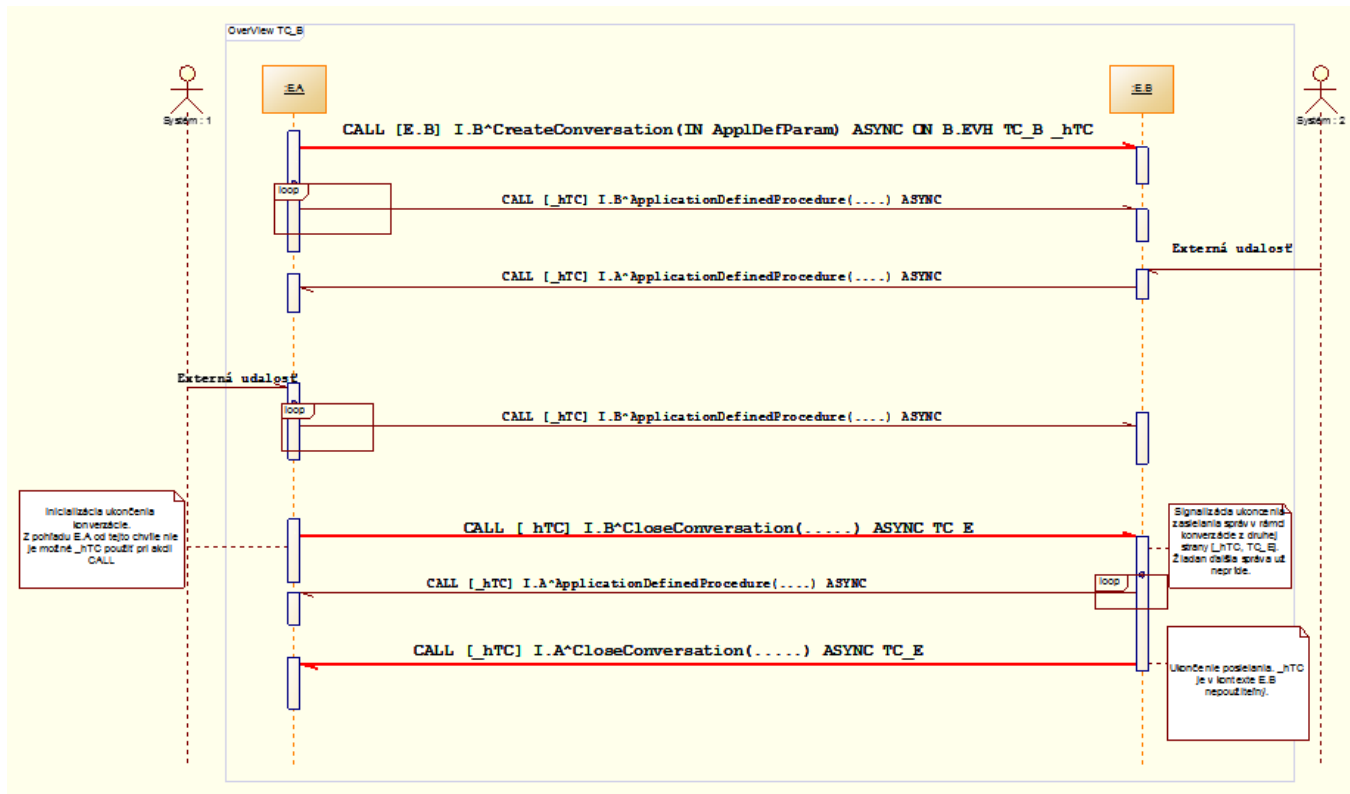
```
RPC PROCEDURE [_hTC, ERROR] OnConversationAborted
```



V prípade úspešného zavolania procedúry *I.B^CreateConversation* je konverzácia považovaná za vytvorenú a oba skripty sú vzhľadom k nej **rovnocenné** a môžu ju využívať na výmenu správ formou **asynchrónneho** volania RPC procedúr.

Zatvorenie konverzácie prebieha volaním procedúry druhej strany (keže sú oba skripty rovnocenné, nezáleží na tom, ktorá strana zatvorenie iniciuje) akciou CALL s kúovým slovom **TC\_E**. V našom prípade akciu volá skript E.A. Po zavolaní akcie je identifikátor konverzácie naalej nepoužitý pre ďalšie volania procedúr (výstup je zatvorený), ale umožňuje identifikovať konverzáciu pri obsluhu prichádzajúcich správ, ktoré môžu ešte prísť (**RPC PROCEDURE [\_hTC]**). K úplnému zániku konverzácie dochádza až potvrdením tejto žiadosti zo skriptu E.B (skript E.B zavolá CALL [\_hTC] I.A^CloseTransaction(...) ASYNC T\_E).

Popísaná výmena správ je znázornená:



erveným sú vyznačené volania, ktoré riadia stav konverzácie.

Systém automaticky generuje nový stav konverzácie pri udalostiach ako napríklad:

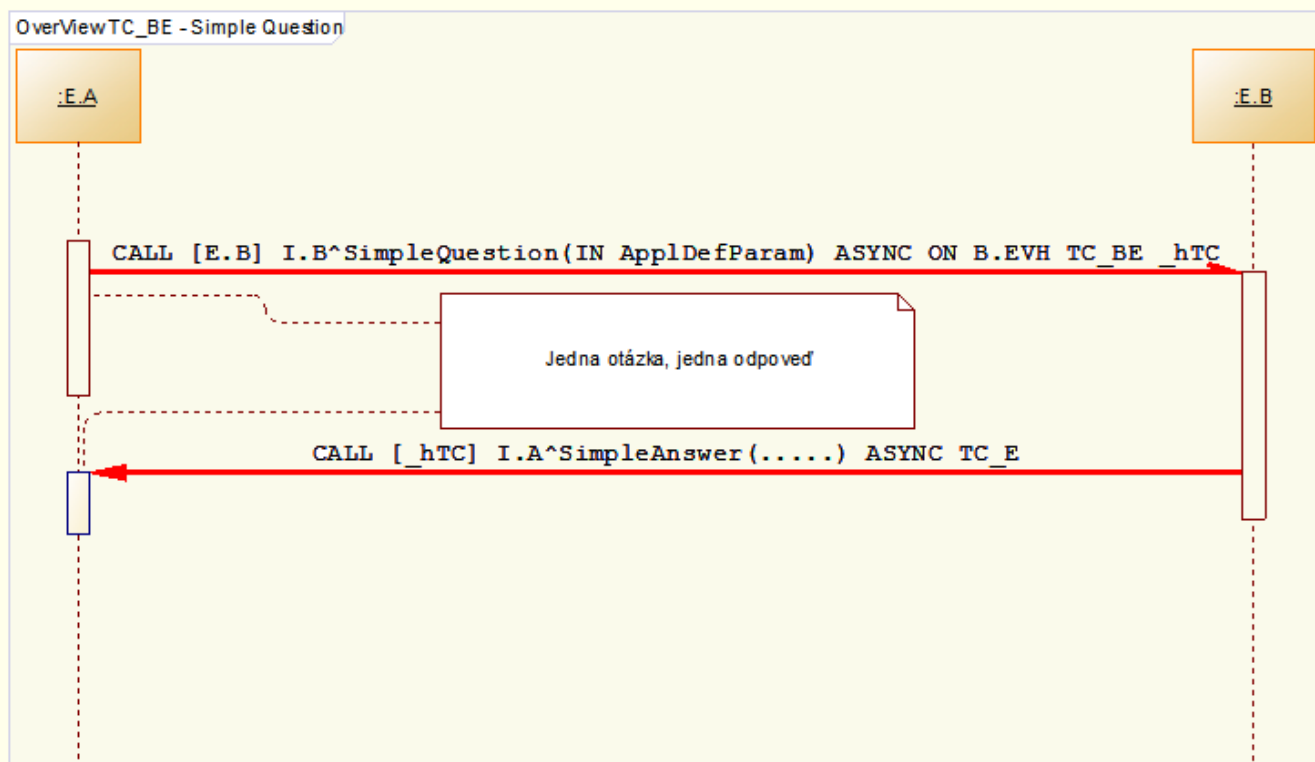
- stop alebo pád procesu (A.EVH alebo B.EVH),
- zastavenie innoti skriptu, ktorý je účastníkom konverzácie (uloženie z CNF, GRE, ...),
- ...

Systémom generované zmeny stavu sú reportované účastníkovi konverzácie formou volania **RPC PROCEDURE [\_hTC, ERROR] ....**

## Vytvorenie a súasné zatvorenie konverzácie - Question (TC\_BE)

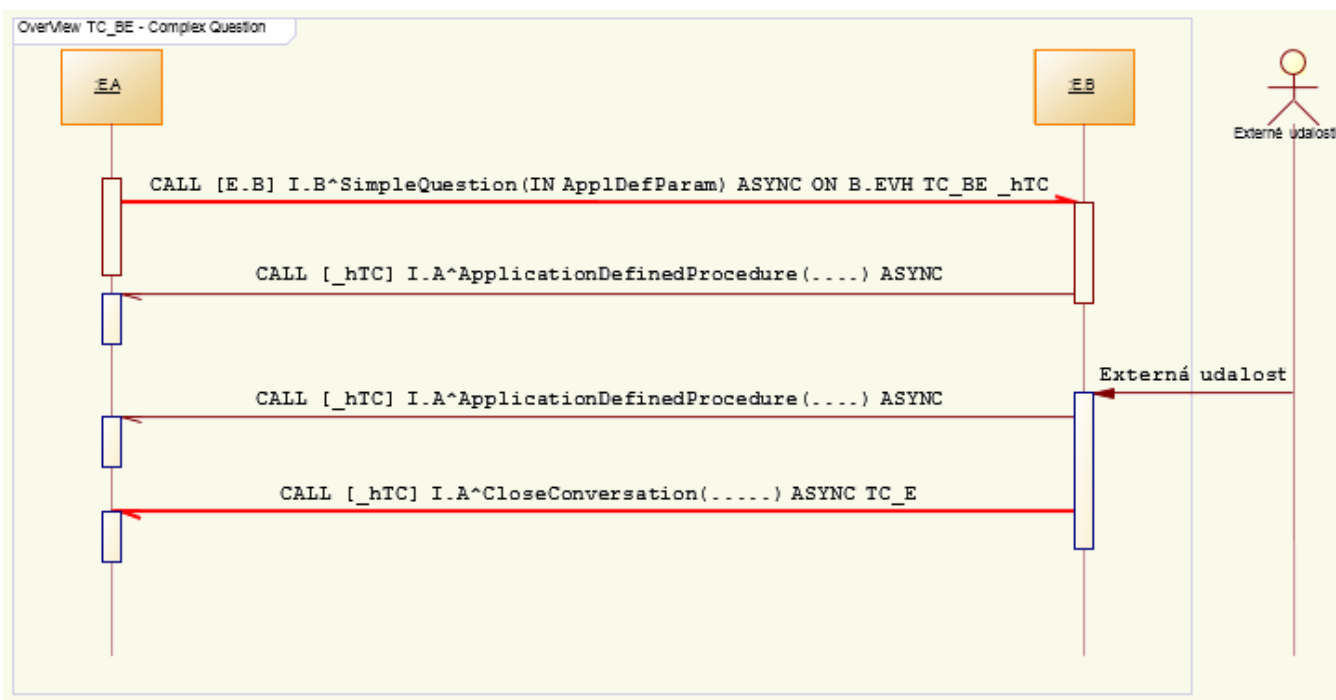
Otvorenie konverzácie typu **TC\_BE** má význam, ak má výmena správ charakter jednej otázky a jednej alebo viacerých odpovedí.

Prvá situácia je znázornená:



V prípade chyby pri doručení otázky (volania RPC procedúry) je o tomto skript E.A informovaný volaním procedúry **RPC PROCEDURE [\_hTC, ERROR]**.

Druhú situáciu znázorňuje:



## Ukonenie konverzácie (TC\_E)

Ukonenie konverzácie je povinnos každej zúčastnenej strany (ak táto nebola ukonená volaním procedúry **RPC PROCEDURE [\_hTC, ERROR]**).

**i** Súvisiace stránky:

