

Aplikane definované konverzácie

Na systémovej úrovni je v D2000 podporený mechanizmus, ktorý umožňuje vytvori konverzáciu medzi **dvoma** "skriptovatenými" a zároveň "adresovatenými" entitami v aplikácii.

V praxi je výhodné použiť konverzácie všade tam, kde na aplikanej úrovni vzniká nejaká relácia (alebo väzba) medzi inštanciami ESL skriptov alebo procesom implementovaným prostredníctvom D2000 JAPI. V prípade, že takéto väzby sú realizované prostredníctvom konverzácie, je možné na aplikanej úrovni jednoducho ošetri stav, kedy "druhá strana" tejto relácie prestane existovať.

Napríklad:

Prostredníctvom server eventu **E.MAIL_SERVER** sa implementuje služba, ktorá umožní klientom (iným server eventom, schémam alebo procesu D2000 JAPI) pripoji sa k odberu správ. Toto pripojenie sa realizuje napríklad metódou:

```
RPC PROCEDURE Attach
....
END Attach
```

Príslušný klient sa prihlási k odberu správ jej zavolaním:

```
CALL [E.MAIL_SERVER] Attach ON SELF.EVH
```

E.MAIL_SERVER implementuje alšiu službu, ktorá umožní posla správu prihláseným klientom:

```
RPC PROCEDURE Send(IN TEXT _msg)
...
END Send
```

Pri zániku klienta je tento povinný sa odregistrova volaním metódy:

```
RPC PROCEDURE Detach
....
END Detach
```

Všetko funguje dobre až do okamihu, kedy klient, ktorý zaniká, nestihne zavola metódu **Detach** a tým spôsobí, že služba **E.MAIL_SERVER** sa nedozvie o tom, že daný klient už neexistuje.

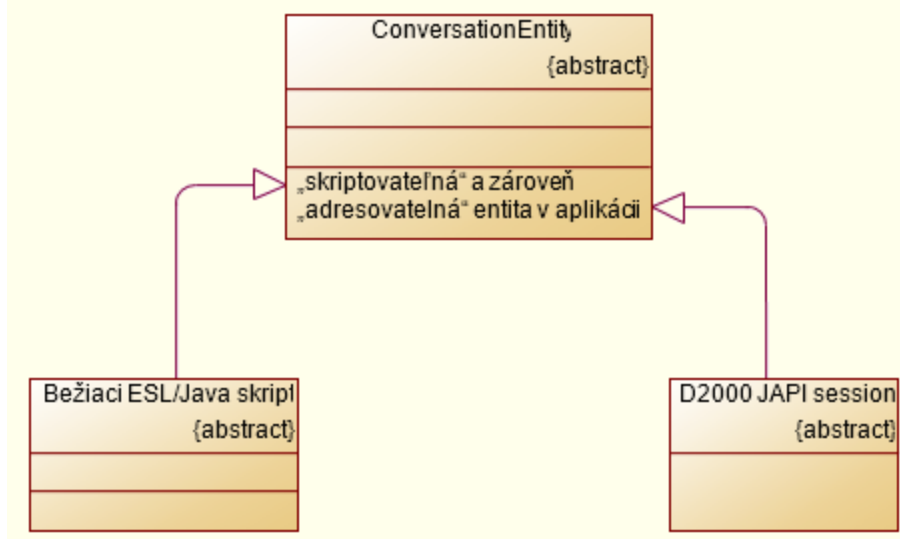
Druhá možná chyba je tá, že ak sa služba **E.MAIL_SERVER** z nejakého dôvodu reinitializuje (reštart procesu, uloženie novej verzie služby), žiadny klient sa nedozvie o tom, že je potrebné sa znovu zaregistrova volaním metódy **Attach**.

Prostredníctvom konverzácií je možné ošetri práve tieto problematické prípady.

Špecifikácia

Zoznam entít, ktoré sa môžu zúčastni na konverzácii:

- bežiaci ESL/Java skript (v kontexte otvoreného objektu typu *Schéma*, alebo bežiaceho server eventu,
- proces pripojený k aplikácii prostredníctvom D2000 JAPI.



Konverzácia je jedinečná relácia medzi entitami a slúži ako prostriedok (alebo transportná vrstva) pre ich vzájomnú **asynchrónnu** procedurálnu komunikáciu (**volaním RPC procedúr a metód rozhraní, ktoré implementujú**), pričom zohaduje ich životný cyklus. V praxi to znamená, že:

1. Vytvorenie konverzácie je riadené aplikovanou požiadavkou (volaním RPC procedúry). Tá vzniká medzi žiadateľom a entitou, ktorú táto požiadavka adresuje. Každá konverzácia je v ESL identifikovaná hodnotou typu *INT* (handle konverzácie). Táto hodnota je jedinečná v rámci inštalácie eventu a nie je možné ju zdieľať medzi rôznymi inštanciami eventu alebo schém.
2. Ukonenie konverzácie môže byť iniciované ubovenou zúčastnenou entitou alebo automaticky systémom, ak došlo k zániku niektorej zo zúčastnených strán (strata komunikácie v dôsledku zlyhania komunikačných ciest, zatvorenie objektu, uloženie objektu (jeho konfigurácia) a následný refresh).

Vytvorenie konverzácie, **zasielanie** správ a jej **ukonenie** je realizované prostredníctvom volaní RPC procedúr. Obsluha neoakávaného **prerušenia** transakcie je realizovaná volaním RPC procedúry skriptu, ktorý je zúčastnený transakcie.

Za týmto účelom je rozšírená syntax deklarácie RPC procedúr.

```

RPC PROCEDURE [_hTC, TC_B] NewTransaction(parameters)
RPC PROCEDURE [_hTC, TC_E] EndTransaction(parameters)
RPC PROCEDURE [_hTC, TC_BE] QuestionTransaction(parameters)
RPC PROCEDURE [_hTC, TC_C] UserProc(parameters)
RPC PROCEDURE [_hTC, ERROR] TransactionInterrupted
  
```

Premenná *_hTC* je použitá pre identifikáciu konverzácie (množstvo konverzácií nie je obmedzené). V rámci obsluhy procedúry je vždy automaticky deklarovaná ako lokálna premenná typu *IN INT*.

Rozšírená syntax pre volanie procedúr.

```

CALL [objIdent] NewConversation(parameters) ASYNC ... TC_B _hTC
CALL [_hTC] EndConversation(parameters) ASYNC ... TC_E
CALL [objIdent] QuestionConversation(parameters) ASYNC ... TC_BE _hTC
CALL [_hTC] UserProc(parameters) ASYNC TC_C
  
```

Premenná *_hTC* je výstupná v kontexte akcií **TC_B** a **TC_BE** a vstupná v kontexte akcií **TC_C** a **TC_E**. Všetky volania sú zásadne **asynchrónne**.

Kúčové slová **TC_B**, **TC_E**, **TC_BE**, **ERROR** sú použité pre procedúry, ktoré sa zúčastňujú na riadení životného cyklu konverzácie (obsluhujú alebo iniciujú udalosti, ktoré menia jej stav). Prostredníctvom kúového slova **TC_C** oznaujeme volanie, ktoré nemení stav konverzácie.

- **TC_B** – "Transaction Conversation - Begin"
V rámci akcie "**CALL ... TC_B _hTC**" ide o volanie procedúry a zároveň o vytvorenie konverzácie. Systém oakáva, že volaná procedúra je deklarovaná použitím kúového slova **TC_B**. Identifikátor novej konverzácie je vrátený prostredníctvom hodnoty *_hTC*. Používa sa na jej identifikáciu pri obsluhu ďalších udalostí (RPC PROCEDURE [_hTC...]) alebo pri posielaní správ v rámci nej (akcia CALL [_hTC]...). Pre posielanie správ je platný až do okamihu jej uzatvorenia pre posielanie (použitie akcie **CALL ... TC_E**) alebo do okamihu príjmu udalosti prerušenia konverzácie (**RPC PROCEDURE [_hTC, ERROR]...**).
- **TC_E** – "Transaction Conversation - End"
V rámci akcie "**CALL [_hTC] ... TC_E**" ide o volanie procedúry a zároveň o zatvorenie konverzácie. Zatvorenie musí byť spracované procedúrou na to určenou (**RPC PROCEDURE [_hTC, TC_E]...**). Od okamihu zatvorenia transakcie akciou CALL je *_hTC* použitý jedine na identifikáciu konverzácie pri obsluhu udalostí až po poslednú udalosť (obsluha **RPC PROCEDURE [_hTrans, TC_E]... alebo RPC PROCEDURE [_hTC, ERROR]...**).

- **TC_BE** – "Transaction Conversation - Begin-End"

V rámci akcie "**CALL ... TC_BC _hTC**" ide o volanie procedúry a zároveň o vytvorenie konverzácie, ktorá je špecifická tým, že volajúci skript konverzáciu zároveň zatvára a oakáva len odpove(e). Hodnota _hTC teda nie je alej použitá pre výmenu správ v akcii **CALL**.

- **ERROR** – Conversation was unexpectedly interrupted – **ERROR**

Udalos je generovaná systémom automaticky:

1. pri neoakávanom zániku konverzácie (prerušenie komunikanej cesty, zánik spolupracujúcej entity v rámci konverzácie),
2. pri volaní neexistujúcej RPC procedúry,

Udalos ERROR bude generovaná pre tú entitu, ktorá už konverzáciu pozná.

V prípade volania typu:

- TC_BE len pre volajúcu,
- TC_C pre obe,
- TC_E pre obe.

Cieom takéhoto zabezpečenia je, aby aplikovaný server nepodporoval konverzácie, ktorých účastníci si nerozumejú.

3. v prípade, že volaná RPC procedúra v rámci konverzácie bude ukonená výnimkou (ESL RunTimeError). Cieom takéhoto zabezpečenia je, aby aplikovaný server nepodporoval konverzácie, ktoré nie sú korektne implementované.

Parameter _hTC je naalej nepoužitý. Takto deklarovaná procedúra sa v zdrojovom kóde ESL skriptu musí nachádzať najviac jeden krát.

Procedúra *UserProc* je užívateľsky definovaná procedúra, ktorá slúži na prenos správ v rámci konverzácie bez zmeny jej stavu. Meno procedúry je vo všetkých prípadoch definované užívateľom. V prípade volania procedúry ESL Interface, ktorý implementuje ESL skript, platí popísaná syntax.

Pre volanie procedúry, ktorú implementuje proces, je syntax nasledovná:

```
CALL [(0)] I.Interface^ProcName(parameters) ON ProcesName.EXT
```

0 – je hodnota nula



Súvisiace stránky:

[Používanie konverzácií](#)

[Príklad E.MAIL_SERVER](#)