

Serwis cytatów – sentencji

Opis protokołu

Piotr Dobrowolski, 291528

19 kwietnia 2011

Spis treści

| | | |
|----------|---|----------|
| 1 | Streszczenie | 2 |
| 2 | Opis celów protokołu | 2 |
| 3 | Terminologia | 2 |
| 4 | Założenia | 2 |
| 4.1 | Warstwa komunikacji | 2 |
| 4.2 | Wykorzystanie innych protokołów | 2 |
| 4.3 | Model komunikacji | 2 |
| 5 | Opis formatu komunikatów | 3 |
| 5.1 | Założenia | 3 |
| 5.2 | Typy pomocnicze | 3 |
| 5.3 | Typy komunikatów | 3 |
| 6 | Opis wymienianych komunikatów | 4 |
| 6.1 | Wysyłane przez klienta | 4 |
| 6.1.1 | Zgłaszanie się do serwera | 4 |
| 6.2 | Wysyłane przez serwer | 4 |
| 6.2.1 | Komunikat z danymi | 4 |
| 7 | Opis stanów | 4 |
| 7.1 | Klient | 4 |
| 7.2 | Serwer | 5 |
| 8 | Numery | 6 |
| 8.1 | Stałe | 6 |
| 8.2 | Kody błędów | 6 |

1 Streszczenie

Niniejszy dokument opisuje specyfikację protokołu serwisu cytatów – sentencji. Składa się z następujących sekcji:

- opis celów protokołu,
- opis założeń,
- opis formatu komunikatów,
- opis stanów.

2 Opis celów protokołu

Protokół ma obsługiwać sposób szybkiego przekazywania krótkiego tekstu (cytatu/sentencji). Cytaty będą udostępniane klientom przez serwer. Protokołu jest nastawiony na szybkość kosztem niezawodności dostarczanych tekstów.

3 Terminologia

Klient: użytkownik instancji protokołu, który chce pobrać cytaty z bazy utrzymywanej na serwerze,

Serwer: aplikacja użytkownika udostępniająca teksty klientom,

Cytat: będę używał wymiennie z tekst, sentencja. Są to dane tekstowe przesyłane protokołem,

Długi cytaty: jest to cytaty, który nie mieści się w jednym komunikacie i należy go podzielić.

4 Założenia

Serwer odpowiada na żądania klientów. W jednej chwili może obsługiwać wielu na raz. Oraz może wysyłać wiele różnych cytatów do tego samego klienta w tej samej chwili.

4.1 Warstwa komunikacji

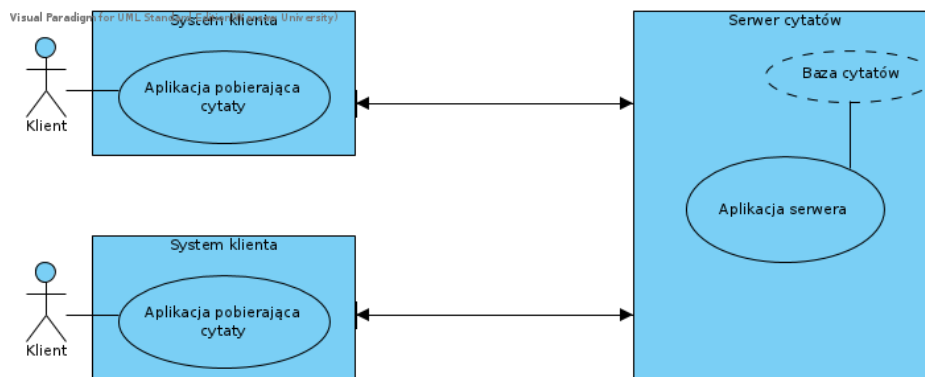
Protokół działa w warstwie aplikacji.

4.2 Wykorzystanie innych protokołów

Protokół ma działać jak najszybciej, nie gwarantując niezawodności. Używany jest protokół UDP.

4.3 Model komunikacji

Protokół działa w modelu klient-serwer. Pojedynczy serwer z cytatami, udostępnia je wielu klientom. Jak przedstawiono na diagramie poniżej. Klient nasłuchuje na dynamicznie przydzielonym porcie. Serwer na porcie `s_port`.



5 Opis formatu komunikatów

5.1 Założenia

- Protokół stosuje sieciowy porządek oktetów w liczbach,
- Dopuszczalne uzupełnianie formatu zerami.

5.2 Typy pomocnicze

```

PART_ID_T {
    uint8    part_nr    = 0;
    uint8    all_parts  = 1;
}
  
```

Typ ten służy do odróżnienia komunikatu z tekstem, tak aby później można było poskładać długi cytat w dobrej kolejności.

Części **part_nr** numerowane są od 0 - oznacza numer kolejny części. **all_parts** - oznacza ilość, na którą został podzielony cytat.

```

QUOT_T {
    PART_ID_T      part_id;
    uint16         quot_length = max_quot_length;
    octet[quot_length]  quot;
}
  
```

quot_length - pole zawierające długość cytatu w oktetach.

Struktura zachowuje cytat (jedną z części cytatu długiego). Oraz identyfikator jednoznacznie określający która jest to część.

5.3 Typy komunikatów

```

QUOT_REQUEST_MSG_T {
    uint8    version = version_one;
}
  
```

Zgłoszenie się klienta. Służy do wysłania do serwera prośby o cytat. Pole **version** jest to aktualna wersja protokołu aplikacji klienta. Jest pozostawiona dla przyszłych wersji protokołu.

```

QUOT_RESPONSE_MSG_T {
    uint8      id;
    QUOT_T     quotation;
}

```

Struktura komunikatu wysyłanego z serwera do klienta, `id` oznacza identyfikator wysłanego cytatu (aby nie dopuścić do wymieszania części różnych cytatów). Wartość `id` pozostawiam do wyboru implementującemu protokół, należy zadbać o różne identyfikatory dla pary klient-serwerem. Pomiędzy klientami identyfikatory mogą się powtarzać.

6 Opis wymienianych komunikatów

Protokół posługuje się dwoma komunikatami: `QUOT_REQUEST_MSG` i `QUOT_RESPONSE_MSG`. (W nawiasach podane są argumenty, a po dwukropku typ komunikatu).

6.1 Wysyłane przez klienta

6.1.1 Zgłaszanie się do serwera

```
QUOT_REQUEST_MSG():      QUOT_REQUEST_MSG_T
```

Klient posługuje się tym komunikatem aby zgłosić chęć pobrania cytatu z serwera. Po odebraniu komunikatu serwer przygotowuje (losuje, dzieli) cytat i przechodzi do stanu wysyłania cytatu do tego klienta.

6.2 Wysyłane przez serwer

6.2.1 Komunikat z danymi

```
QUOT_RESPONSE_MSG(id, quotation):      QUOT_RESPONSE_MSG_T
```

W komunikacie zawarty jest:

- `id`: identyfikator cytatu, potrzebny w celu jednoznacznej identyfikacji cytatu.
- `quotation`: dane, `quotation.quot`: zawiera tekst cytatu. Znaki kodowane są za pomocą systemu UTF-8.

7 Opis stanów

W protokole występują stany tak jak na schematach poniżej. Z założenia pozostawione zostało implementującemu w jaki sposób ma zostać obsłużona sytuacja braku komunikatu (nie dostarczenia całego komunikatu) np. trzykrotne ponowienie pobrania komunikatu. Stany protokołu opisują jedną operację pobrania / wysłania komunikatu. Poniżej rozpatruję stany protokołu występujące po stronie klienta i serwera.

7.1 Klient

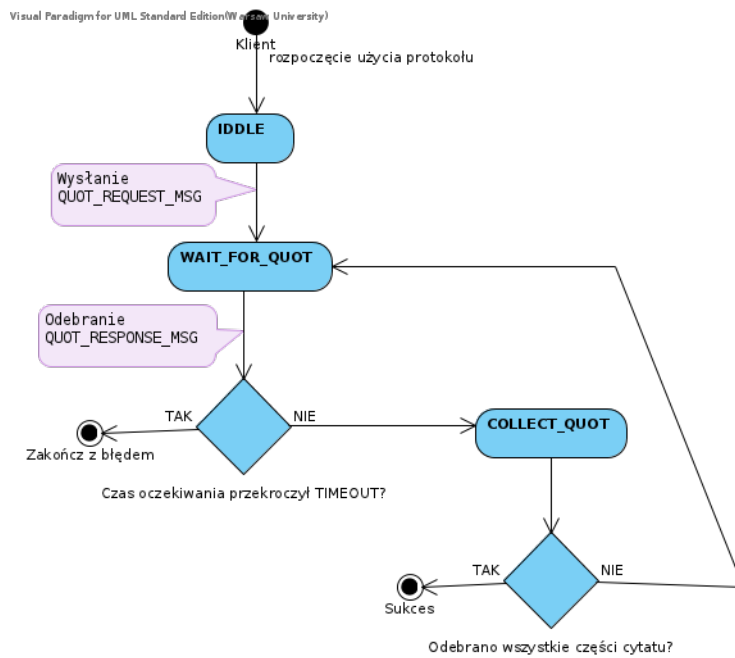
Klient początkowo zaczyna ze stanu `IDDLE`. Użytkownik wysyła żądanie do serwera (`QUOT_REQUEST_MSG()`), w chwili oczekiwania jest w stanie `WAIT_FOR_QUOT` a następnie zbiera pakiety odebrane z serwera `COLLECT_QUOT`. Tak działa każde osobne pobranie cytatu.

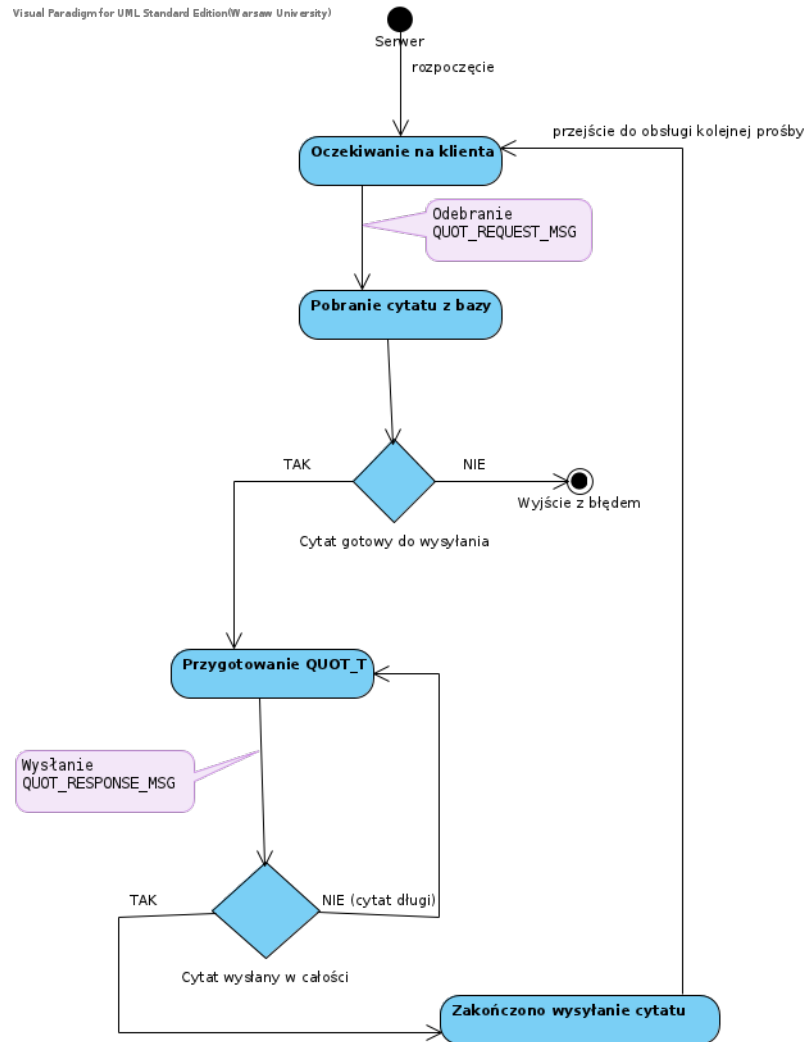
Klient identyfikuje cytat po odebraniu pierwszej części (zapisuje `id`). I na podstawie tego identyfikatora będzie dalej łączył pobrane później części. Podczas wysłania żądania zostaje zainicjowane odliczanie czasu, jeżeli czas oczekiwania na części cytatu które nie doszły przekroczy `timeout` sekund to pobieranie cytatu zostaje anulowane. `timeout` ustalane jest przez autora implementacji.

7.2 Serwer

Serwer odbiera od klientów żądania `QUOT_REQUEST_MSG()`. Przydziela temu zapytaniu identyfikator. Następnie dzieli cytat (jeżeli jest długi) zgodnie z `max_quot_length` i za pomocą komunikatów `QUOT_RESPONSE_MSG(id, quotation)` wysyła wiadomość do danego klienta. Działanie serwera może być zakończone wyłącznie za pomocą przerwania. Wtedy jeżeli serwer jeżeli jest w stanie 'Oczekiwanie na klienta' to nie inicjuje odbierania `QUOT_REQUEST_MSG` tylko zakańcza swoje działanie. W każdym innym stanie serwer dokańcza to co rozpoczął – tak aby znaleźć się w 'Oczekiwanie na klienta'.

Poniższe diagramy przedstawiają opisane wyżej stany.





8 Numery

8.1 Stałe

- `max_quot_length = 1024`; ustalany w czasie kompilacji,
- `timeout = 5`; proponowany,
- `s_port = 8078`;
- `version_one = 1`;

8.2 Kody błędów

- `ETIMEDOUT = 0xffff`; przekroczenie czasu oczekiwania przez klienta,
- `ENETUNREACH = 0xfffe`; sieć jest niedostępna,
- `EBADMSG = 0xfffd`; gdy otrzymany komunikat jest niepoprawny,

- `ESRVERR = 0xfffc`; błąd serwera — w przypadku, gdy serwer nie może odesłać cytatu (np. gdy serwer nie może pobrać cytatu z bazy)