Automatyzacja procesów workflow

Podstawowe informacje

Procedury workflow oparte są o notację BPMN i uwzględniają wszystkie najważniejsze elementy tej notacji. Składają się na nią:

- Etapy (Czynności - bloczki)
- Przejścia (strzałki)
- Decyzje (diament powodujący wyświetlenie decyzji dla użytkownika)
- Warunki (diament dokonujący ewaluacji warunków SQL)
- Złączenia (JOIN - w przypadku wymagania spełnienia poprzednich etapów)
- Pętle multi-instance z podprocesami (etapy ze znakiem +)
- Dane wejściowe (możliwość pobrania od użytkownika danych różnego typu: znakowe, liczbowe, listy pracowników, listy wyboru pobierane ze słowników kwerendami SQL itp).
- Właściwości (definiowalne zmienne)
- Przypisania (możliwość operowania na zmiennych)

Konfiguracja procedur pozwala tworzyć mapy procesów odnoszące się zarówno do dokumentów jak i spraw. Przykłady wykorzystania dostępne są tutaj: Wykorzystanie procedur

Komendy

Komendy mogą być wywoływane na aktywacji lub zakończeniu etapu. Opis komend i lista parametrów

Opis tworzenia własnych komend

Dla zaawansowanych

W workflow biorą udział następujące tabele:

- procedures_def - tabela procedur - przechowuje informacje o procedurze np. Zatwierdzenie faktury kosztowej
- stages_def - tabela etapów - przechowuje definicje poszczególnych etapów np. Akceptacja Prezesa
- stages - instancje etapów - przechowuje informacje o zapisanych etapach konkretnych procesów: spraw, dokumentów
- proc_actions - akcje powiązane z procedurami lub z etapami, wykonują się przed lub po zapisie np. beforeStageChange
- action_commands - komendy wykonywane przez system na akcjach - wybierane spośród zawartych w katalogu commands - można dodać parametry, które dodają się do standardowych dwóch Obiektu Akcji oraz obiektu encji powiązanej z wykonywaną akcją np. Dokument albo Sprawa

Wykorzystanie własności, danych wejściowych i przypisań

Potężne możliwości silnika workflow systemu eDokumenty możliwe są m.in. dzięki wykorzystaniu parametrów i zmiennych które mogą być dynamicznie przetwarzane podczas wykonywania procedury. Dane mogą być pobierane od użytkownika, ale również przetwarzane przez sam workflow.
Do danej wejściowej i własności odwołujemy się (w warunkach lub przypisaniach) poprzez nazwę poprzedzoną znakiem "$" oraz całość zamykamy w nawiasy "{}" (np. {$Akceptant}).

Dane wejściowe

Dane wejściowe służą tym samym czym odczyt standardowego wejścia w konsoli czy programie (czyli pobraniu od użytkownika znaków). Można je pobierać z różnych formantów (pól tekstowych, list wyboru, list pracowników). Najciekawszą opcją jest opcja SELECT która pozwala zdefiniować dowolną kwerendę SQL zwracającą potrzebną nam w danym etapie listę (np. kierowników, księgowych, zasobów itp). Przykładowa lista dla atrybutu CZŁONEK ZARZĄDU potrzebna do wyboru osoby podpisującej umowę:

```
SELECT orunid as value, fullnm || ' - ' || ndenam as caption FROM orgtree_view WHERE orunid IN (3,14,15,16)
```

Inny przykład to pobranie identyfikatora stanowiska, wystarczy w tym celu wybrać opcję orunid[]. 

UWAGA

Dane wejściowe jeśli nie są wypełnione zwracają zawsze ciąg znaków 'NULL' oprócz listy wartości która zwraca pusty string . *W przypisaniach i parametrach do komend należy więc używać konstrukcji NULLIF (param1, param2) która zwraca wartość NULL (bazodanową) jeśli param1 jest równe param2. Przykładowo:*

```
-- na formatce pobierane są parametry z listy i pola tekstowego.
-- aby użyć to w przypisaniu do komendy "Ustaw wartość cechy" należy wpisać:
SQL::SELECT COALESCE(NULLIF('{$LISTA2}',''), NULLIF('{$TEXT}','NULL'))
```

Przypisania

Przypisania służą nadaniu wartości dla zmiennych procedury jak również nadaniu wartości atrybutom etapu którego dotyczą. Najczęściej wykorzystuje się przypisanie stanowisk wykonujących etap poprzez przypisanie do własności {stages.orgarr} tablicy (UWAGA! dane muszą być typem tablicowym, w kwerendach należy pamiętać o rzutowaniu).

Patrz przykład:

Tak więc dane wejściowe typu array o nazwie "Akceptant" zostały przypisane do własności {stages.orgarr} (czyli tablicy wykonujących zadanie workflow).

Przypisanie też możemy użyć bez konieczności pobierania danych od użytkownika, możemy je pobrać z bazy danych. Dla tego przykładu gdybyśmy chcieli pobrać Opiekuna klienta którego dotyczy sprzedaż (ze sprawy) dodalibyśmy Przypisanie własności {stages.orgarr} wartości wyrażenia SQL:

```
SELECT ARRAY[o.orunid] FROM contacts c  JOIN processes USING(contid) JOIN orgtree_view o ON o.usr_id = c.macrtk
WHERE prc_id = {processes.prc_id}
```

Przy przypisywaniu danej z danych wejściowych pobranych w etapie należy zwrócić uwagę żeby ustawić czas przypisania na KONIEC.

Własności

Własności służą do zdefiniowania dodatkowych atrybutów procedury - można je traktować jako zmienne procedury dostępne we wszystkich etapach jak również w parametrach akcji(komend).

Najczęściej zdefiniujemy własność kiedy chcemy aby nadać jej określoną wartość a później wykorzystywać np. w warunkach do sterowania przebiegiem workflow. Np. Zdefiniujmy własność "Czy jest przedpłata", którą napełnimy wartością zależną od wyniku zapytania SQL. Następnie wykorzystamy tą własność w warunku.

Kilka słów o zmiennych

W zapytaniach SQL można używać następujących wyrażeń, które zostaną zastąpione odpowiednimi wartościami:

- {PRC_ID} - prc_id sprawy której dotyczy procedura
- {DOC_ID} - doc_id dokumentu którego dotyczy procedura
- {SOP_ID} - id etapu/czynnności
- {STAGES.PTSTID} - id definicji etapu
- również zawartości obiektów podlegających workflow sprawy i dokumentu np.:
  - {processes.rspuid} - id osoby odpowiedzialnej za sprawę
  - {documents.adduid} - id osoby tworzącej dokument

W dalszej części umieszczone zostały użyteczne konstrukcje przy budowaniu workflow:

Przykłady zapytań

Trochę teorii

Tworzenie prostych procesów workflow nie wymaga dużego przygotowania, ale do tworzenia bardziej zaawansowanych modeli konieczna jest minimalna znajomość teoretycznych zasad rządzących przepływem procesów.

Podstawy teoretyczne