

# HRMS Lokalizacja interfejsu dla Płatnika

Marcin Jahn  
Oracle Polska  
e-mail: Marcin.Jahn@oracle.com

**Abstrakt.** Referat będzie tylko zaczątkiem dyskusji, jaką autor ma nadzieję wywołać swoim wystąpieniem. W pierwszej części zostanie przedstawiony zakres zadań jaki na projektantów aplikacji nałożyła reforma ZUS. Z kolei wyjaśniona zostanie koncepcja opracowana na potrzeby Oracle Applications HRMS. Pod koniec wystąpienia zaprezentowany zostanie Oracle HRMS i interfejs do Płatnika.

## 1. Nowe reformy - nowe wyzwania

Każda reforma nakłada na informatyków nowe specyficzne, wymagania. Jeśli zmienia się system podatkowy to wraz z nowymi przepisami pojawiają się zmiany w pitach. Jeśli zmienia się sposób naliczeń emerytur, chorobowego, etc. – wszystko to ma swoje odbicie, w wymaganiach jakie stawiają nam klienci – trzeba tworzyć nowe raporty, nowe struktury w bazie, tak by znaleźć jak najprostsze rozwiązanie.

Oczywiście analogicznie sytuacja ma się z Płatnikiem. Pojawienie się Płatnika na rynku wymusiło od producentów oprogramowania HRMS obsługę i raportowanie także do tego systemu. O ile słynne i zmieniające się prawie co rok pity nie wymagały właściwie odpowiedniego formatowania – urzędy podatkowe przyjmowały formularze wygenerowane przez systemy płacowe o ile tylko numeracja pól zgadzała się – to w przypadku Płatnika sytuacja skomplikowała się dość poważnie. Konieczne było stworzenie odpowiedniego formatu pliku, który dopiero w tej postaci mógł być zacytowany do interfejsu i tam weryfikowany. Dodatkowo na problemy związane z implementacją Płatnika nałożyły się jeszcze dwa istotne szczegóły wcześniej nie występujące (np. w przypadku systemu podatkowego) - ilość dokumentów generowanych przez system była znacznie większa (komplet to 19 sztuk) z czego dokumenty rozliczeniowe (6) powinny być raportowane do miesiąc. Oczywiście sam interfejs Płatnika pozwala na generację po stronie interfejsu dokumentu ZUSRMUA, zaś ZUSDRA z punktu widzenia działu płac zawiera minimalną ilość danych zbiorczych tak, że właściwie mogłyby być wypełniane ręcznie, to jednak koncepcja interfejsu z systemu HRMS powinna zawierać rozwiązanie globalne, w miarę uniezależnione od powstawania nowych dokumentów, czy nowych informacji w samych dokumentach.

## 2. Oracle HRMS a sprawa polska

Jedną z aplikacji Oracle Applications w wersji 11, obecnie najbardziej rozpowszechnionej na rynku, jest Oracle HRMS. Produkt o dość specyficznych możliwościach, obejmujący zarówno obszar kadrowy jak i płace. Nie chciałbym tutaj omawiać możliwości poszczególnych części Oracle HRMS, myślę że temat ten potraktowany zostanie szerzej na wystąpieniu poświęconemu tylko Oracle HRMS.

Jak każda z części Oracle Applications o ile ma być oferowana klientowi polskiemu, tak i HRMS przechodzi żmudny proces, który dzieli się w zasadzie na dwa etapy.

### 2.1. Lokalizacja

Najpierw produkt jest lokalizowany. Oznacza to, że w strukturach aplikacyjnych danego produktu dokonywane są takie zmiany, które pozwalają na poprawną interpretację wszystkich elementów ważnych dla polskiego klienta. Chodzi tu o polską stronę kodową (format daty, waluty) a także o wiele miejsc, w których umieszczane są kody lokalizacyjne, pozwalające, w przypadku przedsiębiorstwa wielonarodowego, na separowanie danych bez odwoływania się do np.ról

bazodanowych, perspektyw, lub zmianę on-line sposobu naliczeń wartości bez potrzeby tworzenia oddzielnych jednostek programowych.

## 2.2. Dostosowywanie aplikacji do potrzeb użytkownika

Po wykonaniu lokalizacji kolejny etap to dostosowywanie do wymagań klienta (customization). I tu można by proces ten rozbić na dwa etapy. Pierwszy etap, który następuje po zdefiniowaniu funkcjonalnych założeń już u samego klienta, polega na dostosowywaniu istniejących elementów struktur aplikacyjnych tak, by odzwierciedlić podstawowe procesy zachodzące w przedsiębiorstwie (np. tworzenie hierarchii kont). Drugi, to tworzenie raportów oraz interfejsów. O ile w pierwszej części główny wysiłek spada na ramiona konsultantów wdrożeniowych, o tyle w drugiej części procesu, nacisk położony jest głównie na zespół programistów. Oczywiście pominęliśmy proces polonizacji produktu, który odbywa się bez udziału polskiego klienta znacznie wcześniej a także instalacji aplikacji, za który odpowiada administrator bazy.

Standardowy pakiet aplikacyjny nie zawiera oczywiście wielu raportów, który klient chce mieć. Jeśli chodzi o HRMS, to oprócz interfejsu do Płatnika, wykonanych zostało jeszcze cały szereg innych raportów i zestawień, które mogą być teraz pośrednio lub bezpośrednio wykorzystywane przy oferowaniu produktu.

## 3. Jak zrobiliśmy interfejs do Płatnika

Wszystkie aspekty opisane powyżej spowodowały, że powstający interfejs musiał być odporny na ciągle zmieniające się prawodawstwo a także liczbę i ilość dokumentów przetwarzanych co miesiąc. Co więcej musiał stanowić furtkę dla następnych dokumentów. Musiał uwzględniać możliwość wprowadzenia zmian „jednym ruchem” we wszystkich dokumentach naraz. Kolejna rzecz to specyfika języka PL/SQL i samego Oracle HRMS. Dodatkowo za cel postawiliśmy sobie jak największe zautomatyzowanie interfejsu, tak by użytkownik otrzymywał końcowy wynik bez praktycznie żadnej ingerencji.

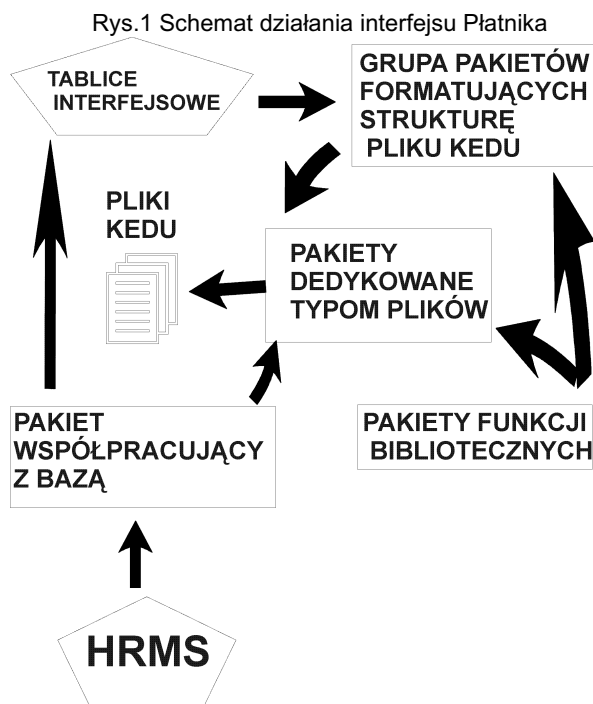
### 3.1. Struktury interfejsu

Na rysunku 1 przedstawiono schemat rozwiązania. Podstawę interfejsu stanowi tu grupa pakietów odpowiedzialnych za stworzenie pliku KEDU. Zawiera ona m.in. procedurę (zrob\_kedu), w której jest tworzony, wypełniany i zamykany plik KEDU będący wejściem do programu Płatnik. Procedura realizuje przetwarzanie poszczególnych dokumentów ZUS, będących zapisami w jednej z tabel interfejsowych (ZUS\_PARAMETRY).

Dla każdego dokumentu obsługiwanego w grupie pakietów dedykowanych typom plików, następuje pobranie informacji z tabeli przechowywującej parametry wejściowe (ZUS\_DANE) – wypełnianej przy uruchomieniu formularza. Następnie wywoływana jest procedura tworząca odpowiednią ilość bloków danego typu formularza (realizuje ona tworzenie odpowiedniego pliku). Wtedy pakiet współpracujący z bazą przekazuje dane do już istniejących struktur plików KEDU.

Tablice interfejsowe przechowują także dane archiwalne. Dane te muszą być tu gromadzone, ponieważ procedura korekty wymaga odwołania się do przesłanych danych, które w systemie mogą już nie istnieć, lub dostęp do nich może być bardzo utrudniony.

Zauważmy, że jedynym miejscem w tym rozwiązaniu zależnym od wewnętrznej struktury systemu HRMS jest jeden pakiet zajmujący się dostarczaniem danych z bazy.



### 3.2. Formularz

Jak już zostało to powiedziane formularz, który obsługuje interfejs został tak skonstruowany, aby maksymalnie uprościć pracę operatorowi. Główny ekran formularza pozwala na wprowadzenie danych informujących na jaki miesiąc ma powstać plik, wersji plików bądź pliku, daty nadania lub wypełnienia. Informacje te zostają później zapisane do tabeli, skąd pobierane są w postaci parametrów uruchomieniowych.. Druga część formularza to blok wielorekordowy, w którym wyświetlane są informacje o tym jakie osoby, bądź (w przypadku formularzy rozliczeniowych) jakie dokumenty zostaną wybrane. Czynności operatora polegają jedynie na zaznaczeniu danej osoby lub dokumentu, który ma zostać wygenerowany. Następnie po naciśnięciu przycisku „Twórz plik”, uruchamiana jest procedura tworzenia pliku bądź zestawu plików KEDU.

## 4. Propozycje

Przedstawione powyżej rozwiązanie wysoce uniezależnia kod od aplikacji HRMS. Praktycznie po zmianie jednego pakietu, który przechowuje wszystkie potrzebne kursory, jesteśmy w stanie zaimplementować nasze rozwiązanie w każdym systemie płacowym bazującym na Oracle. Chciałbym przedstawić zatem kilka propozycji do szerszego omówienia, już w ramach dyskusji.

### 4.1. Biblioteka, lista dyskusyjna, spotkania

Jest duża szansa, że tworząc bibliotekę, w której znalazłyby się rozwiązania, takie jak opisane powyżej, dotyczące z początku Płatnika jak i innych interfejsów istotnych w systemach płacowo-kadrowych (a także raportów), moglibyśmy się obronić przed ciągłymi zmianami prawa czy wersji aplikacji docelowych. Praca nie byłaby tak kosztowna. Zachowane zostałyby standardy. Oczywiście nie oznacza to, że należy zrezygnować z już napisanego kodu, być może jest on podobnie stworzony jak opisany powyżej.

Drugą możliwością jest wydzielenie grupy dyskusyjnej, w której osoby zainteresowane mogłyby uporządkować zarówno wiedzę dotyczącą interpretacji przepisów prawnych, jak i znaleźć rozwiązania dotyczące konkretnych problemów.

Kolejna propozycja to spotkania. Otóż w Anglii istnieje zwyczaj spotykania się osób odpowiedzialnych za tworzenie takich interfejsów z użytkownikami. Wybrani użytkownicy omawiają problemy, interpretują zapisy prawne, wspólnie odbywają szkolenia. Z drugiej strony grupa programistów otrzymuje wytyczne, może podzielić się pracą.

## **5. Zakończenie**

Mam nadzieję, że przedstawiając zarówno sam interfejs jak i problemy, z którymi spotyka się każdy programista, nie wyczerpałem wszystkich tematów. Przedstawiłem tu oczywiście moje subiektywne doświadczenie. Liczę na to, że o ile powyższe propozycje nie wydadzą się Państwu banalne, to może rzeczywiście można by je przedyskutować. Chciałbym zatem zaprosić Państwa do dyskusji.