

6. Działanie systemu informacyjnego w sieci komputerowej.

[Wstęp]

System informacyjny to zestaw współdziałających składników w przechowywania celu i gromadzenia, udostępniania przetwarzania, informacji, aby wspomagać podejmowanie decyzji, koordynowanie, sterowanie, analizowanie i wizualizację informacji. Inaczej – iloczyn:

$$SI = U \times P \times D,$$

gdzie:

SI - system informacyjny danej organizacji,

U - użytkownicy systemu,

P - procesy informacyjne,

D - dane, baza danych.

System informatyczny to taki system informacyjny, który wspomaga funkcjonowanie firm i instytucji z wykorzystaniem infrastruktury teleinformatycznej:

$$SIT = S \times H \times N \times U \times P \times D,$$

gdzie:

SIT - system informatyczny danej organizacji,

S - oprogramowanie,

H - sprzęt komputerowy,

N - sieci komputerowe,

U, P, D - jak wyżej.

[Odpowiedź]

System informatyczny

To taki system informacyjny, który wspomaga funkcjonowanie firm i instytucji z wykorzystaniem infrastruktury teleinformatycznej.

Zadanie systemu informacyjnego (SI):

- Zaspokojenie potrzeb informacyjnych organizacji
- Wsparcie komunikacji poziomej pomiędzy różnymi komórkami organizacyjnymi
- Wsparcie komunikacji pionowej pomiędzy przełożonymi a podwładnymi
- Ułatwienie komunikacji z otoczeniem

Rozwój technologii telekomunikacyjnych i komputerowych w ostatniej dekadzie, zwłaszcza niezwykle ekspansja WWW – stworzyły nowe możliwości rozpowszechniania i wymiany informacji, a tym samym przyczyniły się do powstania „nowoczesnych technik informacyjnych”. Sam System WWW, po wprowadzeniu standardów i metod pozwalających na opis i organizację multimedialnych danych, zawartych na stronach WWW i opracowaniu systemów wspomagających procesy wyszukiwawcze - ma szansę stać się globalnym systemem informacyjnym, efektywnie wykorzystywanym przez użytkowników końcowych do wyszukiwania informacji o wysokiej trafności.

Jedną z cech powstającego społeczeństwa informacyjnego jest rozwój pojęcia telematyka, a więc niejako połączenia telekomunikacji i informatyki. Inaczej mówiąc są to sieciowe cyfrowe multimedia. Obecnie ważniejsze jest dotarcie do informacji niż jej gromadzenie, użytkownik ma możliwość bezpośredniego dostępu do informacji bez instytucji pośredniczących. Czas i przestrzeń dzięki połączeniom sieciowym nie odgrywają już takiej roli, jak niegdyś. Ważna jest także sprawa aktualności danych. W przypadku dokumentów drukowanych istnieje niebezpieczeństwo ich dezaktualizacji już w momencie opublikowania. Dokumenty elektroniczne w sieciach dostępne są w trybie czasu rzeczywistego. Poza tym istnieje możliwość aktualizowania danych na bieżąco.

Najbliższym przykładem jest organizacja dostępu do zasobów Internetu, gdzie:

rolę serwera pełni serwer WWW,
rolę klienta pełni przeglądarka internetowa.

Przeglądając strony internetowe komputer użytkownika jest klientem, a komputery które obsługują bazy danych i inne aplikacje potrzebne do obsługi połączenia to serwer. W momencie gdy przeglądarka żąda jakiejś strony, serwer wyszukuje odpowiednie informacje w bazie danych, przetwarza je do postaci strony internetowej, a następnie wysyła do klienta

so Model-View-Controller must be replaced with Resource-Method-Representation to build RESTful Services and Apps...

the mapping is (more or less):

resource -> "a model with a bit of controller thrown in"

method -> MVC controller

representation -> "like a view in MVC"

- so why RMR?

- "It (MVC) just doesn't model resources, the fundamental element of the Web is totally ignored"

Model-View-Controller (pol. Model-Widok-Kontroler) to architektoniczny wzorec projektowy w informatyce do organizowania struktury aplikacji posiadających graficzne interfejsy użytkownika[1]. Wiele prac traktuje go jako pojedynczy wzorec, lecz może on być także traktowany jako złożony wzorec wykorzystujący idee wzorców prostych takich, jak Obserwator, Strategia czy Kompozyt[1][2]. Oba te podejścia nie wykluczają się[1]. MVC nie był traktowany jako samodzielny wzorec również w pracy „Design Patterns: Elements of Reusable Object-Oriented Software” autorstwa „Bandy Czworga”[2].

Model-View-Controller zakłada podział aplikacji na trzy główne części[3][4]:

Model – jest pewną reprezentacją problemu bądź logiki aplikacji.

Widok – opisuje, jak wyświetlić pewną część modelu w ramach interfejsu użytkownika. Może składać się z podwidoków odpowiedzialnych za mniejsze części interfejsu.

Kontroler – przyjmuje dane wejściowe od użytkownika i reaguje na jego poczynania, zarządzając aktualizacje modelu oraz odświeżenie widoków.

Wszystkie trzy części są ze sobą wzajemnie połączone[1][3].