

Σύγχρονα εργαλεία και τεχνολογίες ανάπτυξης I.S. – Το Microsoft .NET

Δημήτριος Παπαδημητρίου



Microsoft .NET

Πλατφόρμα επικοινωνίας ανθρώπων,
συστημάτων και συσκευών

.NET Framework

Πλαίσιο ανάπτυξης λογισμικού με έμφαση

- στη χρήση Web Services, XML
- στον σαφή διαχωρισμό των τεχνολογιών
- στην ταχύτητα ανάπτυξης

Δυο κατηγορίες τεχνολογιών

- Εσωτερικές τεχνολογίες της Microsoft
 - **.NET Framework** (Κορμός του .NET)
 - **VB 6 ⇒ VB 7** (C#, J#, Pascal, COBOL, FORTRAN κ.α.)
 - **ASP ⇒ ASP.NET** (Εφαρμογές Internet)
 - **ADO ⇒ ADO.NET** (Πρόσβαση σε Β.Δ.)
- Τεχνολογίες δια-λειτουργικότητας
 - **WEB Services** (Πρωτόκολλο επικοινωνίας Υψ.Επιπ.)
 - **XML και XML Schema** (Αναπαράσταση Δεδομένων)
 - **Active Directory** (Οργάνωση δεδομένων Intranet)

Common Language Runtime

- Πλαίσιο εκτέλεσης των εφαρμογών, κοινό για όλες τις γλώσσες (VB, C#, J# κλπ.)
- Έλεγχος του κώδικα (ασφάλεια, error handling, εγκατάσταση και εκδόσεις)
- Εκτέλεση προγραμμάτων σε διάφορες πλατφόρμες υλικού (hardware)

Class Libraries

- Βιβλιοθήκες – Assemblies με όλες τις ενσωματωμένες λειτουργίες του Framework (data, web, windows κλπ.)
- Περιέχουν σύνολο συναρτήσεων και σταθερών
- References

.NET Assemblies - Παραδείγματα

- System.Net
- System.IO
- System.Data
- System.DirectoryServices
- System.Threading
- System.XML

.NET Framework - Χρήση

- .NET Framework Redistributable
 - Έκδοση Desktop
 - Έκδοση Compact
- .NET Framework SDK
- Δωρεάν διανομή και χρήση για ανάπτυξη



Visual Basic 7

- Αναβάθμιση κώδικα – CLS
- Inheritance, Free Threading, Garbage Collection
- COM Interoperability
- Ίδιες λειτουργίες με C# και J#

ADO.NET

3 Επίπεδα:

- Πηγή δεδομένων (βάσεις δεδομένων, αρχεία XML)
- Πρόσβαση (Data Adapter)
- In-memory διαχείριση (Data Sets)

Data Adapters

- Μεταφέρουν δεδομένων από και προς την πηγή δεδομένων (SQL, Oracle, OLEDB, ODBC)
- Αυτόματη εκτέλεση UPDATE, INSERT, DELETE
- Δυνατότητα μαζικής ή τμηματικής ανάγνωσης δεδομένων

DataSet

- In-Memory διαχείριση δεδομένων
- Πολλοί πίνακες δεδομένων σε ένα αντικείμενο
- Ορισμός συσχετίσεων
- Υπολογισμοί – DataView – In-memory JOIN
- Εξαγωγή XML και XML Schema

In-Memory RDBMS

Web Services

- Σύνολο πρωτοκόλλων υψηλού επιπέδου για την επικοινωνία εφαρμογών
- Προσανατολισμός για χρήση στο Internet ανεξάρτητα από την πλατφόρμα εκτέλεσης
- Τυποποίηση δυνατοτήτων διάφορων τεχνολογιών DCOM, RMI, Corba
- W3C

Πρωτόκολλα

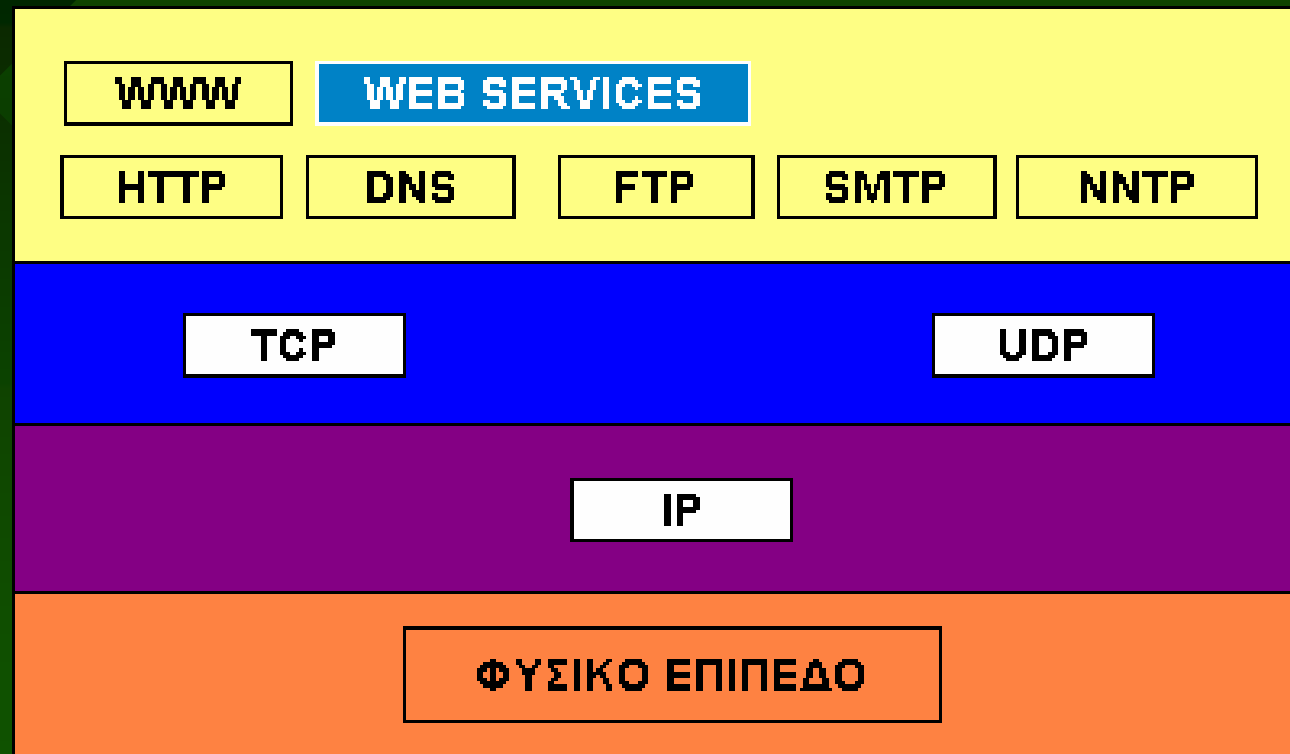


- UDDI
- WSDL, XML Schema
- SOAP, XML
- HTTP, SMTP κ.α.
- Αρχεία - Διανομή
- Type Libraries
- Binary Methods
- Platform Dependent

Υλοποίηση στο Visual Studio

- Αυτόματη υλοποίηση όλων των πρωτοκόλλων με μερικές γραμμές κώδικα
- Υλοποίηση με οποιαδήποτε γλώσσα
- Χρήση από οποιαδήποτε πλατφόρμα

Η θέση των Web Services



Αναπαράσταση Δεδομένων – XML Schema

- Περιγραφή δεδομένων XML (metadata)
- Χρήση σε οποιαδήποτε πλατφόρμα
- Βάση για Web Services και ADO.NET
- W3C

XML Schema στο VS.NET

The screenshot displays the XML Schema design environment in Visual Studio .NET. On the left, a schema diagram shows a **PurchaseOrder** element containing three child elements: **shipTo** (type **USAddress**), **ref=commen**, and **orderDate** (type **date**). The **orderDate** element is selected. Below it, the **USAddress** schema is expanded, showing elements: **name** (string), **street** (string), **city** (string), **state** (string), **zip** (decimal), and **country** (NMTOKEN). On the right, the **Properties** window is open for the selected **orderDate** element, showing its configuration as an **<xs:attribute>**. The **Dataset** section is empty. The **Misc** section shows the **name** property set to **orderDate**, the **type** property set to **date**, and the **use** property set to **optional**. A dropdown menu is open for the **use** property, showing options: (Default), optional, prohibited, and required.

xml

14/3/2008



- Εξέλιξη της ASP ;
- Δυνατότητες ανάπτυξης συγκρινόμενες με Windows εφαρμογές
- Απόκρυψη της HTML
- Χρήση έτοιμων Server Controls

Active Directory

- Οργάνωση πληροφοριών σε ένα σύστημα υπολογιστών
- Διασύνδεση εφαρμογών με κοινόχρηστα δεδομένα
- Κεντρική διαχείριση χρηστών και ασφάλειας των εφαρμογών

Γιατί είναι σημαντικό

- Αυξανόμενες εφαρμογές σε μια εταιρεία
- Πολλαπλά αντίγραφα των ίδιων πληροφοριών
- Αυξημένο κόστος συντήρησης

Εφαρμογή - CertServer

- Χρήση Web Services, Ψηφιακών Υπογραφών και XML
- Αυτόματη ανταλλαγή πιστοποιητικών
- Εξασφάλιση εγκυρότητας εγγράφων

Συμπεράσματα

- Καλύτερο πλαίσιο ανάπτυξης
- Καλύτερη πλαίσιο επικοινωνίας
- Δια-λειτουργικότητα – Το μεγάλο ζητούμενο