Şabloane e-business

Ciprian Dobre
ciprian.dobre@cs.pub.ro

Şabloane

- Şablon
  - descrie o problemă cu repetări multiple în practică, într-un anumit context
  - descrie o soluționare ce a fost validată prin utilizări repetate în practică
- Reutilizare
  - clase şi obiecte multiple, între care există relații şi un comportament global ce pot fi reutilizate
- Şabloanele sunt folosite în implementare, dar şi pentru activitățile de analiză şi proiectare

Definiții

- Un şablon orientat-obiect reprezintă un model format dintr-un număr de clase ce lucrează împreună în rezolvarea unei probleme din domeniul tehnic sau economic.
- Un şablon de proiectare este un şablon orientat obiect ce descrie o soluție pentru rezolvarea unei probleme de proiectare.
- Un şablon de analiză este un şablon orientat obiect ce descrie o soluție pentru o problemă de business/analiză.

Fundamente teoretice (1)

- În dezvoltarea unei aplicații informatice se pot întâlni situații asemănătoare celor întâlnite anterior
  - indicată **reutilizarea** unor soluții deja aplicate şi validate în practică
- De obicei există numeroase detalii care **diferențiază** problemele deja rezolvate de situațiile curente, asemănarea existând doar în punctele esențiale

Fundamente teoretice (2)

- Un ***şablon*** reprezintă ***abstractizarea*** unei probleme
  - este un model de soluționare pentru o problemă apărută într-un anumit context, verificat în practică

- Un şablon este descris prin:
- nume
  - fiecare este identificat printr-un nume unic
- scop
  - problema pe care trebuie să o rezolve un şablon
- soluționarea
  - modul de soluționare propus de şablon pentru problemă
- participanți şi colaboratori
  - entitățile implicate în şablon
- consecințe şi implementare

Fundamente teoretice (3)

- Un şablon trebuie să se refere la o problemă întâlnită în mod repetat
- Soluționarea trebuie să fie o rezolvare efectivă a problemei şi nu numai formulări de principii sau considerații teoretice
- Aplicarea unui şablon trebuie făcută numai în contextul pentru care a fost definit

- Şabloanele pot fi combinate în diverse moduri în vederea soluționării de probleme complexe

Motivele utilizării şabloanelor

- Reutilizarea soluțiilor
- Stabilirea unei terminologii comune
- O perspectivă de nivel înalt (de abstractizare) a problemei cât şi a procesului de proiectare/analiză
- Prin reutilizarea soluțiilor se beneficiază şi se învață din experiența altora
- Nu mai este necesară reinventarea de noi soluții pentru rezolvarea unor situații comune deja întâlnite în practică
- Comunicarea şi lucrul în echipă necesită un vocabular şi un punct de vedere comun asupra problemei de rezolvat, lucru permis de şabloanele de proiectare/analiză ce furnizează un punct de vedere comun în timpul etapelor de analiză şi proiectare a unui soft

Avantaje ale şabloanelor (1)

- *Îmbunătățirea comunicării în echipă şi a studiului individual*
  - Acest lucru se datorează membrilor tineri ai echipei, motivați de cei experimentați, să învețe lucruri noi şi să aplice instrumentele puternice cunoscute de aceştia.
- *Îmbunătățirea posibilității de modificare a codului program*
  - Motivul este faptul că întotdeauna există factorul timp, hotărâtor în orice proces de dezvoltare de soluții software, iar aplicarea şabloanelor permite modificarea mai rapidă a produsului software în timp util, cu minim de efort.
- *Şabloanele ilustrează principiile de bază ale abordării orientate-obiect*
  - De obicei sunt utile în creşterea înțelegerii principiilor obiectuale de bază.

Avantaje ale şabloanelor (2)

- *Facilitarea procesului de proiectare şi de analiză*
  - *Oferă o alternativă față de ierarhiile laborioase de clase*.
  - Aplicarea şabloanelor permite dezvoltarea de soluții complexe fără ierarhii vaste de clase.
- *Adoptarea de strategii îmbunătățite în proiectare şi analiză*
  - Câteva strategii sugerate sunt: proiectarea interfețelor, practicarea compunerii față de ierarhia claselor, identificarea parametrilor ce variază în contextul problemei urmată de încapsularea acestora.
- *Îmbunătățirea productivității proceselor de dezvoltare software, creşterea consistenței între aplicații*
  - datorate reutilizării, cât şi aplicării repetate.

Avantaje ale şabloanelor (3)

- *Oferă mai mult decât o simplă reutilizare de cod sursă*
  - datorat nivelului înalt de abstractizare al reutilizării, astfel putându-se aplica pe orice platformă sistem.
- *Se pot combina în rezolvarea unor probleme mai dificile, complexe*
- *S-au bucurat de o largă acceptare, numărul lor a crescut exponențial*
  - exprimă o realitate şi furnizează suportul necesar reducerii efortului de dezvoltare de aplicații software mai mult decât orice carte, lucrare sau site Web de specialitate.

Dezavantaje ale şabloanelor

- *Necesită învățarea unui număr foarte mare de şabloane*
  - Implică multă muncă în cunoaşterea acestora în vederea aplicării corecte.
- *Sindromul NIH (not-invented-here)*
  - Există neîncrederea în munca altora, de fapt pentru persoanele pentru care ceea ce nu este făcut personal nu reprezintă un lucru bun.
- *Nu includ cod sursă*, cu excepția şabloanelor de proiectare.
- *Şabloanele pot deveni un domeniu haotic/superficial*
  - Cu cât se va dovedi sau înțelege mai mult valoarea acestora tot mai multe persoane vor începe exploatarea lor în vederea creşterii vânzărilor şi mai puțin în vederea utilității şi aplicabilității.

Obiecte (1)

- În mod tradițional obiectele conțin date şi metode
- Noua perspectivă a şabloanelor defineşte un obiect ca o entitate cu responsabilități
- Noua vedere a obiectelor:
  - permite focalizarea asupra a ceea ce pot face obiectele, nu numai asupra simplei lor modalități de implementare
  - permite o mai bună selecție şi definire a obiectelor
  - definirea obiectelor este mai flexibilă
  - focalizarea pe ceea ce fac obiectele permite aplicarea moştenirii diferit din punct de vedere al comportamentului acestora
  - gândirea în termeni de responsabilități facilitează definirea interfeței publice a obiectelor
  - unui obiect cu responsabilități i se poate cere îmbunătățirea acestora, dar informația pentru care un obiect este responsabil nu este în interiorul acestuia, nu implică nimic din interiorul obiectului.

Obiecte (2)

- *Încapsularea* - modul de ascundere a datelor obiectelor
  - aceasta se poate aplica şi pentru metode, subclase sau alte obiecte
- *Moştenirea* – exprimă relațiile de generalizare / specializare dintre clase
  - reutilizarea se obține, de obicei, prin crearea de clase, urmată de derivarea de noi clase din clasele de bază
- Din perspectiva şabloanelor moştenirea se aplică astfel: identificarea variațiilor şi încapsulare lor
  - Primul pas constă în identificarea elementelor variabile în procesul de proiectare
    - metoda total opusă celei tradiționale în care se considera variabil ceea ce produce schimbare în proiectare față de ceea ce se poate schimba fără reproiectare
  - Urmează încapsularea acestora, pasul II
    - de fapt, multe şabloane folosesc încapsularea pentru a crea straturi de obiecte slab cuplate

Abordarea obiectuală din perspectiva şabloanelor

- Din perspectiva şabloanelor într-un proces de dezvoltare software se parcurg următorii paşi:

1. Identificarea şabloanelor corespunzătoare domeniului problemei de rezolvat.
2. Pentru setul inițial de şabloane identificat se parcurg următorii paşi de analiză:
   a. se alege şablonul care furnizează, respectiv creează, cea mai mare parte din contextul celorlalte şabloane;
   b. se aplică acest şablon la nivel conceptual global;
   c. se identifică şi alte şabloane suplimentare ce se vor adăuga la setul inițial de şabloane analizat;
   d. se repetă analiza pentru noul set de şabloane identificat.
3. Se adaugă detaliile necesare proiectării, se extind prin metode şi clase.

Alegerea şablonului (2a)

- În alegerea şablonului se analizează câte o pereche de şabloane ce se pot aplica o dată
  - întotdeauna un şablon existent într-un sistem stabileşte relații cu celelalte existente, prin crearea unui context pentru acestea

- În etapa de analiză se începe cu identificarea relațiilor dintre şabloane, cât şi a modul în care se stabilesc aceste relații
  - Acestea conduc la identificarea contextului în care se pot aplica şabloanele, dar şi a celor ce creează context pentru alte şabloane.

Alegerea şablonului (2a)

- Un concept folosit în etapa de analiză este şablonul *seniormost* – acel şablon ce impune constrângeri asupra a ceea ce vor face celelalte şabloane.
  - Există un şablon care defineşte modul de comportament al altor şabloane?
  - Există două şabloane care se influenţează reciproc?
  - Se poate defini un şablon fără altul, există un şablon ce are nevoie de un altul?

Metoda de realizare matriceală a sistemelor informatice din perspectiva şabloanelor

- Analysis Matrix - metodă de analiză a unei aplicații e-business
  - permite identificarea, coordonarea şi regăsirea elementelor variabile în concepte obiectuale din perspectiva şabloanelor
- Paşii necesari acestei metode sunt:
  - Identificarea celor mai importante caracteristici, cerințe, pentru un caz de utilizare şi organizarea acestora într-o matrice bidimensională.
  - Procesarea celorlalte cazuri de utilizare urmată de extinderea matricei, dacă este cazul. Fiecare caz se procesează independent de celelalte.
  - Extinderea metodei matriceale cu noi concepte.
  - Liniile matricei identifică regulile problemei.
  - Coloanele identifică cazurile speciale.
  - Identificarea şabloanelor.
  - Dezvoltarea unui model global pentru sistemul informatic.

Observații (1)

- Construind matricea bidimensională pentru celelalte cazuri din problemă (respectiv cazul 2, apoi cazul 3, …) se identifică lipsurile şi inconsistența în informații
- Beneficiarii aplicațiilor software ridică cel mai adesea probleme.
  - Ei îşi cunosc cel mai bine problema de rezolvat dar din păcate nu gândesc la nivel conceptual ci pe cazuri particulare, folosesc termenul de *întotdeauna* drept similar al termenului *de obicei*, sau *niciodată* față de *rareori*.
- Toate aceste lucruri duc la acele inconsistențe ce trebuiesc identificate încă din etapa de analiză.

Observații (2)

- Fiecare linie a matricei bidimensionale reprezintă un mod specific de implementare a conceptelor generale identificate din analiză.
- Fiecare coloană reprezintă un mod specific de implementare a cazurilor de utilizare identificate.
- Pentru fiecare linie sau coloană se identifică un şablon sau o clasă de obiecte, apoi se elaborează un model de proiectare a sistemului de realizat la nivel global, conceptual.

Observații (3)

- Aplicabilitatea acestei metode se justifică în special, în cazul problemelor complexe, cu multe cazuri specifice.
- Se analizează fiecare caz în parte, se identifică caracteristicile şi comportamentul comun tuturor cazurilor, după care se construieşte matricea de analiză care permite implementarea soluției software prin şabloane.
- Această metodă permite capturarea tuturor aspectelor particulare ale unei probleme.

Şabloane pentru aplicații de comerț electronic tip B2B

- *Şabloanele de analiză* (business pattern)
  - au un caracter conceptual şi sunt utilizate pentru a crea modele adecvate şi flexibile care să descrie comportamentul proceselor de afaceri studiate.
- *Şabloanele de proiectare*
  - propun structuri şi relații care asigură rezolvarea, în contexte bine definite, a unor probleme de proiectare.

Şabloane de analiză

- Şabloanele de analiză descriu o soluție pentru probleme uzuale ce aparțin domeniului de afaceri/analiză a unei aplicații
  - se caracterizează prin mai multă specificitate decât cele de proiectare, descriind o parte din domeniul de afaceri studiat
- În această categorie se includ următoarele şabloane: Item-Item Description, Business Entity, Contact Point, Place şi Shipping/Billing.

Exemple de şabloane de analiză

- *Item-Item Description*
  - Este unul dintre cele mai uzuale şi utile şabloane de analiză fiind descris drept colecția de obiecte ce împarte aceeaşi descriere dar cu instanțe diferite.
- *Business Entity*
  - Uzual, orice organizație dezvoltă relații cu alți parteneri de afaceri, persoane fizice sau juridice. Un şablon tip Business Entity descrie relațiile dintre diferite tipuri de organizații şi partenerii săi de afaceri.

Exemple de şabloane de analiză

- *Contact Point*
  - Şablonul tip Contact Point descrie modul de interacțiune organizație-parteneri de afaceri.
  - Subclasele acestui şablon trebuie să descrie cel puțin două lucruri: modul de transmitere de informații de la organizație la parteneri, cât şi modul de catalogare al acestora (label info).
- *Shipping/Billing*
  - În general, în orice aplicație de comerț electronic tip B2B este necesară livrarea către client a produselor/serviciilor sau informațiilor, cât şi facturilor întocmite, atât fizic cât şi electronic.
  - Şablonul Shipping/Billing descrie tocmai modul de transmitere al acestora către client.

Exemple de şabloane de analiză

- *Place*
  - Majoritatea organizațiilor implicate într-o afacere online au nevoie de urmărirea zonelor în care s-au dezvoltat procesele de afaceri.
  - În cazul firmelor internaționale se impune şi cunoaşterea orelor, perioada vacanțelor, legile privind taxele, etc.
  - Şablonul Place descrie modul de urmărire a tuturor acestor informații utile.

Şabloane de proiectare

- Şabloanele de proiectare descriu o soluție pentru problemele uzuale ce apar în etapa de proiectare a sistemelor informatice
- Cele mai întâlnite sunt *Singleton*, *Proxy* şi *State*

- „Gang of Four” (GoF) - *Design Patterns: Elements of Reusable Object-Oriented Software*, Gamma, Helm, Johnson, and Vlissides.
- Ian Shalloway, James R. Trott, *Pattern Oriented Design:Using Design Patterns From Analysis to Implementation*, Addison-Wesley, 2000

Exemple de şabloane de proiectare

- *Singleton*
  - Acest şablon este de tip creațional şi asigură o singură instanțiere, în orice moment, pentru o clasă.
  - Şablonul Singleton poate fi aplicat în cazul profilelor clienților unde este necesară menținerea informațiilor de bază ale acestora (nume, număr de telefon, etc.).

Exemple de şabloane de proiectare

- *Proxy*
  - Şablonul Proxy indică modul de reprezentare a unui obiect ce nu este momentan în memorie.
- Să presupunem un ecran de căutare de clienți în care utilizatorul poate defini un criteriu de căutare şi obține lista clienților ce îndeplinesc respectivul criteriu.
  - Există vreo posibilitate de minimizare a volumului informațiilor trimise prin rețea?
  - Toate aceste se pot realiza printr-un şablon proxy. Aplicând acest şablon se reduce volumul informațiilor transmise prin rețea pentru că inițial se aduce doar câtă informație este necesară identificării fiecărui obiect client (nume), iar în momentul alegerii din sursă a unui client, obiectul ClientProxy aduce obiectul client din baza de date.

Exemple de şabloane de proiectare

- *State*
  - Şablonul State descrie modul de implementare al unui obiect ce-şi poate schimba comportamentul la schimbarea stării lui interne.
  - Ideea este de a construi un obiect ca o colecție de mai multe obiecte stări, ce prezintă comportamente diferite cerute de obiect pentru fiecare stare în care poate fi.
  - Fiecare comportament ce este afectat de starea internă a obiectului este implementat în obiectele tip stare ca metodă.

Aplicarea şabloanelor în soluții B2B

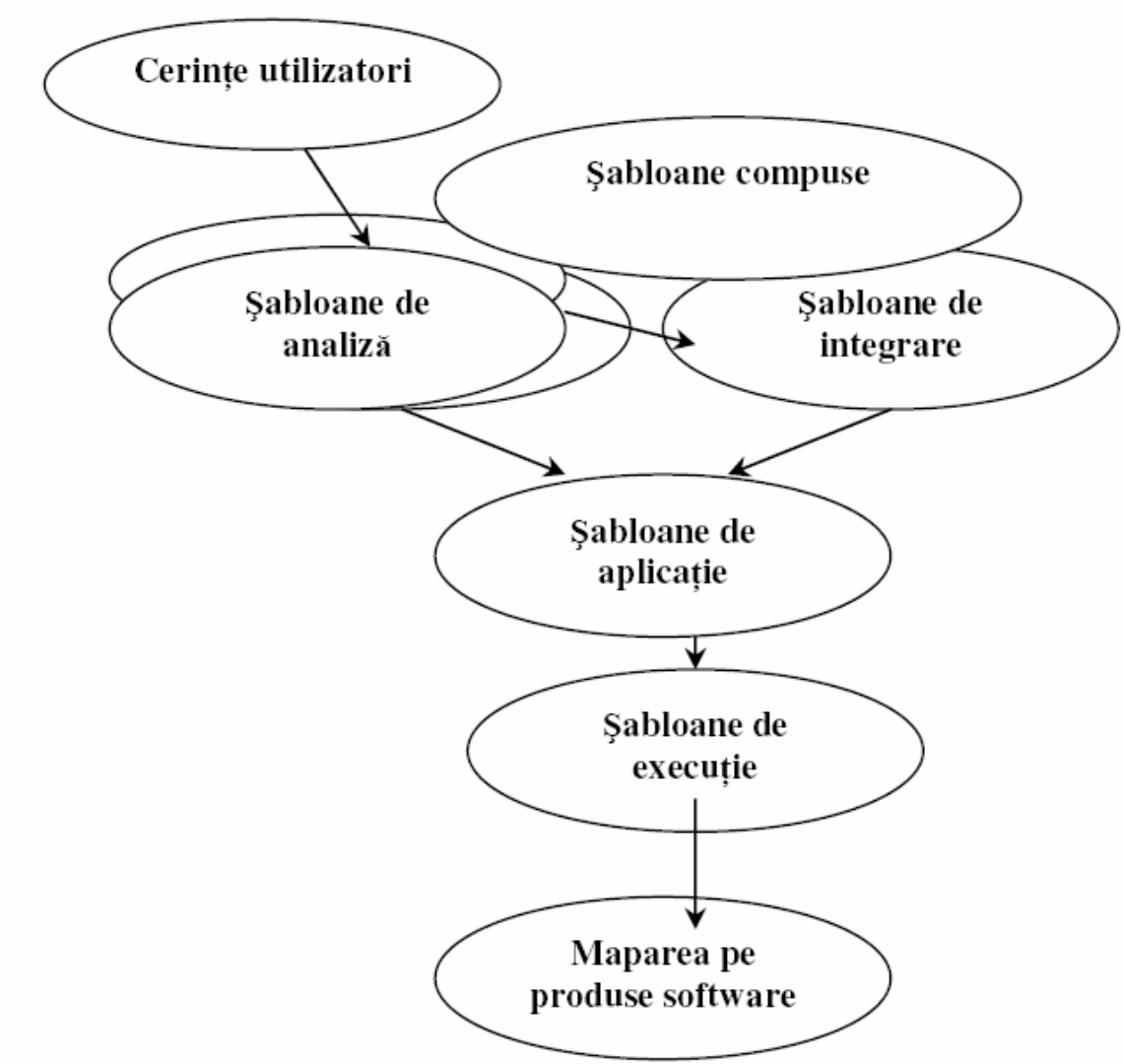
- Rolul dezvoltatorilor IT este de a evalua problema de afaceri şi de a construi soluții software care să o rezolve
  - Necesită timp
  - Soluția: utilizarea şi aplicarea experienței arhitecților IT din domeniu
- Pentru a putea captura experiența dobândită de alți experți IT trebuie creat un depozit de valori, de soluții deja dezvoltate în practică, pe baza cărora se pot dezvolta soluții noi
- Această tehnică a reutilizării implică câştig de timp, bani şi efort iar într-un proces de dezvoltare soluții IT, facilitează obținerea unei soluții solide, funcționabile.

Şabloanele e-business IBM

- Aceste şabloane permit implementarea cu succes a soluțiilor B2B aplicând principiul reutilizării componentelor

- Abordarea din perspectiva şabloanelor se bazează pe un model structurat pe straturi de elemente ce pot fi exploatate şi aplicate în orice metodologie de realizare de sisteme informatice
  - Fiecare strat este astfel structurat încât detaliile fiecăruia se construiesc pe baza stratului anterior.

Modelul structurat de şabloane pentru aplicaţii e-business



Şabloanele e-business IBM

- Modelul include:
  - şabloane de analiză (business patterns) - identifică interacţiile dintre utilizatori, date şi reguli de afaceri
  - şabloane de integrare (integration patterns) - realizează legătura dintre şabloanele de afaceri atunci când o soluţie B2B nu poate fi descrisă printr-un singur şablon de afaceri, ajută la combinarea şabloanelor business pentru aplicaţii complexe
  - şabloane compuse (composite patterns) - reprezintă combinaţia dintre primele două straturi de şabloane
  - şabloane de aplicaţie (application patterns) - descrie, la un nivel conceptual, interacţiile dintre aplicaţii (reguli de afaceri) şi date din cadrul primelor două categorii de şabloane

Şabloanele e-business IBM

- Modelul include:
  - şabloane de execuție (runtime patterns) - descrie structura logică, de middleware, suportată de şabloanele tip aplicație, înfățişând nodurile majore, rolurile şi interfețele dintre acestea
  - implementări software testate din punct de vedere practic (product mappings) - identifică implementările software testate pentru fiecare şablon tip runtime (execuție)
  - linii directoare (best-practice guidelines) ce ghidează proiectarea, dezvoltarea, cerințele non-funcționale, realizarea şi managementul aplicațiilor dezvoltate.

Şabloane de analiză

- Descrie relațiile dintre utilizatori şi organizatie, regulile de afaceri cât şi datele ce vor fi accesate.
- Există patru tipuri de şabloane de afaceri de bază:
- *self-service* (user-to-business): utilizatorii interacționează direct cu organizația prin Internet sau Intranet
  - Ex: aplicații Web simple
- *information aggregation* (user-to-date): utilizatorii pot extrage volume mari de informație
  - Ex: managementul cunoştințelor, inteligența afacerilor
- *collaboration* (user-to-user): Internetul suportă colaborarea dintre utilizatori
  - Ex: email, chat, video conferințe
- *extended enterprise* (business-to-business): aplicații ce fac legătura între multiplele procese de afaceri între organizații diferite
  - Ex: EDI, SCM.

Şabloane de integrare

- Permit combinarea mai multor şabloane de afaceri în vederea rezolvării unei probleme.

- Principalele şabloane de acest tip sunt:
- *access integration*: permit integrarea unui număr de servicii printr-un punct comun de acces
  - Ex: portaluri
- *application integration*: permite integrarea unui număr de aplicații şi surse de date fără cererea directă a utilizatorilor, cum ar fi fluxuri manageriale, brokeri de mesaje.

Custom Design

- Şabloanele de afaceri şi de integrare pot fi combinate pentru implementarea unei soluții e-business specifică, cum ar fi Custom Design

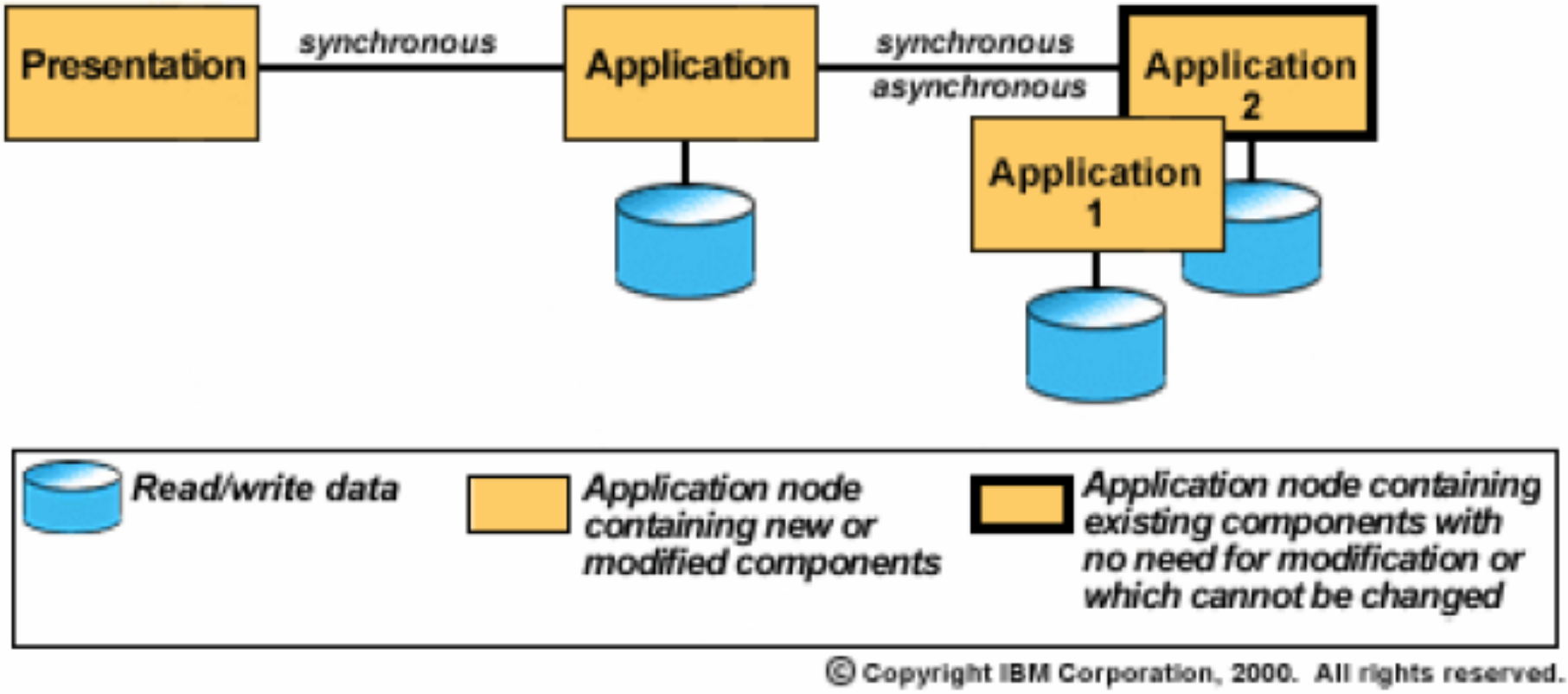
Şabloane de aplicație

- După identificarea şabloanelor business, se definesc componentele logice la nivel global şi modul lor de interacțiune, ceea ce reprezintă un şablon de aplicație.
- De obicei un şablon de analiză are multiple şabloane de aplicație.
  - poate conține componentele logice ce descriu stratul de prezentare al aplicației (presentation tier), stratul regulilor de afaceri sau stratul de backend al aplicației.

Şabloane de aplicație

- Şablonul de aplicație împarte aplicația într-un număr de componente conceptuale de bază, identificând obiectivele acesteia.
- Ex. şablonul business – *Self-Service*
  - scopul este de a permite utilizatorilor accesul la informațiile din organizație de tip back-end.
  - în acest şablon componentele conceptuale sunt: aplicația de prezentare, aplicația ce descrie regulile de afaceri şi aplicațiile existente în firmă (back-end) ce reprezintă stratul de persistență a datelor.

Self-Service::Directly Integrated Single Channel

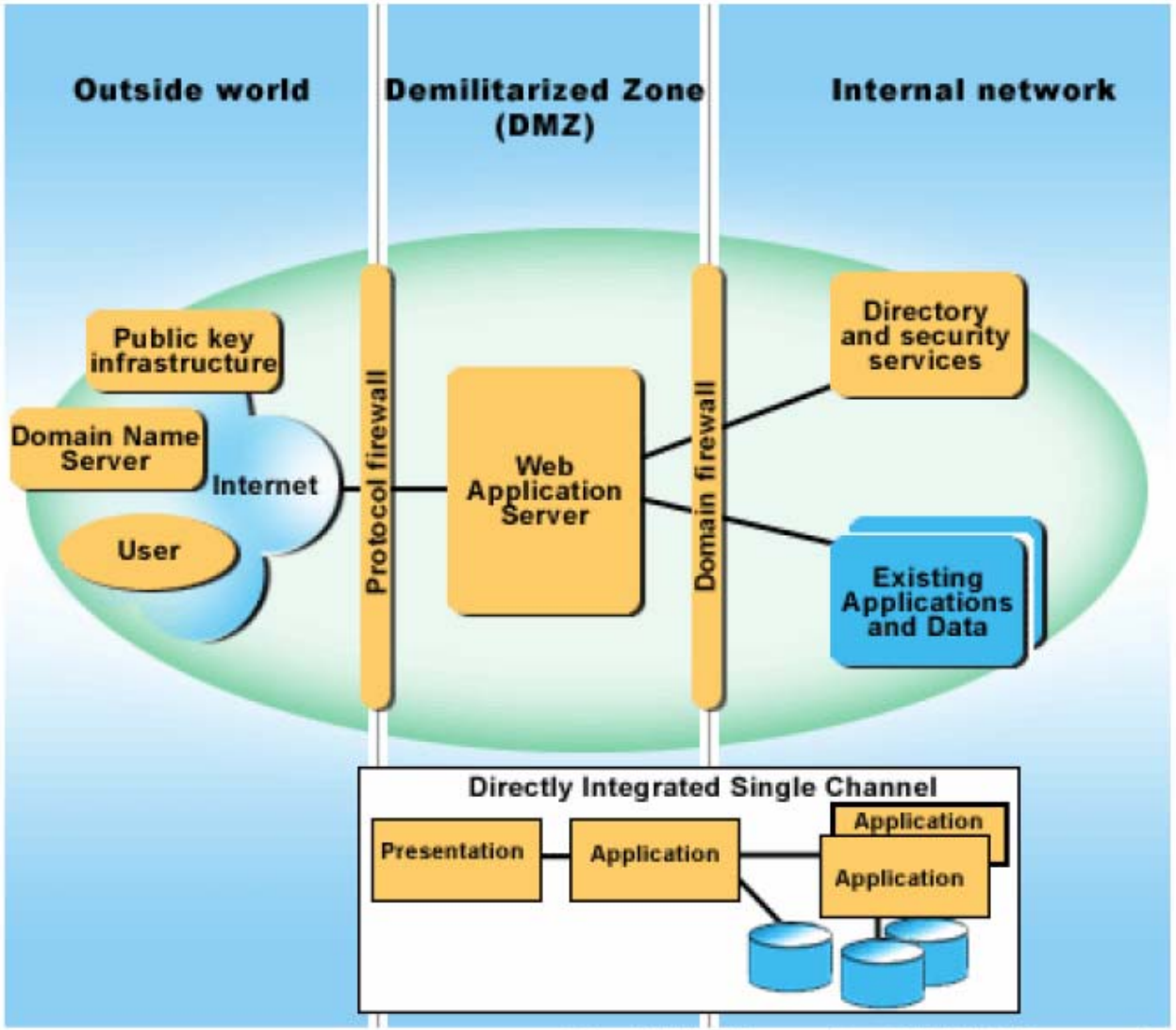


Şabloane de execuție

- În şabloanele de execuție (runtime), atenția se focalizează pe nodurile logice şi pe locul de plasare în întreaga structură a rețelei
  - Şabloanele aplicație se dezvoltă prin adăugarea de noi funcții explicite, fiecare fiind asociată unui nod de execuție ce pot exista pe maşini diferite sau pe aceeaşi maşină
- Continuând exemplul anterior, se trece la identificarea şablonului de execuție potrivit pentru şablonul business şi cel de aplicație ales

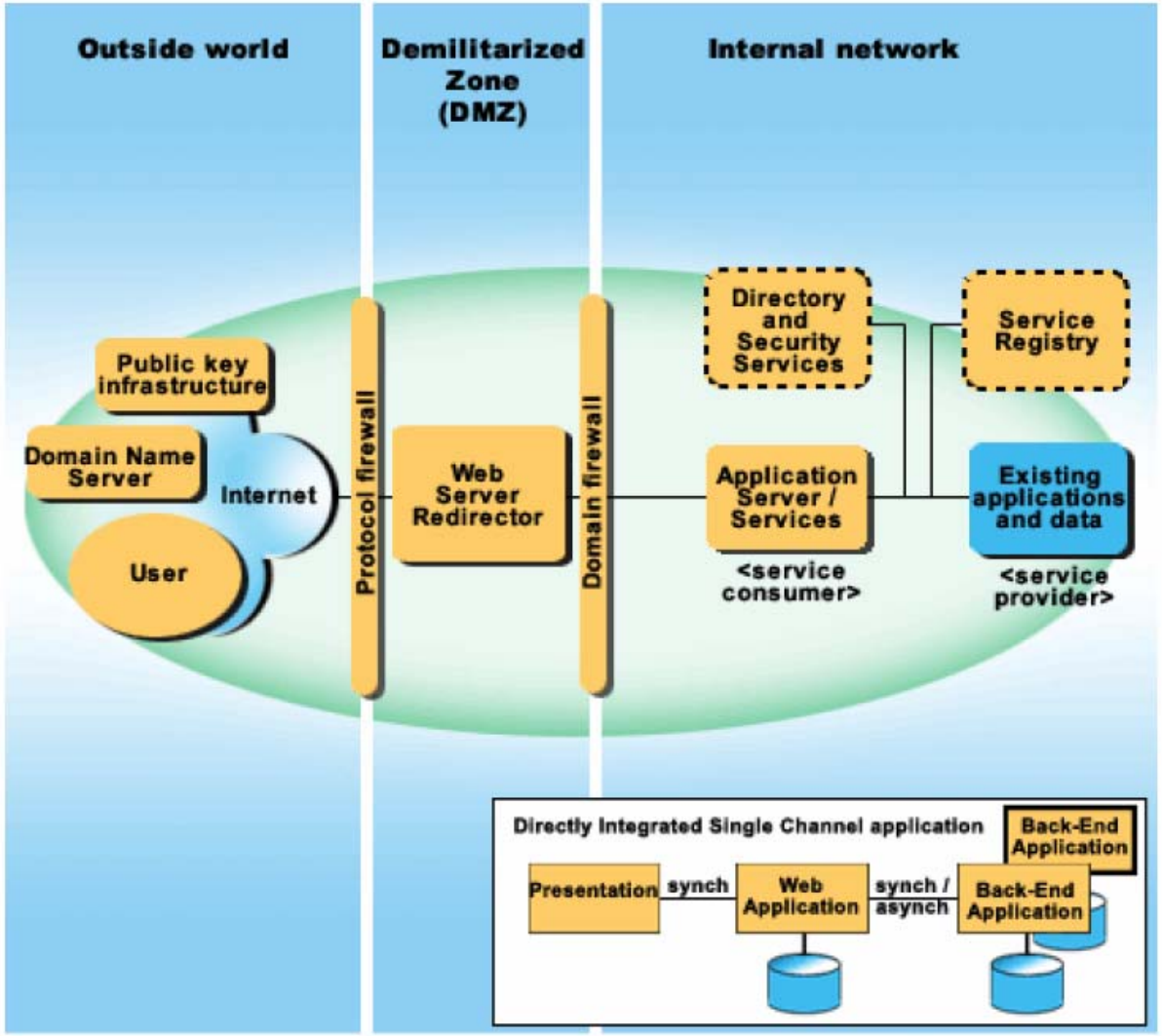
Şabloane de execuție

- Presupunând accesul utilizatorilor la datele organizației prin Internet, se impun şi anumite măsuri de securitate.



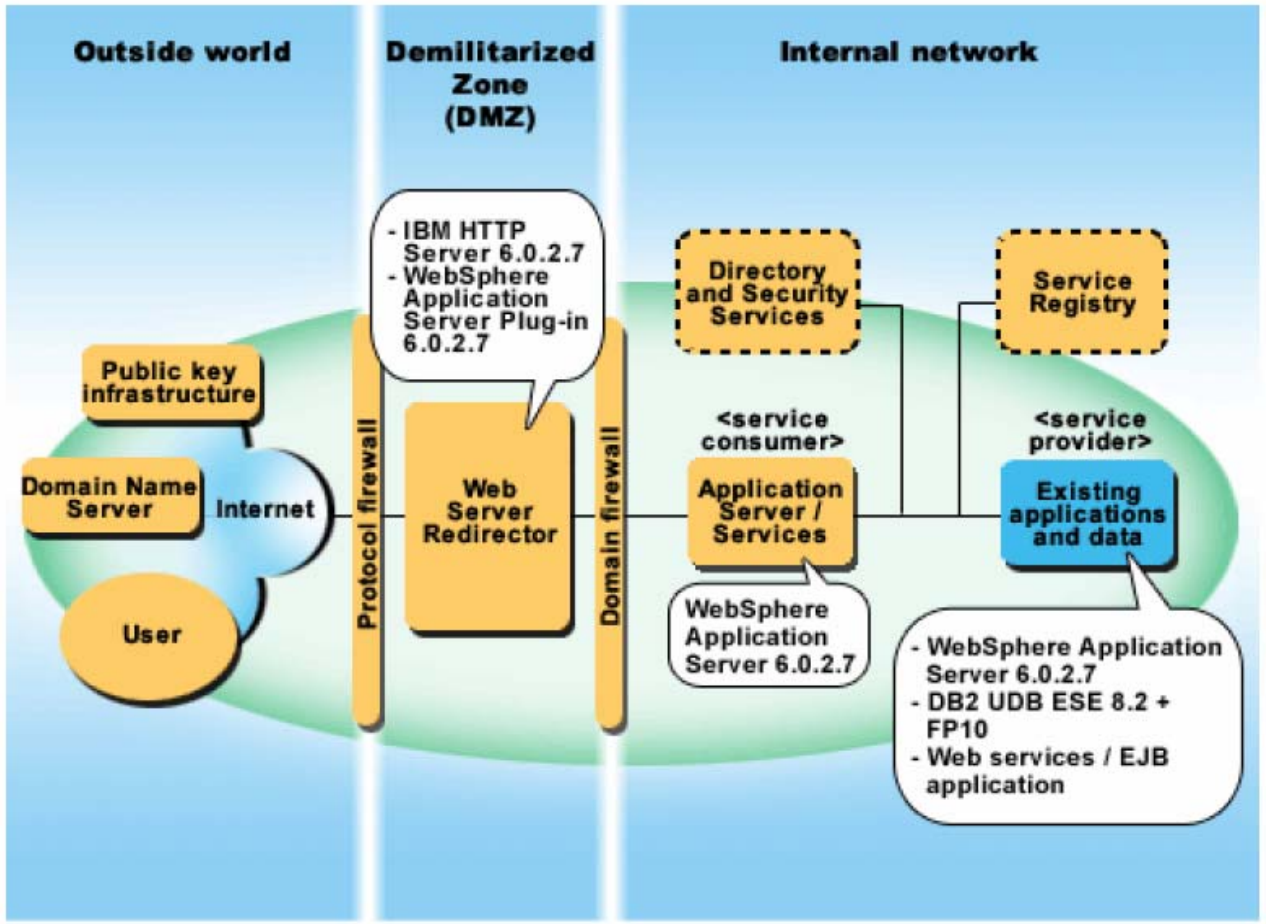
Şabloane de execuție

- O caracteristică a şablonului de execuție este plasarea serverului Web de aplicație între două firewall-uri.
- În cazul cererilor crescute de securitate există un şablon de execuție ce împarte serverul Web de aplicație în două noduri funcționale, separând funcțiile serverului HTTP de serverul de aplicație



Implementări software testate

- Ultimul pas în definirea structurii unei aplicații este corelarea produselor IT existente cu unul sau mai multe noduri logice de execuție.
- Fiecare mapare pe produse IT este orientată către o platformă software particulară funcție de preferințele beneficiarilor.



Sumar

- Şabloanele facilitează înțelegerea şi analiza problemelor complexe, cât şi divizarea lor în componente mai mici, mai uşor de administrat şi de implementat.
- Prezentarea este doar o abordare succintă a tipurilor de şabloane ce pot fi aplicate în domeniul e-business.
- Teoria poate fi studiată mai pe larg pe site-ul IBM: http://ibm.com/developerworks/patterns/