Universitatea Tehnica a Moldovei

Catedra Automatica si Tehnologii Informationale

Disciplina: SIPC

Lucrare de Control

Tema: Protocolul SOAP

A efectuat : Ion Filipski studentul(a) anul IV grupa TI 061

A verificat: lector universitar Andrei Postaru

Chisinau 2009

În raport se scrie (*Exemplu*): Scopul si sarcina lucrarii; Indicatii teoretice; Realizarea sarcinii; Concluzie; Bibliografie; Anexa.

1 Scopul si sarcina lucrarii

* 1. Scopul: Descrierea protocolului SOAP
  2. Sarcina lucrarii: Prezentarea tehnologiei SOAP, importantei, avantajelor utilizarii SOAP in comparatie cele ale utilizarii altor tehnologii similare.

1. Indicatii teoretice (numai ce se refera la elaborarea sarcinii)

2.1

2.2

1. Realizarea sarcinii (Pe pasi cum sa realizat sarcina de explicat si de aratat rezultatul)

3.1

3.2

Introducere

Odata cu aparitia retelelor, cresterea numarului calculatoarelor implicate in retea, creşterea dimensiunilor şi complexităţilor aplicaţiilor distribuite a început să crească şi complexitatea protocolurilor de comunicare.

Implementarea unei aplicatii distribuite insemna:

1. implementarea business-logicii aplicatiei,
2. elaborarea mesajelor noi de transmitere prin reţea,
3. implementărea prelucrării fiecărui mesaj, pe partea client şi pe partea server separat.

Cand creşte complexitatea aplicaţiilor, creşte şi complexitatea mesajelor. Unele mesaje pot sa fie insuficiente pentru scopul lor iniţial, şi apare necesitatea de a le modifica sau de a le extinde. Deseori mesajele sunt grupate în protocoluri, si asupra lor se aplică constrângeri globale de sintaxă. Deoarece prelucrarea unui mesaj la nivel de business logică echivalează cu apelarea unei proceduri, apare necesitatea apariţiei protocolurilor procedural orientate. Un protocol care a ajuns sa fie larg răspândit este protocolul RPC, ce se descifrează ca Remote Procedure Call. Implementarea aplicaţiilor cu utilizarea protocolului RPC se aseamânâ cu elaborarea aplicaţiilor obişnuite. Clientul execută procedura, iar serverul o executa. Şi se crează efectul apelului direct de proceduri. Iar comunicaţia reală între client şi server are loc la nivel de serviciu RPC. Comunicatia reală între aplicatie şi serviciu RPC are loc la un nivel incpsulat în surse rezultate în urma compilării declaraţiilor procedurilor. Declaraţiile procedurilor se fac la un nivel abstract, intr-un limbaj declarativ numit IDL (Interface Definition Language). Maparea limbajului declarativ IDL in limbaj anumit de elaborare a aplicatiilor este realizată de compilator special de IDL. Acest lucru permite ca aplicaţia să fie eterogena din punct de vedere al limbajelor utilizate. Serverele pot fi scrise în anumite limbaje, iar clienţii în alte limbaje. Iar implementarea aplicaţiilor are loc cu impresia că un client apelează direct prodeduri de la server, şi invers, chiar daca ele sunt scrise în limbaje totalmente diferite. Logica comunicatiei este abstractizată la nivel IDL. RPC şi-a găsit loc în multe servicii ale sistemelor operationale.

Însă în aplicaţiile orientate pe obiecte a fost necesară aparitia protocolurilor de nivel mai înalt ca RPC. Aşa au apărut protocoluri orientate pe obiecte. Asemenea protocoluri orientate pe obiecte si cu model orientat pe componente sau fara el, au apărut ulterior. Tehnologii care au obtinut răspândire mai largă sunt DCE-RPC, COM, CORBA, RMI. RMI (Remote Method Invocation) este specific limbajului Java, şi are două varietăţi principale. RMI over IIOP este bazat pe acelasi protocol pe care este bazata tehnologia CORBA. Şi RMI over RMP.

Tehnologiile orientate pe componente ActiveX şi OLE2 sunt bazate pe COM.

Neajunsurile CORBA constau în faptul ca avea portabilitate joasă la nivel de vendor, din motiv ca implementarile Basic Object Adapter (BOA) variau de la o implementare la alta. Modernizarea tehnologiei CORBA a dus la înlăturarea multor neajunsuri legate de portabilitate. Asta s-a datorat apariţiei specificaţiilor standarte Portable Object Adapter (POA). Din cauza complexităţii CORBA, specificaţiile standarte orientate pe componente au apărut abia nudemult în CORBA3, aşa numitul Corba Component (CCM).

Tehnologii orientate pe componente in CORBA au existat şi mai devreme in VisiBroker (Borland Enterprise Server), Enterprise Java Bean (EJB), însă erau vendor specific.

Însă protocoalele binare bazate pe RPC, odata cu avantajele aparute, au o serie de neajunsuri:

1. Împachetarea datelor (marshalling) în format RPC şi despachetarea datelor(unmarshalling) din format RPC trebuie să ţină cont de arhitectura procesorului calculatorului sursă si arhitectura procesorului calculatorului target. Cu toate ca dezvoltatorii aplicaţiilor nu trebuie să ţină cont neapărat de aceste nuanţe, de ele totuşi trebuie să ţină cont dezvoltatorii protocolului RPC. Iar în lipsa unor compilatoare IDL dezvoltatorii aplicatiilor oricum trebuie sa elaboreze aplicaţiile cu multa grija la arhitecturi.
2. Respectarea corectitudinii sintactice şi arhitecturale a regulilor protocolului e problematică. Unica posibilitate de asigurare ca protocolul va fi respectat, este de a presupune ca programatorul nu lucrează cu protocolul propriu zis, ci utilizează cod generat de compilatoare IDL.
3. Urmărirea problemelor aparute la apariţia anomaliilor de comunicare deasemeni e extrem de complicată.
4. Cu toate ca protocolurile RPC sunt relativ compatibile, comunicarea între RPC de la diferiţi vendori e un lucru care necesită foarte mult lucru de rutină pentru a evita extensiile incompatibile, sau pentru a le porta. Ca exemplu poate servi elaborarea aplicatiilor CORBA-COM.
5. Serviciile RPC sunt vulnerabile la atacuri prin reţea.
6. Rularea serviciilor RPC pe anumite porturi fac administrarea securităţii să fie complicată. Multe dintre aplicaţiile RPC deschid porturi noi. Deasemeni diferite servicii RPC pot utiliza diferite porturi. Din acest motiv configuratia firewall-urilor în vederea permiterii aplicatiilor sa lucreze, devine problematică.
7. Identificarea serviciilor RPC impune creare de servicii noi, precum ar fi Name Service, şi altele. Adresarea si identificarea este complicata, practic obfuscata. Necesita mult lucru de configurare. Iar configurarea depinde de vendor. De exemplu configurarea COM necesita a fi scrisa in registri. Visibroker necesita Name Service si configurarea fisierelor specifice Borland Enterprise Server.
8. Apelarea diferitor aplicatii RPC situate în reţele locale, sau/şi din reţele locale pune problema creării protocolurilor noi, exemplu ESIOP pentru CORBA. Sau se recurge la tunelling. Tunelling-ul deasemeni e un proces problematic pentru tehnologii bazate pe RPC.
9. Nu in ultimul rand arhitectura tehnologiilor existente este complicata.

În limbajul Java a aparut un protocol similar numit RMP, si tehnologia de comunicare RMI bazata pe acest protocol. Insa problema consta in faptul ca protocolul e legat de Java, si nu este interoperabil cu alte limbaje. Pentru a rezolva aceasta problema a fost creat RMI over IIOP, care este interoperabil cu limbajele pentru care exista implementare CORBA.

Tehnologia CORBA este interoperabilă, dar numai în limitele standartelor OMG. Iar fiecare producător de CORBA ca regulă dezvoltă, distribuie şi promovează seturi proprii de extensii. Respectiv producătorii recomandă utilizarea produsului lor. Interoperabilitatea cu CORBA nu este sustinută de compania Microsoft. Iar interoperabilitatea cu DCOM nu este sustinuta de companiile ce lucrează cu tehnologii non Microsoft. Interoperabilitatea CORBA-DCOM cere cunostinte detaliate a realizatiei protocolului RPC, precum şi a tehnologiilor utilizate, ceea ce contravine insuşi filozofiei utilizării acestor tehnologii.

Concluzii

Bibliografia

Anexa A .(imagini din programul realizat,listing de program)

Titlul capitol, subcapitol - se scrie cu litere minuscule (Bold)– **cu excepţia literei iniţiale**, fără punct la sfârşit. După cifra (cifrele) ce indică numărul capitolului, subcapitolului, ..., punct nu se pune. Fiecare capitol nu se începe pe pagină nouă.(se scrie una dupa alta)

***Reguli de redactare a Raportului***

Raportul se culege cu fontul Times New Roman, Normal, Justify - 12, interval – 1,5, sau cu fontul Times New Roman, Normal, Justify - 14, interval – 1;

Culoarea Fontului – Negru; Textul de bază (după cuprins) se culege cu următoarele alinieri faţă de margini (fără chenar şi inscripţii): dreapta(Righ)-10 mm; sus(Top)-20 mm; stînga, jos(Left, Buttom)-20 mm.

Capitolele, subcapitolele, punctele şi subpunctele – sunt numerotate cu cifre arabe, şi se scriu din abzaţ. Nu se scrie cuvîntul capitol.

Exemplu: 1, 2, 3, etc., sau 1.1, 1.2, 1.3, etc…

***Enumerarea paginilor in Raport***

Numărul paginii – jos, la mijloc fără punct. Foaia de titlu este inclusă în numărul de pagini.

Daca in text sunt enumerari simple acestea se numeroteaza prin liniuta, daca enumerarea are sens atunci se incepe cu litera mica, punct si virgula la sfirsit, ultima enumerare punct.

*Exemplu:*

* unu;
* doi;
* trei.

Daca enumerarea nu are sens se incepe cu litera mare, punct la sfirshit.

*Exemplu:*

* Matematica.
* Informatica.
* Istoria.

Dacă în text sunt enumerări multiple, acestea se numerotează prin litere latine (minuscule) cu paranteză (dacă enumerările conţin enumerări, acestea se numerotează prin cifre arabe cu paranteză).

*Exemplu:*

а) \_\_\_\_\_\_\_\_\_\_\_\_\_\_

b) \_\_\_\_\_\_\_\_\_\_\_\_\_\_

1) \_\_\_\_\_\_\_\_\_\_\_\_\_\_

2) \_\_\_\_\_\_\_\_\_\_\_\_\_\_

- ) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

c) \_\_\_\_\_\_\_\_\_\_\_\_\_\_

***Nu se admit următoarele semne:***

* ***1***
  + ***2***
    - ***3***
      * ***4***

***Figuri***

Către toate figurile din Raport, trebuie să fie date referinţe. Dacă figura este una, ea se numerotează ca “Figura 1” sau “Figura 1.1”. Cuvântul “figura” şi numărul ei este amplasată pe centru, fără punct la sfârşit.

Figura se plaseaza dupa explicarea ei.

*Exemplu:*

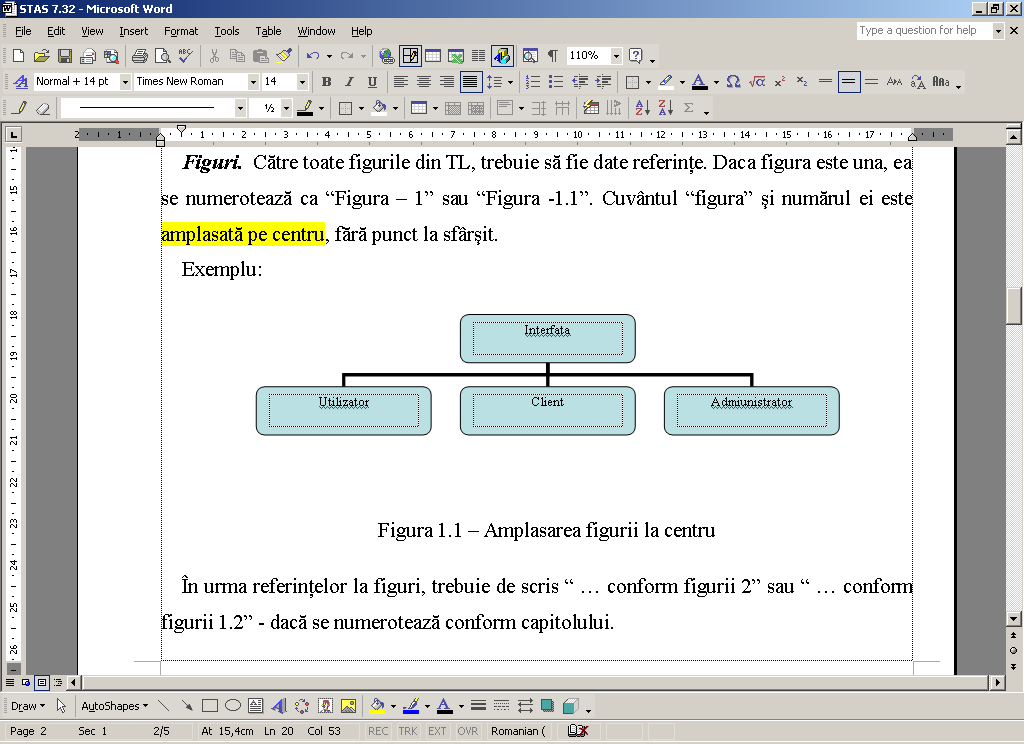


Figura 1.1 – Amplasarea figurii la centru(*denumirea figurii*)

În urma referinţelor la figuri, trebuie de scris “ … conform figurii 2” sau “ … conform figurii 1.2” – dacă se numerotează conform capitolului.

***Tabele***

Denumirea tabelei se scrie în partea dreaptă fără abzaţ, de la începutul tabelului într-un rând cu numărul tabelei.

Tabela se plaseza in urma explicarii ei.

*Exemplu:*

Tabelul 1.1 – Alinierea tabelului(*denumirea tabelului*)

|  |  |  |
| --- | --- | --- |
| **A** | **B** | **C** |
| 1232 | 4322 | 23423 |
| 2344 | 234235 | 5454346 |

Doar primul cimp in tabela se scrie cu litera mare, restul cimpurilor se scrie cu litera mica, textul din tabela se poate de facut cu fontul mai mic.

La continuarea tabelului pe altă pagină, se indică “Continuarea tabelului 1.1”,linia de jos a tabelului trebue sa fie invizibila (respectiv cea de sus). Titlurile rubricilor tabelului, în cazul divizării acestuia pe două sau mai multe pagini, se indică doar în cadrul primei pagini.

Dacă tabelele şi figurile se numerotează pe capitole şi sunt în introducere sau anexe, ele se numerotează în felul următor:

Figura I.2, Figura A.3, Tabelul B.1, Figura C.4,

unde:

Figura I.2 – figura 2 din introducere;

Figura C.4 – figura 4 din anexa C;

Tabelul B.1- tabelul 1 din anexa B.

Dacă în Raport este o singură tabelă, ea trebuie sa fie numerotată „Tabelul 1”, sau „Tabelul B.1”, dacă ea este prezentată in anexa B.

Se permite de a utiliza mărimea fontului în tabelă mai mic, decât cel din text.

***Formule***

Dacă formula nu se încadrează într-un rînd, atunci ea trebuie trecută din rând nou după semnul (=) sau după semnul (+) , (-), înmulţirea(×), împărţirea (:), sau alte semne matematice, la care semnul se repetă în noul rând. Trecerea formulei din rând nou la semnul înmulţirii, se utilizează semnul (×).

Explicaţia simbolurilor şi coeficienţilor din formule trebuie prezentate sub formulă, în acea numeraţie, cum sunt date în formulă.

Formulele sunt numerotate în formă crescătoare, cu cifre arabe în paranteze rotunde, în partea dreaptă a liniei.

***Referinţe***

În Raport se permite referinţe în documentul dat. Referinţe la subcapitole, puncte, tabele şi figuri nu se permit, în afară de cele incluse în document. Referintele trebue sa coencida si cu bibliografia.

***Concluzii***

Concluziile trebuie să corespundă cerinţelor. Concluziile nu trebuie să conţină tabele, figuri, formule, enumerări.

***Bibliografie***

Dacă la elaborare au fost folosite **materiale în formă electronică,** în bibliografie se indică denumirea materialului şi notarea:

[Resursă electronică];

[Электронный ресурс];

[Electronic resource] ----- în limba materialului bibliografic.

*Exemplu:*

Comerţul electronic [Resursă electronică]

***Anexele***

Se numerotează prin litere latine (majuscule, excluzând O şi I). Fiecare anexă se începe pe pagină nouă. Dacă este o singură anexă, ea se numerotează Anexa A . După cuvântul Anexa, alineat la centru, urmează titlul anexei.

*Exemplu:*

**Anexa A**

**Codul sursă**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Anexa B**

**Diagrame UML**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

Figura B1 – Reprezentarea schemei de dirijare a programului

**etc.**