

A dark blue vertical bar is on the left. A blue arrow points right from it, containing the date.

12.10.2017

Tema 1 - PSSC

Analiza unui sistem software (Tool de configurare
automată a pinilor unui microcontroler)

Several thin, curved lines in shades of blue and grey sweep upwards from the bottom left corner.

Emanuel Kovacs

UPT, AC, IV IS, 1.3

Cuprins

1. Prezentarea cerințelor funcționale si non-funcționale si identificarea acelor cerințe care influențează arhitectura	2
1.1 Cerințe funcționale si non-funcționale.....	2
1.1.1 Cerințe non-funcționale	2
1.1.2 Cerințe funcționale.....	2
1.2 Cerințe care influențează arhitectura	2
1.2.1 Performanța	2
1.2.2 Fiabilitate	2
1.2.3 Scalabilitata	2
2. Descompunere in componente, definirea responsabilităților componentelor si a relațiilor dintre ele	3
2.1 Componente.....	3
2.2 Definirea responsabilităților componentelor.....	3
2.3 Relațiile dintre componente.....	3
3. Prezentarea sistemului software.....	3
3.1 Arhitectura privită din perspectiva logică	4
3.2 Arhitectura privită din perspectiva proces.....	4
4. Identificarea celor mai importanți 3 indicatori de calitate, specificarea măsurii alese pentru fiecare indicator de calitate.....	6
4.1 Performanța	6
4.2 Fiabilitatea	6
4.3 Scalabilitata	6
4.4 Toleranța la modificări	6
5. Identificarea tehnologiilor middleware folosite pentru a comunica intre componente	7
6. Identificarea principalelor modele și stiluri arhitecturale folosite.....	7
6.1 Module	7
6.2 Stiluri arhitecturale.....	7
7. Prezentarea scenariilor de validare a arhitecturii	8

1. Prezentarea cerințelor funcționale si non-funcționale si identificarea acelor cerințe care influențează arhitectura

1.1 Cerințe funcționale si non-funcționale

1.1.1 Cerințe non-funcționale

Constrângeri tehnologice

- Folosirea unei baze de date SQL
- Conexiune OLE DB pentru Excel

Indicatori de calitate

- Performanța
- Scalabilitatea
- Toleranța la modificări
- Testabilitatea

1.1.2 Cerințe funcționale

Sistemul trebuie să acceseze informații (specificații) din baza de date, pe care trebuie să le prelucreze și să completeze un fișier în care se configurează pinii microcontrolerului.

1.2 Cerințe care influențează arhitectura

1.2.1 Performanța

Aplicația trebuie să aibă un timp de răspuns sub 5 secunde și trebuie să proceseze (parseze) fișierul Excel în timp mai mic de 4 secunde.

1.2.2 Fiabilitate

Pierderea datelor nu este acceptată.

1.2.3 Scalabilitate

Aplicația trebuie să fie în stare să lucreze într-un mod eficient cu un volum mare de date.

2. Descompunere in componente, definirea responsabilităților componentelor si a relațiilor dintre ele

2.1 Componente

- I. Acces date
- II. Calculator
- III. Import Excel
- IV. Export XML
- V. ReportGenerator
- VI. InfoGenerator
- VII. Database
- VIII. Dataset

2.2 Definirea responsabilităților componentelor

- Acces date – E compusă din două sub-componente (conexiune SQL si conexiune OLE DB). E responsabilă cu obținerea datelor din baza de date și cu conexiunea OLE DB pentru Excel
- Calculator – Proceasează datele necesare configurării pinilor: datele sunt modificate astfel încât să satisfacă constrângerile impuse de fișierul XML
- Import Excel – Parsează fișierul Excel și extrage datele din acesta.
- Export XML – Completează fișierul de export XML
- ReportGenerator - Generează informații referitoare la modificari asupra pinilor
- InfoGenerator – Generează o listă cu pini și caracteristicile lor
- Database – Stochează informațiile necesare prelucrării și cele prelucrate
- Dataset – Stochează informații în timpul rulării aplicație

2.3 Relațiile dintre componente

Componenta Acces Date realizează accesul la datele (din baza de date) necesare prelucrării, componentei Import Excel. Aceasta furnizează date componentei Calculator. Informațiile prelucrate aici sunt folosite pentru modificarea datelor necorespunzătoare din fișierul XML, deci Calculator transferă date componentei Export XML. Modificările fișierului XML sunt input-ul componentei ReportGenerator care afișează utilizatorului modificările efectuate ale caracteristicilor pinilor.

Componenta Calculator direcționează date și spre InfoGenerator, care afișează lista porturilor cu pini și caracteristicile lor.

3. Prezentarea sistemului software

3.1 Arhitectura privită din perspectiva logică

Perspectiva logică - descrie elementele semnificative ale unei arhitecturi și relațiile dintre ele (Fig. 1).

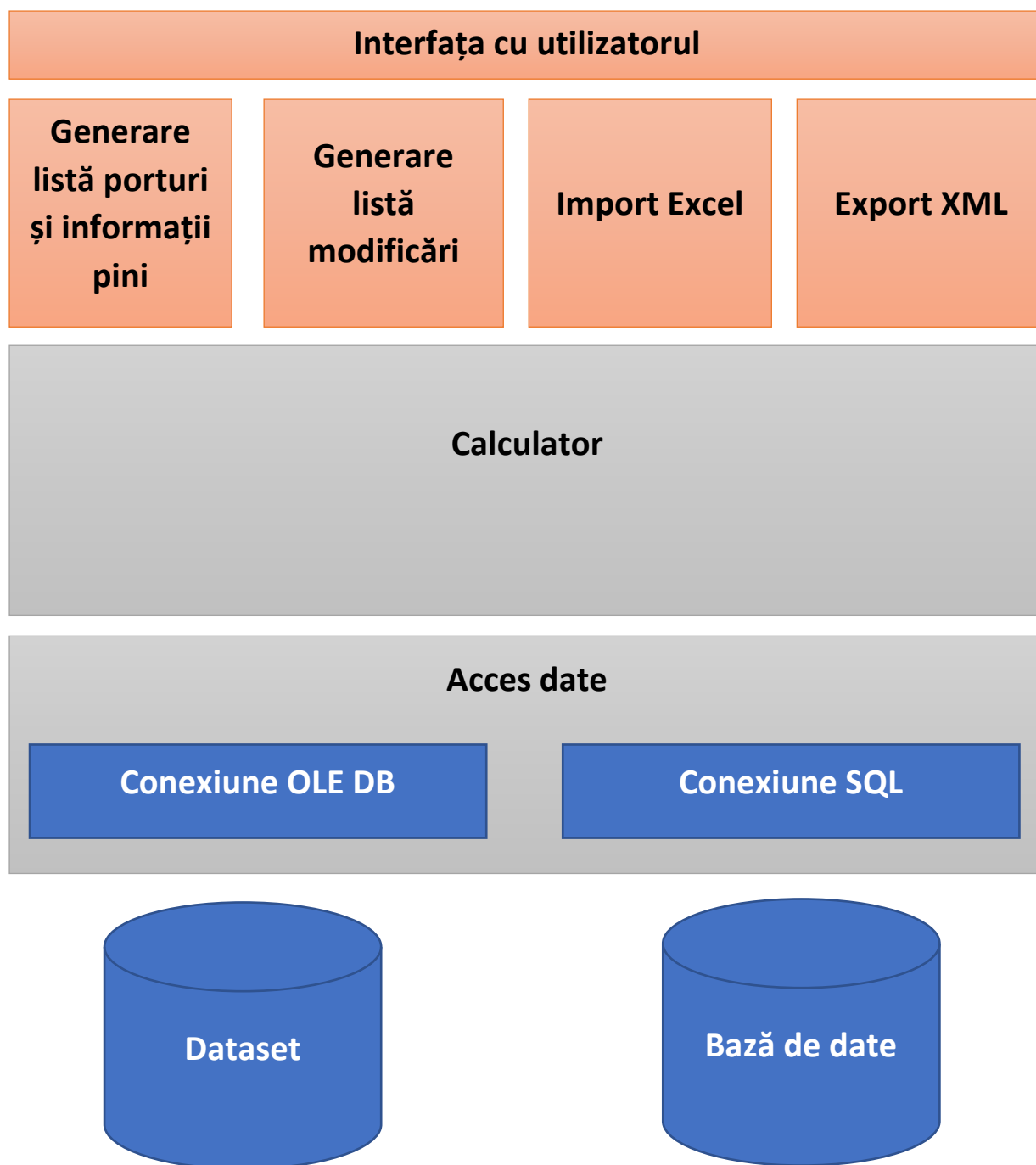


Fig. 1 – Arhitectura sistemului din perspectivă logică

3.2 Arhitectura privită din perspectiva proces

Perspectiva proces - accentul se pune pe descrierea concurenței și a comunicării între componentele unui sistem software (Fig. 2).

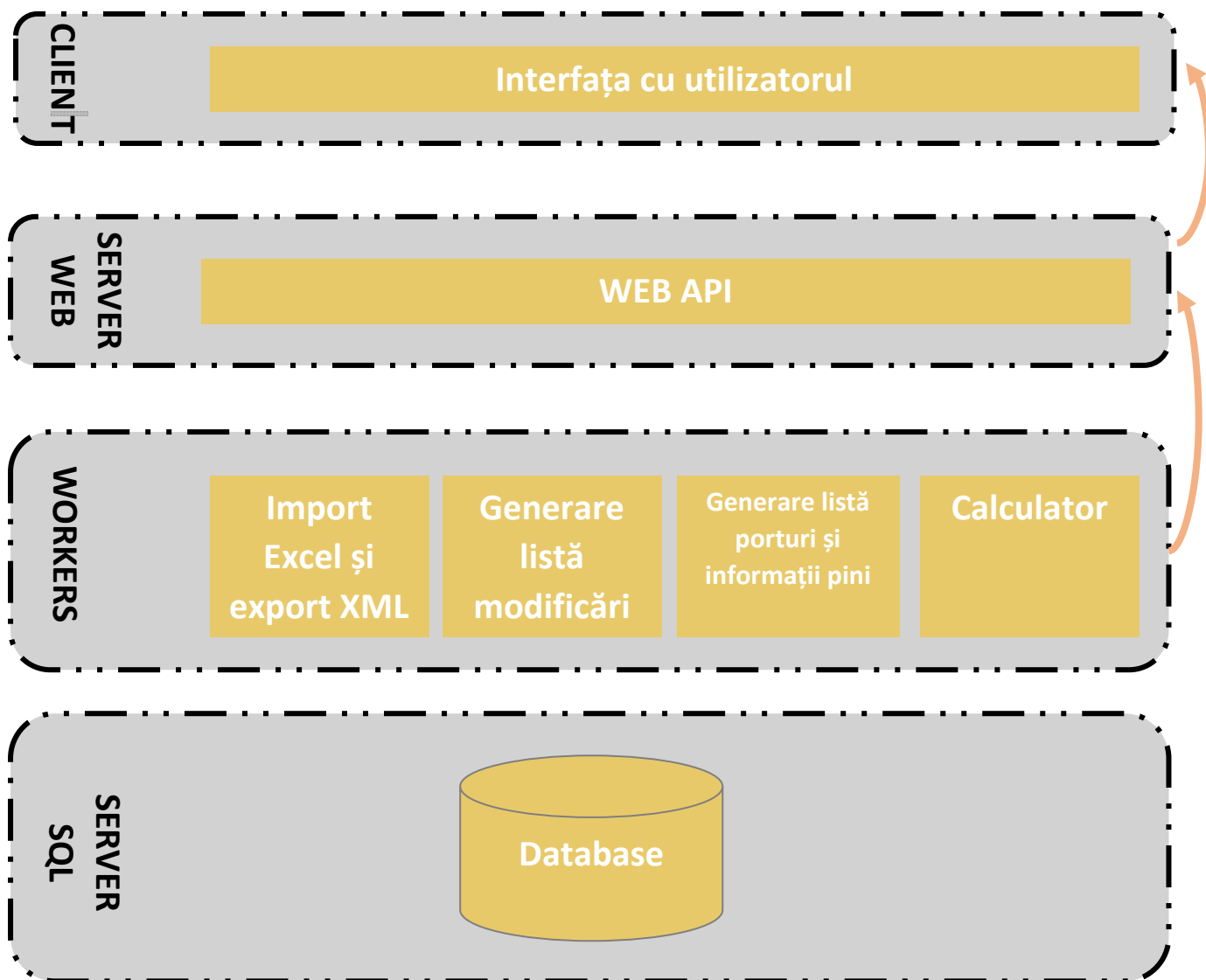


Fig. 2 – Arhitectura sistemului din perspectiva proces

4. Identificarea celor mai importanți 3 indicatori de calitate, specificarea măsurii alese pentru fiecare indicator de calitate

4.1 Performanța

Reprezintă fie o măsură a volumului de procesări în unitatea de timp, fie termenul (deadline) care trebuie respectat pentru finalizarea corectă a unei operații

Măsuri:

- Termenul (deadline)

Aplicația trebuie să aibă un timp de răspuns sub 5 secunde și trebuie să proceseze (parseze) fișierul Excel în timp mai mic de 4 secunde.

- Puterea de procesare

Aplicația trebuie să fie capabilă să proceseze un volum cât mai mare în unitatea de timp (astfel se economisește cu adevărat resurse – bani - și timp)

4.2 Fiabilitatea

Pierderea datelor nu este acceptată.

4.3 Scalabilitatea

Dimensiunea datelor – comportamentul sistemului atunci când volumul de date crește.

Aplicația trebuie să fie în stare să lucreze într-un mod eficient cu un volum mare de date.

4.4 Toleranța la modificări

Măsoară cât de ușor sau cât de greu este să se modifice sistemul software pentru a implementa cerințe noi; trebuie anticipate modificări; se evaluează costul implicat de realizarea modificărilor; implementarea de noi cerințe nu trebuie să ducă la modificări în lanț.

În această privință se observă că aplicația se poate adapta ușor la modificări datorită modului de dependență între componente și chiar testabilitatea poate fi menționată deoarece se poate testa într-un mod destul de eficient și ușor prin input-uri de diferite dimensiuni și valori.

5. Identificarea tehnologiilor middleware folosite pentru a comunica între componente

- **OLE DB connection** (*Object Linking and Embedding, Database*, sometimes written as **OLEDB** or **OLE-DB**)
- **WEB Services**

6. Identificarea principalelor modele și stiluri arhitecturale folosite

6.1 Module

Principalele module folosite sunt:

- WEB
- WORKER
- SQL
- INTERFACE

6.2 Stiluri arhitecturale

Stiluri arhitecturale:

- CLIENT-SERVER (cerere și obținere date)
- BROCĂR (informația se transmite modificată)
- COORDONATOR DE PROCESE (un proces care consistă din mai mulți pași)

7. Prezentarea scenariilor de validare a arhitecturii

Motivul etapei de validare este acela de a verifica faptul că arhitectura proiectată este potrivită pentru sistemul software care urmează să fie dezvoltat. Principala dificultate în ceea ce privește validarea unei arhitecturi constă în faptul că în acest moment nu există un produs software “fizic” care să poată fi executat și testat.

Pentru validarea arhitecturii folosim scenarii. Acestea pot fi realizate utilizând UML (use case-uri, state diagram, activity diagram). Pentru aceasta se va acționa cu diferiți stimuli care să aibă efect asupra arhitecturii și se va vedea comportamentul sistemului.

O altă metodă pentru validarea arhitecturii software este realizarea unui prototip, care verifică dacă se poate implementa arhitectura astfel încât cerințele să fie satisfăcute, și, pe de altă parte, se testează dacă tehnologia de comunicare între componente, tehnologia middleware, se comportă așa cum se așteaptă.