

ACADEMIA ROMÂNĂ
Institutul de Cercetări pentru Inteligență Artificială

Sisteme Suport pentru Decizii Bazate pe Comunicații

Rezumatul tezei de doctorat

Doctorand: Mihai BÎZOI

Coordonator științific: Acad. Florin Gheorghe FILIP

București, 2010

Cuprinsul tezei de doctorat

Listă de acronime

Listă de figuri

Listă de tabele

Index de cuvinte

Capitolul 1. Introducere

- 1.1. Introducere
- 1.2. Obiectivele activității de cercetare
- 1.3. Contextul activității de cercetare
- 1.4. Organizarea lucrării

Capitolul 2. Sisteme Suport pentru Decizii: stadiul actual

- 2.1. Sisteme Suport pentru Decizii – prezentare generală
 - 2.1.1. Evoluția Sistemelor Suport pentru Decizii
 - 2.1.2. Definiții și concepte
 - 2.1.3. SSD în clasificarea sistemelor informatice
 - 2.1.4. Clasificarea Sistemelor Suport pentru Decizii
 - 2.1.5. Sisteme Suport pentru Decizii Bazate pe Comunicații
- 2.2. Utilizarea SSD
 - 2.2.1. Tipuri de utilizatori. Clasificare.
 - 2.2.2. Modalități de utilizare
 - 2.2.3. Implicații ale utilizării SSD
 - 2.2.4. Utilizarea SSD Bazate pe Comunicații
- 2.3. Tehnologia SSD
 - 2.3.1. Tehnologii utilizate la construirea SSD
 - 2.3.2. Tehnologii pentru SSD Bazate pe Comunicații
- 2.4. Construirea SSD
 - 2.4.1. Metode de proiectare
 - 2.4.2. Strategii de abordare și realizare aplicabile SSD
 - 2.4.3. Construirea SSD Bazate pe Comunicații

Concluzii

Capitolul 3. Arhitectură pentru SSD Bazat pe Comunicații

- 3.1. Arhitecturi pentru sisteme informatice
 - 3.1.1. Ciclul de viață al unui produs/sistem informatic
 - 3.1.2. Procesul de proiectare al unei arhitecturi
 - 3.1.3. Proiectarea software
- 3.2. Stabilirea cerințelor de proiectare
 - 3.2.1. Elaborarea deciziilor în grup
 - 3.2.2. Tehnici pentru susținerea lucrului în grup
 - 3.2.3. Caracteristicile Sistemelor de Asistare a Deciziilor Multiparticipant
- 3.3. Proiectarea modelului SSD experimental
 - 3.3.1. Limbajul UML
 - 3.3.2. Descrierea modelului experimental

Concluzii

Capitolul 4. Construirea SSD Bazat pe Comunicații

- 4.1. Construirea modelului experimental
 - 4.1.1. Strategia orientată obiect
 - 4.1.2. Principii ale programării orientate pe obiecte
 - 4.1.3. Avantajele programării orientate pe obiecte
 - 4.1.4. Modelul MVC
 - 4.1.5. Limbajul Perl și interfața CGI
 - 4.1.6. Metoda iterativă de dezvoltare
- 4.2. Demonstrarea funcționalității modelului experimental
 - 4.2.1. Caracteristicile aplicației Allego
 - 4.2.2. Organizarea bazei de date
 - 4.2.3. Modelul aplicației Allego
 - 4.2.4. Generarea interfeței web
 - 4.2.5. Aspecte de securitate

Concluzii

Capitolul 5. Concluzii și perspective

5.1. Contribuții personale

5.2. Perspective de continuare a cercetărilor

Concluzii

Referințe bibliografice

Anexe (Cod sursă)

- A. Clasa AppModel::Stare::Sesiune
- B. Clasa AppModel::Generare_idei::Brainstorming
- C. Clasa AppModel::Configurare::Generare
- D. Pagina index

Capitolul 1. Introducere

1.1. Introducere

Această teză urmărește aducerea de contribuții la domeniului de cercetare *Sisteme Suport pentru Decizii* (SSD), din cadrul Științei Calculatoarelor (Științe Inginerești), în general și în particular la subdomeniul *Sistemelor Suport pentru Decizii Bazate pe Comunicații* (SSDBC).

Sistemele Suport pentru Decizii (SSD) formează o clasă distinctă de sisteme informatice. Acestea integrează instrumente informatice specifice de asistare a deciziilor împreună cu cele de uz general pentru a forma o parte constitutivă a sistemului global al organizației (Filip, *Sisteme suport pentru decizii*, 2004).

Performanțele anumitor echipe de lucru pot fi îmbunătățite folosind *Sisteme Suport pentru Decizii de Grup*, aplicații de tip *groupware* și alte instrumente de comunicare și colaborare. Din această cauză, managerii implementează *Sisteme Suport pentru Decizii Bazate pe Comunicații* (Power D. , *Decision support systems: concepts and resources for managers*, 2002).

Sistemele Suport pentru Decizii Bazate pe Comunicații (SSDBC) sunt o categorie de *Sisteme Suport pentru Decizii* care utilizează rețeaua și tehnologiile de comunicare pentru a facilita colaborarea și partajarea suportului de luare a deciziei. Tehnologiile de comunicare reprezintă pilonul central în activitățile de asistare al deciziei și furnizează elementele de funcționalitate pentru acestea. SSDBC sunt *Sisteme Suport pentru Decizii Multiparticipant* (SSDM) care permit mai multor persoane să comunice între ele, să partajeze informații și să coordoneze activitățile.

Există mai multe denumiri care descriu aplicațiile software de asistare a grupurilor în luarea deciziilor, cum ar fi: sisteme de întâlniri electronice, software pentru lucru colaborativ, groupware, *Sisteme Suport pentru Decizii de Grup* (SSDG) etc. Scopul acestor tehnologii este de a ajuta membrii unui grup să ia decizii în condiții mai bune și să execute activități care s-ar fi desfășurat mai greu fără ajutorul calculatorului. Toate aceste tehnologii pot fi înglobate

sub numele *Sisteme Suport pentru Decizii Bazate pe Comunicații* (Power D. , Decision support systems: concepts and resources for managers, 2002).

Dezvoltarea Web-ului și a infrastructurii Internet constituie factori importanți pentru realizarea de *Sisteme Suport pentru Decizii Bazate pe Comunicații* din ce în ce mai puternice. De aceea, SSDBC trebuie să aibă cel puțin una din următoarele caracteristici: să faciliteze comunicarea între mai multe grupuri de persoane; să permită partajarea informațiilor; să susțină colaborarea și coordonarea dintre participanți și/sau să asiste activitățile de luare a deciziilor în grup.

1.2. Obiectivele activității de cercetare

Primele studii teoretice referitoare la *Sisteme Suport pentru Decizii* au apărut în anii 1950-1960, iar cele pentru *Sistemele Suport pentru Decizii de Grup* după anii 1980. Dezvoltarea rețelelor de calculatoare și a sistemelor de comunicații a condus la apariția unei noi clase de *Sisteme Suport pentru Decizii: Sistemele Suport pentru Decizii Bazate pe Comunicații*, care au prins contur odată cu dezvoltarea tehnologică de după anii 1990 - 2000. Deși nu sunt întâlnite de obicei sub acest nume, acest tip de sisteme informatice sunt larg răspândite, într-o rapidă și continuă dezvoltare. Principalul obiectiv a activității de cercetare l-a constituit aducerea de contribuții în domeniul *Sistemelor Suport pentru Decizii Bazate pe Comunicații*.

Obiectivele specifice urmărite prin intermediul planului de pregătire au fost:

- ✓ Fundamentarea cunoștințelor de bază în privința metodelor numerice pentru analiza deciziilor, a sistemelor de comunicații bazate pe calculator și a metodologiilor de construire a sistemelor informatice;
- ✓ Realizarea unui studiu sistematic pentru cunoașterea stadiului actual al clasei sistemelor informatice *Sisteme Suport pentru Decizii*. Acest studiu urmărea realizarea unui prezentări de ansamblu a domeniului de cercetare, atingând următoarele puncte: cazurile de utilizare, tehnologii folosite pentru realizarea sistemelor suport pentru decizii și metodologii de construire ale acestora;

- ✓ Realizarea unui studiu pentru cunoașterea [sub] domeniului *Sisteme Suport pentru Decizii Bazate pe Comunicații* și identificarea unor noi direcții de cercetare în cadrul acestui domeniu;
- ✓ Propunerea unei arhitecturi pentru un Sistem Suport pentru Decizii Bazat pe Comunicații experimental;
- ✓ Prezentarea rezultatelor obținute în urma activității de cercetare;
- ✓ Validarea rezultatelor cercetării prin prezentarea acestora la manifestări științifice și publicarea în volumele unor conferințe sau jurnale.

1.3. Contextul activității de cercetare

În vederea stabilirii contextului activității de cercetare, a fost realizat un studiu pentru evidențierea modului de publicare a materialelor care prezintă rezultatele activității de cercetare din domeniul studiat. Astfel, au fost interogate patru baze de date internaționale: două baze de date care indexează rezultatele cercetării științifice: *ISI Web of Knowledge* (***, Web of Knowledge - Science - Thomson Reuters) și *Science Direct* (***, ScienceDirect - Browse Journals and Books), precum și două biblioteci digitale ale unor prestigioase asociații profesionale cu recunoaștere internațională: *IEEE Xplore Digital Library* (***, IEEE Xplore - Home) și *The ACM Digital Library* (***, ACM Digital Library).

Căutarea s-a realizat având în vedere șapte șiruri de cuvinte cheie (în limba engleză) relevante pentru domeniul studiat: ("Decision Support Systems"), ("Communications Driven Decision Support Systems"), ("Decision Support System" și "Communications"), ("Collaborative Platform"), ("Groupware"), ("Unified Communications" și "Systems") și (Web și "Decision Support System").

Ca o scurtă concluzie, studiul realizat evidențiază interesul comunității științifice în domeniul *Sistemelor Suport pentru Decizii*. Numărul mare de lucrări într-o continuă creștere ne indică faptul că acesta este un domeniu de cercetare dinamic, cu potențial și pentru cercetări ulterioare.

Numărul foarte mic de lucrări, dar în creștere, care abordează *Sistemele Suport pentru Decizii Bazate pe Comunicații*, precum și lucrările referitoare la sistemele suport pentru

decizii care abordează domeniul comunicațiilor, platformele de colaborare și platformele de tip *groupware* arată că acest domeniu de cercetare se află încă la început.

1.4. Organizarea lucrării

Această lucrare abordează domeniul *Sistemelor Suport pentru Decizii Bazate pe Comunicații* din perspectiva clasificării acestora în superclasa sistemelor informatice. Din cauza domeniului vast de cercetare, sunt prezentate succesiv la un grad mai înalt sau redus de generalitate metode, tehnici, tehnologii, abordări (uneori în baza unei localizări în timp), care în principiu descriu ciclul de viață al sistemelor informatice. Bineînțeles, aprofundarea noțiunilor are în vedere strict referințele *Sistemelor Suport pentru Decizii Bazate pe Comunicații*.

Materialul este organizat în cinci capitole după cum urmează: (1) Introducere, (2) Sisteme Suport pentru Decizii: stadiul actual, (3) Arhitectură pentru SSD bazat pe comunicații, (4) Construirea SSD bazat pe comunicații și (5) Concluzii și perspective.

Capitolul 1 prezintă obiectivele planului de cercetare și contextul activității de cercetare, prin prezentarea unui studiu referitor la publicațiile din domeniul de cercetare care au apărut în ultimii zece ani în patru baze de date internaționale.

Capitolul 2, se axează pe definiția și clasificarea *Sistemelor Suport pentru Decizii* (subcapitolul 2.1), cuprinzând un scurt istoric ce prezintă evoluția SSD (2.1.1), definiții și concepte (2.1.2), clasificarea SSD în clasificarea sistemelor informatice (2.1.3), clasificarea SSD (2.1.4) și o scurtă prezentare a SSDBC (2.1.5). De asemenea, în subcapitolul 2.2 este abordat subiectul tipurilor de utilizatori (2.2.1), modalitățile de utilizare (2.2.2), implicațiile utilizării SSD (2.2.3) precum și elemente referitoare la utilizarea SSDBC (2.2.4).

În subcapitolul 2.3 *Tehnologia SSD* sunt prezentate tehnologii care pot fi utilizate la construirea SSD în general și se particularizează ce tehnologii pot fi folosite pentru construirea *Sistemelor Suport pentru Decizii Bazate pe Comunicații*, iar în ultimul subcapitol, 2.4 *Construirea SSD* sunt evidențiate metodele proiectării SSD (2.4.1), strategii de abordare și realizare aplicabile SSD (2.4.2.), precum și elemente referitoare la construirea *Sistemelor Suport pentru Decizii Bazate pe Comunicații* (2.4.3).

Capitolul 3, *Arhitectură pentru sistem suport pentru decizii bazat pe comunicații* prezintă un model experimental de arhitectură al unui *Sistem Suport pentru Decizii Multiparticipant*. Aceasta este organizat în trei subcapitole: 3.1. *Arhitecturi pentru sisteme informatice*, 3.2. *Stabilirea cerințelor de proiectare* și 3.3 *Proiectarea modelului SSD experimental*. Subcapitolul 3.1 *Arhitecturi pentru sisteme informatice* prezintă ciclul de viață al unui produs informatic din perspectiva arhitecturală (3.1.1), luându-se în considerare cele patru faze: pre-proiectarea, analiza de domeniu, proiectarea schematică și realizarea propriu-zisă a proiectului.

Procesul de proiectare al unei arhitecturi (3.1.2) este descris la modul general pe baza următorilor pași: înțelegerea problemei, identificarea elementelor de proiectare și a relațiilor dintre ele, evaluarea arhitecturii, transformarea arhitecturii. De asemenea, sunt prezentate eventualele probleme care pot apărea în faza de proiectare software (3.1.3), precum și sarcinile și activitățile de proiectare.

În subcapitolul 3.2 *Stabilirea cerințelor de proiectare* sunt prezentate caracteristicile generale pentru luarea deciziilor în grup (3.2.1), precum și potențialele avantaje și dezavantaje ale lucrului în grup.

În cadrul aceluiași subcapitol sunt evidențiate trei tehnici pentru susținerea lucrului în grup (3.2.2): brainstorming-ul, tehnica grupului nominal și metoda Delphi. Totodată sunt prezentate și caracteristicile *Sistemelor de Asistare a Deciziilor Multiparticipant*.

Ultimul subcapitol, 3.3 *Proiectarea modelului SSD experimental* descrie la modul general limbajul de modelare UML (3.3.1) și prezintă un exemplu: arhitectura unui sistem suport pentru decizii experimental (3.3.2). Modelul arhitecturii sistemului a fost realizat folosind diagrame de clase, iar subsistemele modelului au fost reprezentate sub formă de pachete software.

Capitolul 4, *Construirea sistemului suport pentru decizii bazat pe comunicații*, reprezintă o continuare firească a celor două capitole menționate anterior și vizează prezentarea modului de implementare al unui *Sistem Suport pentru Decizii Bazat pe Comunicații* din cele două perspective: teoretică și practică.

Pentru realizarea aplicației (denumită *Allego*), s-a ales abordarea mixtă (care a fost prezentată în capitolul 1), strategia de dezvoltare fiind cea orientată pe obiecte, iar metoda de implementare aleasă: metoda iterativă.

Subcapitolul 4.1. *Construirea modelului experimental* prezintă strategia orientată obiect (4.1.1), elemente fundamentale din teoria programării orientate pe obiecte: principiile de bază ale programării orientate pe obiecte (4.1.2), definiția obiectelor, mesajelor și metodelor, claselor și instanțelor, precum și avantajele programării orientate pe obiecte (4.1.3). Această fundamentare teoretică a fost considerată absolut necesară pentru implementarea modelului aplicației, deoarece limbajul de programare ales funcționează atât în mod procedural cât și în mod orientat obiect.

În același subcapitol este prezentat limbajul de programare ales (*Perl*), însoțit de motivația alegerii acestuia (4.1.5). Deoarece modelul aplicației a fost conceput în arhitectura prezentată în capitolul 3, ca fiind separat de interfață, a fost acordată o atenție deosebită modelului MVC (Model View Controller) – un concept teoretic care presupune separarea funcțională a interfeței de aplicația model (4.1.4). Pentru dezvoltarea aplicației a fost aleasă metoda iterativă, prezentată în sub-subcapitolul 4.1.6.

În subcapitolul 4.2. *Demonstrarea funcționalității modelului experimental*, sunt prezentate caracteristicile aplicației *Allego* (4.2.1) prin comparație cu caracteristicile ideale ale unui *Sistem Suport pentru Decizii*.

De asemenea, se prezintă modul de organizare al bazei de date (4.2.2) și se dau câteva exemple de structurare a bazei de date. Modul de realizare al claselor aplicației model (4.2.3) și de generare a interfeței (4.2.4) se prezintă prin câteva exemple practice. Codul sursă pentru câteva clase ale aplicației model și prima pagină a interfeței web sunt anexate acestei lucrări. O serie de aspecte de securitate sunt prezentate în sub-subcapitolul 4.2.5.

Mulțumiri

Autorul mulțumește în primul rând domnului Acad. Florin Gheorghe FILIP, coordonatorul științific al tezei de doctorat, pentru sprijinul acordat în toată perioada desfășurării planului de cercetare, dar și în afara acestuia, pentru implicarea activă în susținerea activităților de cercetare, dar și pentru sprijinul acordat autorului în alte activități menite să îi crească prestigiul profesional.

De asemenea, autorul mulțumește instituției organizatoare al programului de doctorat, Institutul pentru Cercetări în Inteligență Artificială, în special domnului prof. dr. ing. Ioan Dan TUFIȘ, membru corespondent al Academiei Române și doamnei dr. Angela IONIȚĂ pentru eforturile de a organiza în bune condiții programele doctorale și pentru participarea în comisiile de examinare din cadrul acestor programe.

Pentru recomandările referitoare la modul de prezentare al acestei teze, autorul mulțumește membrilor colectivului de specialiști de la Institutul pentru Cercetări în Inteligență Artificială.

Autorul mulțumește membrilor comisiei de examinare a tezei de doctorat pentru timpul alocat analizei acestei lucrări și pentru referatele făcute.

Pentru condițiile de studiu și cercetare, care au contribuit la realizarea acestei teze, autorul mulțumește tuturor colaboratorilor și colegilor din Facultatea de Inginerie Electrică, Universitatea Valahia din Târgoviște.

Nu în ultimul rând, autorul mulțumește familiei pentru sprijinul și încrederea acordată, precum și pentru eforturile făcute în a-i asigura o bună educație.

Capitolul 2. Sisteme Suport pentru Decizii: stadiul actual

2.1. Sisteme Suport pentru Decizii – prezentare generală

În acest subcapitol se prezintă pe scurt **evoluția** Sistemelor Suport pentru Decizii (SSD), însoțită de o serie de **definiții și concepte**, cu scopul de a contura **definiția SSD** pe baza ideilor prezentate de mai mulți autori. De asemenea, sunt abordate **tipurile de decizii, fazele procesului decizional și o serie de modele alternative pentru luarea deciziilor**. Din cauză că nu există o definiție exactă a Sistemelor Suport pentru Decizii, procesul de definire al acestora este continuat prin prezentarea **caracteristicilor principale** ale acestei clase de sisteme informatice. Pentru a introduce o **clasificarea a Sistemelor Suport pentru Decizii**, prin utilizarea modelului piramidal se prezintă apriori o **clasificare a sistemelor informatice**. Clasificarea SSD se face prin expunerea punctelor de vedere ale mai multor autori și concluzionează în mod parțial, unele **avantaje și limitări** rezultate în urma implementării Sistemelor Suport pentru Decizii. Ultima parte se concentrează pe încercarea de definire a **Sistemelor Suport pentru Decizii Bazate pe Comunicații (SSDBC)**. Pornind de la ideea că SSDBC se regăsesc în literatura de specialitate și sub alte denumiri, au fost abordate pe rând: **Sistemele Suport pentru Decizii Organizaționale (SSDO), Sistemele Suport pentru Decizii de Grup (SSDG)**, platformele de tip **groupware / platforme de colaborare** și relația dintre acestea, precum și rolul **videoconferinței** în procesul luării deciziei.

Sistemele Suport pentru Decizii aparțin unui mediu cu fundamente multidisciplinare, incluzând cercetarea în baze de date, inteligență artificială, interacțiunea om-calculator, metode de simulare, inginerie software și telecomunicații.

Conceptul *Sistemului Suport pentru Decizii* (SSD) este extrem de larg și definiția sa diferă în funcție de punctul de vedere al autorului (Druzdzet și Flynn, Decision Support Systems, 1999).

Prima definiție a *Sistemelor Suport pentru Decizii* a fost dată de Little, la începutul anilor '70. El definea SSD-ul ca fiind: "un model bazat pe un set de proceduri pentru procesarea datelor și pentru asistarea unui manager în procesul decizional. Un SSD trebuie să fie simplu, robust, ușor de întreținut, adaptiv, ușor de comunicat cu el etc.". Aceste atribute, formulate de Little sunt valabile și astăzi.

Gorry și Scott Morton (1971) citați de (Filip, Sisteme suport pentru decizii, 2004) identificau SSD-urile drept sisteme informatice care au menirea să ajute la elaborarea deciziilor în situații nestructurate și semi-structurate (în care problema nu poate fi analizată complet pentru a lua o decizie, iar rezolvarea ei nu se poate programa sub forma unei secvențe de pași).

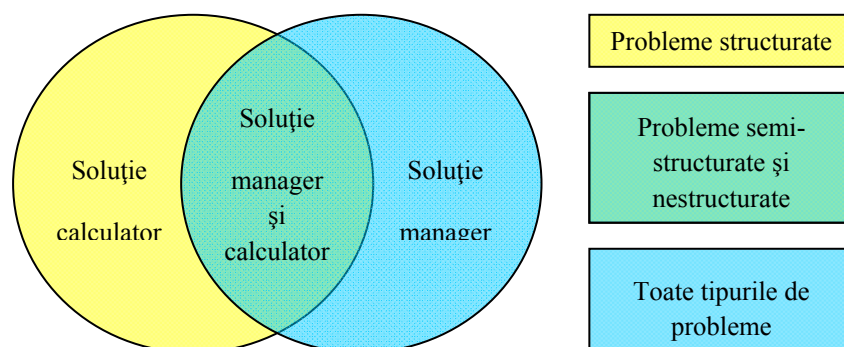


Figura 1. Găsirea soluțiilor la problemele decizionale.

În figura de mai sus, sunt prezentate tipurile de probleme decizionale și modul de găsire al soluțiilor la aceste tipuri de probleme.

Caracteristicile SSD

Deoarece nu există o definiție exactă a *Sistemelor Suport pentru Decizii*, nu există o concordanță evidentă între caracteristicile standard și posibilitățile SSD. (Turban, Aronson și Liang, *Decision Support Systems and Intelligent Systems*, 7th Ed., 2005) au identificat un set ideal de caracteristici și posibilități ale *Sistemelor Suport pentru Decizii*, după cum urmează:

- ✓ Asistă decidenții în probleme semi-structurate și nestructurate;
- ✓ Asistă managerii la toate nivelurile;
- ✓ Asistă indivizii și grupurile;
- ✓ Asigură suport pentru decizii interdependente sau secvențiale;
- ✓ Susține informarea, proiectarea, alegerea și implementarea;
- ✓ Suportă o varietate de procese decizionale și stiluri;
- ✓ SSD trebuie să fie adaptabile și flexibile;
- ✓ SSD trebuie să fie interactive și ușor de folosit;
- ✓ Eficacitatea echilibrată cu eficiența (beneficiile trebuie să depășească costurile);
- ✓ Decidenții au control total;
- ✓ Ușurință în dezvoltare de către utilizatorii finali (modificări care să răspundă nevoilor și schimbărilor de mediu);

- ✓ Susține modelarea și analiza;
- ✓ Asigură acces la date;
- ✓ E o aplicație independentă, integrată și bazată pe web.

Clasificarea sistemelor informatice

Majoritatea autorilor clasifică sistemele informaționale plecând de la rolul acestora în procesele manageriale și se axează pe relația informație-decizie. În figura de mai jos sunt prezentate sub formă de piramidă, principalele tipuri de sisteme informaționale.

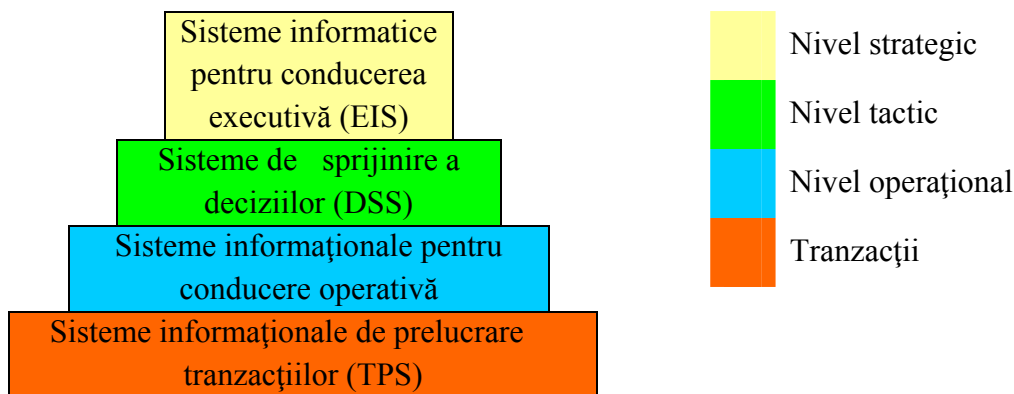


Figura 2. Reprezentarea piramidală a sistemelor informaționale

Apariția unor sisteme informatice cu arii extinse de cuprindere, cum ar fi sistemele expert, sistemele informaționale pentru grupurile de lucru etc., a determinat puncte noi de vedere referitoare la structura piramidală a sistemelor informatice, deoarece acestea se adresează tuturor nivelurilor din piramidă.

Din cauza numărului mare de termeni, care au creat probleme în domeniul cercetării SSD, în decursul timpului au fost propuse mai multe criterii pentru realizarea unei clasificări a *Sistemelor Suport pentru Decizii* (Cioca și Cioca, 2010).

Sisteme suport pentru decizii organizaționale (SSDO)

Dezvoltarea termenului de *Sistem Suport pentru Decizii* (SSD) a condus la conceptul de *Sistem Suport pentru Decizii Organizațional* (SSDO), introdus inițial de (Hackathorn și Keen, 1981). În funcție de numărul decidenților implicați în luarea deciziei, au definit trei tipuri de sisteme suport pentru decizii: individuale, de grup și organizaționale. Asemănător termenului SSD, numeroși autori au stabilit diferite definiții ale SSDO.

(Kivijärvi și Kuula, 1996) definește SSDO astfel: “Un Sistem Suport pentru Decizii Organizațional reprezintă un scop general, multiutilizator, un sistem la scara largă, care este proiectat pentru o varietate a deciziilor organizaționale și are relativ definită poziția continuă și organizată în planificare și procesele luării deciziilor unei companii.” Watson (1990) se referă la termen ca o combinație de calculatoare și tehnologia comunicării care este proiectată să coordoneze și să disemineze luarea deciziei în organizații în care deciziile care coincid cu scopurile organizaționale și în care există o viziune comună a managerilor în ce privește mediul competițional în organizație.

După King și Star (1990), citați de George (2008), conceptul SSDO este o aplicație a tehnologiei calculatoarelor și a comunicațiilor a cărui scop este să îmbunătățească procesul luării deciziei în organizație. Astfel, SSDO furnizează același tip de suport tehnic pentru un grup de decidenți la fel ca un sistem suport de decizii de grup.

În multitudinea definițiilor date de diferiți autori care accentuau aspecte diferite ale termenului SSDO, George (1991) a găsit câteva caracteristici comune ale acestora:

- SSDO se concentrează asupra activităților și deciziilor organizaționale unde sunt implicate diferite probleme și/sau unități organizaționale;
- Rolul SSDO nu este limitat doar la probleme și/sau unități organizaționale, SSDO poate afecta diferite niveluri funcționale și ierarhice în organizație;
- În general, un SSDO încorporează diferite tipuri de tehnologii de comunicații bazate pe calculator.

Sisteme suport pentru decizii de grup (SSDG)

Sistemele suport de grup sau sistemele suport pentru decizii de grup constă în tehnologie care susțin activitățile desfășurate de decidenți în grup. Conform lui Turban și Aronson (Decision Support Systems and Intelligent Systems, 1998), grupul decidenților este asistat de un lider care planifică întâlnirile, coordonează activitățile echipei de asemenea ca facilitator ale cărui responsabilități sunt să accepte promovarea utilizării tehnicilor de rezolvare a problemelor și încurajează atingerea consensului.

Aceste tipuri de sisteme susțin generarea ideii, analiza problemei, facilitează luarea deciziei în grup și îmbunătățește calitatea deciziei prin reducerea riscului de “gândire de grup” prin furnizarea posibilității de a face anonime opiniile.

DeSanctis și Gallupe (1987) au definit SSDG ca un sistem computerizat interactiv al cărui scop este facilitarea luării deciziei de către un grup de decidenți în cazul problemelor nestructurate.

O altă definiție a sistemelor suport pentru decizii de grup este dată de Vogel și Nunamaker (1990). Aceștia considerau că un SSDG folosește tehnologie pentru a susține rezolvarea de probleme în situațiile grupurilor de decizie, îmbunătățind performanțele deciziei și eficacitatea grupului.

Huber (1984) consideră că un sistem suport pentru decizii de grup este un sistem informațional bazat pe calculator care intensifică luarea deciziei de grup prin facilitarea schimbului și utilizării de informații între membrii grupului și interacționează între grup și calculator pentru a formula și rezolva probleme nestructurate.

Turban și Aronson (Decision Support Systems and Intelligent Systems, 1998) au stabilit o serie de caracteristici și capabilități al SSDG după cum urmează:

- Un SSDG nu este doar o configurație a unui sistem deja existent, este un sistem proiectat special;
- Scopul SSDG este să susțină grupurile de decidenți în munca lor. Acesta ar trebui să îmbunătățească procesul luării deciziei și rezultatele grupului;
- Asemănător SSD, SSDG trebuie să fie ușor de utilizat și de utilizatorii care nu au aptitudini deosebite în e privește calculatorul;
- Proiectarea SSDG poate fi direcționată numai spre tipul de probleme sau varietatea de probleme care reprezintă subiectul luării deciziei;
- Un SSDG ar trebui proiectat astfel încât să încurajeze rezolvarea conflictelor, libertatea de exprimare și generarea de idei;
- Un SSDG ar trebui să furnizeze mecanisme care descurajează gândirea de grup negativă, lipsa de comunicare și conflictele distructive.

Groupware și platformele de colaborare

Anumiți autori fac o distincție clară între *Computer Supported Cooperative Work* (CSCW) și *Group Decision Support System* (GDSS) (Dennis, George, Jessup, Nunamaker și Vogel, 1998), considerând că GDSS sunt mai potrivite pentru susținerea activităților, iar instrumentele CSCW furnizează suport pentru comunicare în general (Bîzoi, Gorghiu, Suduc,

și Alexandru, 2006), cu toate că definiția celor două concepte sugerează o strânsă relație între acestea.

În mod curent, definițiile prezintă GDSS ca pe un sistem care susține un efort de grup concertat și coordonat în vederea completării procesului de luare a deciziei. *Groupware* este o tehnologie proiectată să faciliteze comunicarea, cooperarea, coordonarea, rezolvarea de probleme, competițiile sau negocierile grupurilor de lucru (Bîzoi, Suduc, Gorghiu și Gorghiu, 2009).

CSCW se referă la ramura de studiu care examinează proiectarea, adoptarea și utilizarea groupware (Brinck, 1998). Din această perspectivă, se poate considera ca GDSS este o subclasă de groupware. În același timp GDSS este un tip particular de DSS, care se ocupă de grupuri și nu de indivizi (Bîzoi, Suduc și Filip, Using Collaborative Platforms for Decision Support, 2009). În Figura 3 este prezentată relația dintre DSS, GDSS și groupware.



Figura 3. Relația dintre DSS, GDSS și Groupware

Între groupware și GDSS există asemănări și deosebiri (Yen, Gong, Wen și Chen, 1998). Principalele asemănări sunt: (a) ambele tehnologii susțin lucrul în grup și echipele care au o sarcină comună; (b) ambele sunt proiectate pentru a crește productivitatea grupului prin furnizarea de canale de comunicare simultane; (c) ambele sisteme folosesc interfețe de grup; (d) nu reprezintă o problemă localizarea fizică a membrilor grupurilor.

În ciuda asemănărilor, cele două tehnologii au și caracteristici unice: (a) Groupware se referă la o varietate largă de sisteme, fiecare dintre acestea fiind proiectate să îmbunătățească productivitatea grupului în anumite privințe. GDSS este orientat către îmbunătățirea procesului de luare a deciziei; (b) groupware utilizează o interfață generală pentru tot grupul, GDSS folosește interfețe pentru managementul dialogului; (c) GDSS se folosesc mai ales pentru întâlniri care se desfășoară în același timp și loc, groupware se folosesc atât pentru întâlniri sincrone cât și pentru întâlniri asincrone, la distanță.

Videoconferința

Asistarea deciziei cu ajutorul videoconferinței este proiectată a se realiza atunci când operațiile de afaceri sunt dispersate la mare distanță. În acest moment, din punct de vedere

tehnologic, există două tipuri de videoconferință: (a) primul tip creează o legătură între un calculator personal cu un alt calculator personal, permițând interacțiunea între două persoane; (b) al doilea tip se referă la interacțiunea între indivizi multipli sau indivizi și grupuri în camere de conferință sau la calculatoare personale (Suduc, Bîzoi și Filip, Exploring Multimedia Web Conferencing, 2009).

Aplicațiile software de videoconferință pot fi împărțite în două categorii: (a) o categorie o reprezintă platformele software care permit numai conferința audio / video și o structură care facilitează implicarea unui moderator; (b) a doua categorie, mai evoluată, furnizează facilități pentru partajarea documentelor și a fișierelor, acces la desktop-uri partajate, editare simultană a materialelor și alte forme electronice de comunicare ce permit datelor să fie partajate, editate și copiate pe parcursul întâlnirii (Suduc, Bîzoi și Filip, Exploring Multimedia Web Conferencing, 2009).

Instrumentele bazate pe web și videoconferința pot accelera procesul de luare a deciziei reducând timpul și costurile de organizare a întâlnirii. De asemenea, sistemele de videoconferință cu instrumente bazate pe web pot furniza managerilor accesul la resurse informatice locale și materiale aflate pe web în timpul întâlnirilor (Suduc, Bîzoi, Gorghiu și Gorghiu, 2010).

Sistemele de videoconferință pentru luarea deciziilor fac posibilă implicarea mai multor persoane la întâlnirea de luare a deciziilor și reduc stresul asociat cu transportul participanților în diferite locații.

Condiția de bază ca un astfel de sistem să funcționeze îl reprezintă disponibilitatea unei lărgimi de bandă a rețelei suficient de mare pentru susținerea fluxului multimedia transmis (Bîzoi, Suduc, Gorghiu și Gorghiu, Risk Assessment of Information and Communication Technology Use in Multinational Educational Projects, 2010).

2.2. Utilizarea SSD

Pentru a realiza un tablou cât mai complet în ce privește utilizarea Sistemelor Suport pentru Decizii, au fost tratate inițial **tipurile de utilizatori și clasificarea** acestora. În cadrul **modalităților de utilizare** sunt expuse două modele: **modelul procesual** și **modelul bazat pe cunoaștere**. Utilizarea SSD conduce la o serie de implicații; astfel, au fost prezentate o serie de **caracteristici ale utilizării SSD**, precum și o **serie de efecte** apărute în urma utilizării acestora. **Utilizarea Sistemelor Suport pentru Decizii Bazate pe Comunicații**

accentuează câteva aspecte referitoare la **luarea deciziilor în grup** și evidențiază **situațiile pentru luarea deciziilor în grup**.

Sistemele suport pentru decizii sunt folosite de toate persoanele plasate pe diferite niveluri de autoritate ierarhică ale organizației, care sunt împuternicite să rezolve probleme decizionale, dar și de specialiștii chemați sistematic sau ocazional să contribuie la elaborarea deciziilor conform cunoștințelor pe care le posedă. De asemenea, sunt considerați utilizatori ai SSD atât persoanele care operează la calculator, cât și cei care solicită și analizează alternativele furnizate de personalul cu rol mijlocitor în folosirea sistemului. Acești utilizatori direcți ai SSD pot fi împărțiți în două subclase: decidenți și asistenți ai acestora (consilieri și consultanți care ajută la folosirea sistemului).

În general, definiția Sistemelor Suport pentru Decizii exprimă ideea de sprijinire a decidentului pentru a-și depăși limitele și restricțiile cognitive, temporale, de implementare, de comunicare și de încredere în sistemele informatice.

Consecințe ale controlabilității sistemului asupra utilizării lui, pot fi: utilizarea sistemului la orice moment de timp în care este nevoie de sprijinul acestuia, nu numai la momente prestabilite; libertatea utilizatorului de a schimba cursul desfășurării sesiunii de lucru sau chiar întreruperea acesteia pentru a fi reluată mai târziu (Suduc, Bîzoi și Filip, 2010).

Un SSD ar trebui să fie suficient de flexibil pentru a se adapta la caracteristicile personale ale individului care utilizează sistemul (decident sau asistent decizional). Flexibilitatea poate da încredere utilizatorului să recurgă tot mai des la sistem și să-i folosească mai multe funcții conducând astfel la o evoluție a utilizării (Suduc, Bîzoi și Filip, Aspecte legate de interfețele om-calculator pentru reducerea riscurilor utilizării sistemelor informatice, 2009).

Luarea deciziilor în grup

Luarea deciziilor în grup are la bază mai multe aspecte, după cum urmează:

- ✓ Apariția echipelor de lucru virtuale. Echipa este un grup format dintr-un număr relativ mic de persoane, cu deprinderi și cunoștințe complementare angajate activ în realizarea unor sarcini comune (Banker, Field, Schroeder și Sinha, 1996). Spre

deosebire de grup, performanța echipei este analizată prin prisma capacității sale de a se autoperfecționa (Ancona și Chong, 1996).

- ✓ Agregarea rezultatelor individuale. SSDG sprijină decidenții individuali care nu necesită coordonare în cadrul grupurilor de lucru. Deși nevoia de comunicare între decidenți este aproape inexistentă, productivitatea SSDG este dată de rezultatele obținute de toți decidenții.
- ✓ Îmbunătățirea comunicării și coordonării. Componentele SSDG sunt proiectate pentru a îmbunătăți comunicarea și coordonarea prin serviciile clasice care includ: poșta și agenda electronică, automatizarea fluxului de lucru, analiza interdependenței activităților și mecanisme de anunțare.
- ✓ Colaborarea decidenților. Eficacitatea și eficiența SSDG este influențată în acest caz, deoarece toți decidenții trebuie să contribuie în mod colaborativ la luarea deciziei, iar slaba performanța a unuia dintre ei influențează în mod negativ activitatea.

Situații pentru luarea deciziilor de grup

Situațiile de care pot beneficia *Sistemele Suport pentru Decizii Bazate pe Comunicații* se vor analiza în funcție de locul și timpul desfășurării acțiunii. În mod tradițional, se folosește o matrice 2x2 pentru a reprezenta cele patru situații (Tabelul 1).

Tabelul 1. Combinații pentru asistarea deciziilor de grup

	Același loc	Locuri diferite
Același timp	Camera decizională Calculator cu proiector Instrumente de votare	Videoconferință Audioconferință Tablă albă Ecran partajat Mesagerie instant
Timpuri diferite	Software pentru schimbul de lucrări Partajarea documentelor	Sisteme de conferință Panouri de afișare Poștă electronică Mesagerie vocală

O echipă de lucru, în general, are nevoie de un *Sistem Suport pentru Decizii Bazat pe Comunicații* care să opereze în toate cele patru situații prezentate mai sus.

2.3. Tehnologia SSD

Tehnologiile care pot fi utilizate la construirea Sistemelor Suport pentru Decizii reprezintă un domeniu vast și dificil de prezentat într-o formă concisă. Astfel, în acest subcapitol, au fost prezentate **arhitectura SSD** și **componentele principale** ale Sistemelor Suport pentru Decizii. Având în vedere ideea că Sistemele Suport pentru Decizii au evoluat în paralel cu alte clase de sisteme informatice în urma apariției unor noi tehnologii și nu s-au dezvoltat noi tehnologii pentru construirea SSD, s-a pus accentul pe **integrarea tehnologiilor existente** în vederea realizării Sistemelor Suport pentru Decizii. La modul general, tehnologiile pentru Sistemele Suport pentru Decizii, au la bază **arhitectura client-server** (indiferent dacă acestea sunt bazate pe web sau nu). În ultima parte, a fost introdus conceptul de **Sisteme de Comunicații Unificate** ca platformă pentru dezvoltarea Sistemelor Suport pentru Decizii Bazate pe Comunicații. Tendința convergentă a tehnologiilor actuale de a se integra **într-o singură infrastructură** și caracteristica dominantă a comunicațiilor unificate de a accelera **procesul intensiv de colaborare**, va modifica modul de a defini și dezvolta Sistemele Suport pentru Decizii Bazate pe Comunicații.

Integrarea tehnologiilor

O nouă provocare a apărut odată cu dezvoltarea sistemelor de tip client/server: cum să se controleze toate procesele afacerii cu o singură arhitectură software în timp real. Soluția apărută a fost integrarea, cunoscută și sub numele de *Enterprise Resource Planning* (ERP). Aceasta promitea avantaje de la creșterea eficienței până la creșterea calității, a productivității și profitabilității. Obiectivul major este integrarea tuturor departamentelor și funcțiilor dintr-o întreprindere într-un sistem informatic unic care poate servi tuturor necesităților organizaționale (Airinei, 2006).

Pentru a dezvolta un sistem integrat propriu, opțiunile disponibile sunt: folosirea pachetelor funcționale comerciale sau dezvoltarea propriilor module. O altă variantă constă în folosirea pachetelor software integrate. Aplicațiile de acest tip forțează disciplina și organizarea în cadrul proceselor afacerii, oferind o interfață unică pentru gestionarea tuturor activităților de rutină desfășurate.

Soluțiile ERP au avut un rol deosebit de important în focalizarea afacerilor mici și mijlocii și în modificarea concepțiilor clasice asupra întreprinderii. De asemenea, există și puncte slabe în privința ERP: soluțiile de acest gen sunt centrate pe tranzacții, fără a oferi

modele matematice de asistare a deciziilor pentru a răspunde în timp real la modificările care au loc de-a lungul lanțului de aprovizionare și distribuție. Această deficiență va fi probabil atenuată de a doua generație de ERP, care încorporează capacități de asistare a deciziilor (Airinei, 2006).

Arhitectura client-server

Marea majoritate a sistemelor de tip groupware și workflow din zilele noastre au fost create pe baza arhitecturii client-server (Chaffey, 1998, citat de Fredman, 1999). Această arhitectură implică o serie de clienți, de obicei calculatoare de tip desktop, care sunt conectate la un calculator de tip server prin intermediul unei *rețele locale* (LAN) sau a unei *rețele extinse* (WAN). Un server este o componentă în sistem care administrează sau controlează resurse particulare din sistem pe care le furnizează altor componente de sistem, acestea fiind numite clienți (care sunt serviți de server).

Arhitectura de tip client-server generează importante beneficii: costuri reduse prin folosirea resurselor partajate (imprimante, memorie etc.), creșterea posibilității de a partaja date și îmbunătățirea posibilităților de comunicare.

Există numeroase cai alternative prin care se pot implementa arhitectura client-server. Distribuirea procesării și a datelor poate fi concentrată mai mult către server sau mai mult către client. În funcție de strategia aleasă, clientul poate fi caracterizat adesea ca fiind slab (“thin”) când procesarea și cantitatea de date este foarte mică la client și mare la server sau gras („fat”) când procesarea și cantitatea de date este foarte mică la server și mare la client.

Sistemele de comunicații unificate

Comunicațiile unificate (CU) reprezintă convergența serviciilor de voce, video și comunicații de date în cadrul unui infrastructuri partajate, bazate pe protocolul IP.

Majoritatea organizațiilor folosesc deja aceste tehnologii. Acestea au instalat în decursul timpului tipuri diferite de servicii folosind furnizori și tehnologii care au la bază diferite standarde și modele de implementare. Acest model bazat pe subsisteme independente a condus la folosirea multiplelor sisteme de comunicații, interconectate într-un mod imprecis și care pot executa funcții de la arhitectura hardware până la echipamentele de tip utilizator final. (Bîzoi, Suduc și Filip, Riscurile utilizării sistemelor de comunicații unificate, 2009).

Valoarea de bază a sistemelor de comunicații unificate nu provine de la caracteristica de integrare foarte strânsă a aplicațiilor, ci mai degrabă din numărul mare al acestora. Deoarece CU este încă o tehnologie curs de formare, mediul de afaceri poate culege recompense mai mari prin utilizarea aplicațiilor de colaborare și a schimbului de cunoștințe decât prin investirea într-o soluție integrată (***, Building toward a unified communications strategy, 2009).

Comunicațiile unificate nu sunt un sistem sau un produs care poate fi pur și simplu cumpărat și apoi instalat (Simpson, 2009). De fapt, reprezintă o soluție de interconectare a unor sisteme de comunicații diferite într-o singură infrastructură integrată și totodată reduce personalul suport al acelor sisteme separate, micșorând dramatic costurile

Conform lui Bailey (2007) citat de Bîzoi, Suduc și Filip în Riscurile utilizării sistemelor de comunicații (2009), comunicațiile unificate cuplează tehnologii și procese pentru a permite organizației să dobândească capacități și viteză pe care nu a le-a avut anterior.

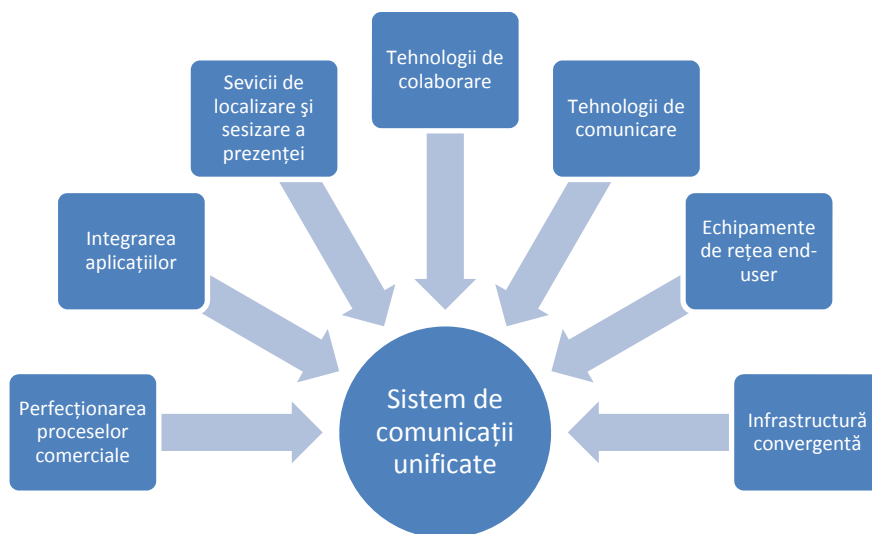


Figura 4. SCU – integrator de tehnologii

În Figura 4 este prezentată tendința convergentă a tehnologiilor existente de a se integra într-o infrastructură unificată de comunicații.

2.4. Construirea SSD

Scopul acestui subcapitol este de a prezenta o serie de **metode și strategii** pentru construirea Sistemelor Suport pentru Decizii. Din punct de vedere al realizării, între subclasa

Sistemelor Suport pentru Decizii Bazate pe Comunicații și clasa Sistemelor Suport pentru Decizii nu sunt diferențe semnificative, ceea ce conduce la ideea că **principiile generale de realizare** se aplică în același mod. Au fost prezentate **principii pentru proiectarea SSD, strategiile de abordare** (descendentă, ascendentă și mixtă) aplicabile în fazele de proiectare a unui SSD și **strategii de realizare (prototipizarea, paradigma secvențială, paradigma evolutivă)** utilizate în fazele de construire propriu-zisă a SSD. De asemenea, au fost prezentate două **abordări de realizare (participativă și fenomenologică)** referitoare la influența utilizatorilor asupra modelului aplicației. În ultima parte a acestui subcapitol sunt prezentate pe scurt **Sistemele Suport pentru Decizii Bazate pe Web (SSDBW)** prin **accentuarea avantajelor și dezavantajelor** folosirii acestora și considerații privind **evaluarea Sistemelor Suport pentru Decizii Bazate pe Comunicații**.

Principii pentru proiectarea SSD

Au fost formulate zece principii pentru proiectarea Sistemelor Suport pentru Decizii, după cum urmează (***, 10 Guiding Principles for the Design of Computer-Based Decision-Support Systems):

1. *Accentuarea parteneriatului om-calculator.* Un SSD de succes este acela care asistă și nu înlocuiește decidentul uman.
2. *Cooperativ și distribuit.* Problemele complexe de mediu implică în mod normal părți care colaborează din locații răspândite pe suprafețe întinse și utilizează resurse informaționale care sunt dispersate în mod egal.
3. *O arhitectură deschisă.* Componentele sistemului probabil se vor schimba în timp prin modificări, înlocuiri, ștergeri sau extinderi.
4. *Instrumente și nu soluții.* Sistemul suport pentru decizii ar trebui proiectat ca un set de instrumente și nu ca un set de soluții la un set de probleme predeterminat.
5. *Reprezentare internă la un nivel înalt.* Abilitatea unui SSD de a avea un anumit nivel de înțelegere a semnificației informației pe care o procesează este cea mai importantă condiție pentru un mediu de rezolvare a problemelor de cooperare și colaborare.
6. *Cunoaștere încapsulată.* Sistemul suport pentru decizii ar trebui să fie un sistem bazat pe cunoaștere.
7. *Descentralizarea luării deciziei.* Sistemul suport pentru decizii nu are nevoie și nu ar trebui să exercite un control centralizat în mediul luării deciziei.
8. *Accentuarea identificării conflictului.* Sistemul suport pentru decizii ar trebui să se concentreze asupra identificării decât asupra rezolvării automate a conflictului.
9. *Interfața calculator-utilizator.* Interacțiunea este facilitată de două caracteristici ale sistemului: o reprezentare a obiectelor la nivel înalt și o interfață utilizator intuitivă. Interfața cu utilizatorul ar trebui să fie grafică.

10. *Integrarea funcțională.* În sistemele suport pentru decizii distribuite și cooperative nivelul necesar de integrare are potențialul de a fi obținut din moment ce modulele funcționale și resursele informaționale sunt componente partajate.

Evaluarea Sistemelor Suport pentru Decizii Bazate pe Comunicații (SSDBC)

Implementarea SSDBC într-o organizație tradițională sau implementarea acestor instrumente pentru a crea organizații virtuale sau structuri organizaționale inovative reprezintă o decizie tehnologică majoră. Deoarece aceste instrumente sunt achiziționate și doar instalate de personalul sistemului informatic, evaluarea se concentrează pe produsele aflate la vânzare.

Power (Decision support systems: concepts and resources for managers, 2002) a identificat șase criterii care ar trebui să fie luate în considerare când se evaluează orice SSDBC:

- ✓ *Încrederea.* Multe organizații doresc o soluție care a dovedit că întrunește toate cerințele pentru a răspunde nevoilor acestora.
- ✓ *Costurile.* Având în vedere costul semnificativ al tehnologiilor și apariția rapidă a unor noi tehnologii, organizațiile doresc un pachet accesibil ca preț.
- ✓ *Scalabilitatea.* Companiile au nevoie de un pachet software care să se poată integra cu aplicațiile software și infrastructura hardware existente.
- ✓ *Securitatea.* Odată cu creșterea nivelului de date partajate, crește și îngrijorarea legată de securitatea acestor date.
- ✓ *Facilitățile de proiectare.* Este important pentru multe organizații ca pachetul standard de aplicații să poată fi personalizat în funcție de necesitățile acestora.
- ✓ *Ușurința în instalare și utilizare.* Managerii doresc un pachet software care este ușor de instalat și necesită un volum redus de pregătire al utilizatorilor.

Capitolul 3. Arhitectură pentru SSD Bazat pe Comunicații

3.1. Arhitecturi pentru sisteme informatice

Scopul acestui subcapitol este de a expune o viziune teoretică asupra **ciclului de viață a unui sistem informatic din perspectivă arhitecturală**. Sunt prezentate **fazele procesului de realizare al unei arhitecturi**, după care se detaliază **procesul de proiectare al unei arhitecturi**. Pornind de la **înțelegerea problemei**, sunt **identificate elementele de proiectare și relațiile dintre ele**, sunt enunțate câteva considerații privind **evaluarea arhitecturii și transformarea acesteia**. În ultima parte, sunt identificate o serie de **probleme care apar în proiectarea arhitecturii software** și sunt enumerate **sarcini și activități de proiectare**.

Perspectiva arhitecturală furnizează puncte de vedere diferite dar complementare asupra managementului software-ului și proiectării ingineresti. Rolul tradițional al arhitectului de sistem/software este să ajute beneficiarul să înțeleagă nevoile sale în întregime și cu exactitate și să creeze un concept arhitectural care va conduce la un sistem fezabil de construit pe baza tehnologiei, resurselor și a timpului disponibile. De asemenea, arhitectul de sistem va supraveghea construcția produsului/sistemului. Perspectiva arhitecturală a ciclului de dezvoltare software este centrată pe proiectarea aplicației sau sistemului și pe modul în care proiectul conduce la realizarea acesteia. Aceasta poartă numele de arhitectură bazată pe construirea software (Sewell și Sewell, 2002).

Cele patru faze ale procesului de realizare al arhitecturii sunt: (a) Faza de pre-proiectare; (b) Analiza domeniului; (c) Proiectarea schematică; (d) Faza de dezvoltare a proiectului.

Aceste faze sunt executate secvențial, dar ca și ingineria software și în cazul fazelor de inginerie ale proiectului nu este absolut necesar ca acestea să aibă loc într-un singur pas secvențial. Cele patru faze de mai sus când sunt combinate cu următoarele faze formează o metodă de arhitectură bazată pe construcția software: (a) Faza de elaborare a documentelor proiectului; (b) Faza de contractare sau angajare de personal; (c) Faza de construcție; (d) Faza post-construcție.

În Figura 5 sunt prezentate fazele realizării unei arhitecturi software din perspectiva arhitecturală a ciclului de viață al unui produs informatic.

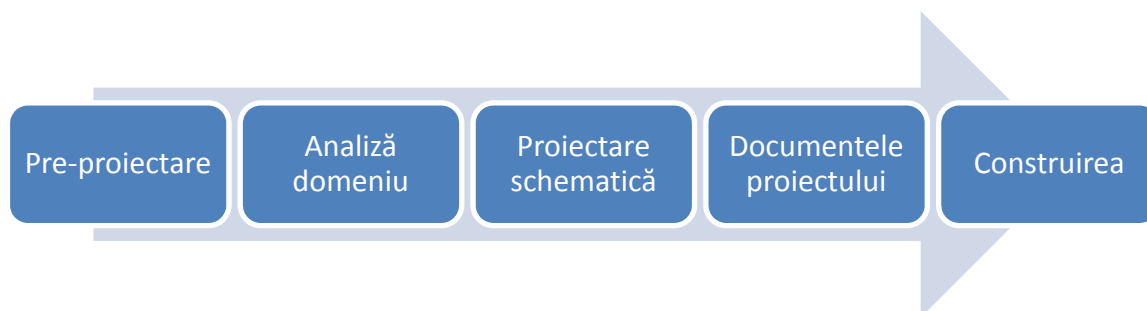


Figura 5. Fazele realizării unei arhitecturi.

3.2. Stabilirea cerințelor de proiectare

În cadrul stabilirii cerințelor de proiectare, un aspect important este ocupat de **elaborarea și adoptarea deciziilor în grup sau în organizație**. În aceste condiții, au fost identificate o serie de potențiale **avantaje și dezavantaje ale lucrului în grup**. De asemenea, susținerea lucrului în grup poate fi făcută prin intermediul unor **tehnici**, în acest subcapitol fiind amintite: **brainstorming-ul**, **tehnica grupului nominal** și **metoda Delphi**. Nu în ultimul rând, au fost prezentate **caracteristicile Sistemelor de Asistare a Deciziilor Multiparticipant (SADM)**, prin evidențierea **avantajelor și dezavantajelor implicării mai multor participanți** în procesul de luare al deciziei.

Elaborarea și adoptarea deciziilor în grup sau luarea deciziilor în organizație poate fi descrisă ca un proces de luarea a deciziilor unde sunt implicați mai mulți decidenți. Luarea deciziei în grup diferă de activitatea decizională care implică un singur decident din moment ce negocierea deciziei trebuie să aibă loc între decidenți, exceptând cazul în care ei au exact aceeași opinie în ce privește decizia care trebuie luată.

Simon (1957) considera că influența unui context organizațional este foarte importantă în acest caz. Organizația reprezintă un tipar complex de comunicare și alte relații într-un grup de oameni. Acest tipar furnizează fiecărui membru al grupului multe informații, presupuneri, scopuri, atitudini care influențează decizia sa și îi furnizează cu un set stabil și inteligibil de așteptări referitor la ce fac alți membri ai grupului și cum vor reacționa ei la ce spune și ce face.

Conform lui Beach (1997), este fundamental pentru înțelegerea luării deciziei în grup sau la nivel organizațional, mai întâi să înțelegem cum un mediu organizațional formează construcția unor interpretări sociale partajate ale evenimentelor. Acesta considera că în organizație există o gândire comună partajată de mai mulți membri ai organizației care le permite acestora să lucreze împreună și să comunice despre evenimentele apărute și scopurile comune. De fapt, această înțelegere partajată dezvoltă organizațiile. Fără ea, nu ar mai fi o organizație în nici un sens real.

Potențiale avantaje și dezavantaje ale lucrului în grup

Turban și Aronson (1998), se referă la grupuri ca fiind entități compuse din doi sau mai mulți (uzual până la 25) oameni a căror misiune este să execute anumite sarcini și să se comporte ca o singură unitate. Cei doi autori identifică potențialele avantaje și dezavantaje (Tabelul 2).

Tabelul 2. Potențialele avantaje și dezavantaje ale lucrului în grup

Potențiale avantaje	Potențiale dezavantaje
1. Grupurile înțeleg problemele mai bine decât indivizii;	1. Gândirea de grup în care indivizi din același grup au tendința să gândească la fel și să supprime noile idei;
2. Oamenii sunt luați în considerare pentru deciziile la care participă;	2. Luarea deciziei în grup este un proces, în general lent, care consumă timp și în care numai un decident individual la un moment dat poate vorbi;
3. Grupurile sunt mai eficiente față de indivizi în găsirea erorilor;	3. Este mult mai dificil să coordonăm munca desfășurată de un grup decât munca desfășurată de un individ;
4. Un grup deține mai multe informații (cunoștințe) decât orice membru al grupului și poate combina aceste cunoștințe pentru a crea unele noi. Astfel, rezultă mai multe alternative, care conduc la o soluție mai bună.	4. Influențe nepotrivite cu privire, de exemplu, la dominația timpului, opinia sau subiectul unuia sau a mai multor decidenți din cadrul grupului;
5. S-ar putea produce efecte de sinergie în timpul rezolvării problemei;	5. Tendința membrilor grupului de a avea încredere în alții cu privire la distribuția muncii în legătură cu decizia;
6. Lucrul în grup ar putea stimula participării în cadrul grupului și al procesului decizional;	6. Tendința îndreptată către o soluție de
7. Membrii grupului își vor încapsula ego-ul	

<p>în decizie, devenind astfel devotați soluției;</p> <p>8. Riscul este echilibrat în condițiile în care grupul are tendința de a modera pe cei care își asumă riscuri mari și de a-i încuraja pe cei conservativi.</p>	<p>compromis rezultă o calitate slabă;</p> <p>7. Existența riscului pentru incompleta analiză a sarcinilor, timp neproductiv, care poate consta în așteptarea participanților să sosească, socializare;</p> <p>8. Costuri mari în luarea deciziei.</p> <p>9. Tendința grupului de a lua decizii mai riscante decât ar trebui.</p>
---	---

Există în principal trei tehnici proiectate să susțină lucrul în grup (Delbecq și alții, 1975); acestea fiind câteodată folosite în legătură cu utilizarea sistemelor suport pentru decizii de grup (Gray, 1994). Cele trei tehnici sunt: brainstorming-ul, tehnica grupului nominal și metoda Delphi.

Brainstorming-ul

Termenul *brainstorming* a apărut prima dată într-o carte scrisă de Osborn (1957). Aceasta a provocat lungi discuții despre acest termen și despre valoarea metodei aplicată într-un grup de luare a deciziei.

Brainstorming-ul este o încercare de a amplifica creativitatea decidenților prin încurajarea schimbului de idei și a discuției creative libere între indivizii implicați. Brainstorming-ul ca tehnică, poate fi folosită și individual, dar este mult mai eficientă când este folosită în grup.

Pentru a elimina riscul de a neglija ideile bune, Osborn (1957) a enunțat patru reguli pentru brainstorming care, în opinia sa, sunt importante de luat în considerare în vederea evitării evaluării premature a ideilor:

1. *Generarea a cât mai multe idei posibile.* Cu cât sunt mai multe idei, cu atât mai bine. Datorită numărului mare de idei, șansele de apariție a ideilor bune vor crește.
2. *Nu se va critica modul de exprimare al ideilor.* Este important să nu se critice noile idei pe parcursul stagiului de generare și exprimare al ideilor de către decidenți. Nerespectându-se această regulă, participanții se vor simți descurajați pentru a veni cu idei noi.

3. *Se vor încuraja ideile diferite.* Deși pare ciudat, diferențele de opinie vor fi încurajate. Astfel, pot fi descoperite idei noi unice, care anterior nu păreau importante.
4. *Construirea pe baza ideilor altora.* Prin construirea pe baza sugestiilor altora și folosirea lor ca sursă de inspirație, pot fi produse noi idei. Utilizarea unor idei vechi ca sursă de inspirație ar trebui să fie încurajată în proces.

Tehnica grupului nominal

După (Turban și Aronson, 1998), tehnica grupului nominal constă dintr-o secvență de activități în procesul de luare a deciziei, cum ar fi: generarea silențioasă a ideilor prin scrierea lor, listarea succesivă a ideilor, discuții referitoare la ideile prezentate, listarea silențioasă și votarea priorităților, o discuție referitoare la aceste priorități și în final o reclasificare și revotare a priorităților.

Prin folosirea tehnicii grupului nominal, decidenții furnizează un forum în cadrul căruia pot dezvolta și scrie idei față în față, dar dezvoltarea ideii devine individuală și independentă (în ce privește vizualizarea și influențarea) față de membrii altui grup. Delbecq și alții (1986) consideră că folosirea tehnicii grupului nominal evită multe din problemele asociate cu brainstorming-ul și conform lor, tehnica nominală de grup este mai eficientă decât brainstorming-ul.

Marele avantaj al tehnicii grupului nominal este acela că evită două probleme cauzate de interacțiunea în grup: anumiți participanți nu își împărtășesc ideile de teamă să nu fie criticați, iar alți participanți se tem de generarea unor conflicte în cadrul grupului și doresc astfel să păstreze un climat calm. Această tehnică elimină problemele prezentate prin minimizarea diferențelor și asigurarea unei egalități relative în ce privește participarea.

Un alt avantaj îl reprezintă numărul mare de idei produse, precum și ideea-concluzie care încheie procesul decizional; această idee nefiind prezentă la alte metode de grup mai puțin structurate. De asemenea, în multe cazuri, timpul de desfășurare al procesului decizional poate constitui un avantaj.

Dezavantajul major al metodei îl constituie lipsa de flexibilitate: aceasta se ocupă cu o singură problemă la un moment dat. Participanții trebuie să simtă confortabil în cadrul grupului și să fie de acord cu modul de structurare.

Această metodă nu implică spontaneitatea intervențiilor participanților. Timpul câștigat în cadrul procesului decizional poate fi anulat de timpul necesar pentru pregătirea activităților. Moderatorul / facilitatorul trebuie să planifice foarte atent desfășurarea activităților.

Un alt dezavantaj îl constituie faptul că, în cadrul procesului de votare, e posibil ca ideile să nu convergă, încercarea de structurare să ducă la apariția constrângerilor, iar întreg procesul să pară mecanic.

Metoda Delphi

Clayton (1997) considera că metoda Delphi este similară în multe privințe cu tehnica grupului nominal, dar are caracteristici care nu se găsesc în această tehnică sau în brainstorming. Prima caracteristică se referă la generarea ideii, care în această metodă se face nu numai individual și independent, dar și izolat și în mod anonim de către fiecare decident implicat.

O caracteristică secundară reprezintă comunicația dintre indivizi care în această abordare este supravegheată de un moderator și se desfășoară prin utilizarea chestionarelor scrise și al rapoartelor de feedback (Filip, 2002).

Un avantaj important este oferit de metoda Delphi comparativ cu tehnica grupului nominal prin furnizarea unui canal de comunicare unde indivizii pot participa fără a se întâlni fizic. Astfel, se reduc costurile în timp și bani în ce privește nevoia de a călători, deseori la distanță. Întâlnirile față în față sunt de multe ori necesare la un grup de lucru pentru luarea deciziilor, iar acestea au avantajul alegerii unei alte metode de lucru (tehnica nominală de grup, brainstorming etc.).

Decidenții participă sub acoperirea anonimatului în procesele Delphi. Aceasta este o cerință esențială a metodei. Anonimitatea are un avantaj important în aceea că reduce anumite comportamente social-emoționale deseori întâlnite când se folosesc alte metode și care pot distorsiona procesul de luare a deciziei și conduce la obținerea unor rezultate inferioare.

Caracteristicile deciziilor multiparticipant

După Filip (2005), decizia este rezultatul unor activități conștiente, specifice omului, care constau în acumularea crearea și prelucrarea de cunoștințe în cadrul procesului de rezolvare a unei probleme de alegere dintre mai multe alternative identificate sau proiectate

anume, în vederea efectuării de acțiuni care implică alocarea unor resurse, în scopul realizării unor obiective. O serie de autori au remarcat, de multă vreme, necesitatea considerării deciziilor de grup (denumite și "de tip multiparticipant").

Avantajele implicării mai multor participanți în elaborarea și adoptarea deciziilor sunt numeroase și diverse:

1. Bagajul de cunoștințe al grupului este în mod evident mai bogat decât al oricărui participant component al grupului, care, la rândul său, are posibilitatea și este stimulat să dobândească mai multe elemente de cunoaștere de la ceilalți participanți;
2. Grupul are performanțe superioare în ceea ce privește calitatea soluției și poate detecta mai ușor eventualele erori;
3. Membrii grupului se simt coautori ai soluției adoptate și, în consecință, o vor sprijini și, dacă e cazul, se vor angaja în transpunerea acesteia în execuție.

Limitele și dezavantajele implicării mai multor participanți în elaborarea și adoptarea unei decizii sunt (Filip 2005):

1. Performanța grupului poate să fie afectată negativ de o planificare necorespunzătoare și de nerespectarea agendei de lucru;
2. Unii membri ai grupului tind să se alinieze la părerea altora, din cauză că, fie își pierd interesul, fie că se tem să exprime păreri discordante, sau care ar putea „încinge spiritele” (aceasta poate conduce la o gândire de grup, într-o adunare dominată de o personalitate sau de o coaliție prea puternică);
3. Monopolizarea discuțiilor de un număr restrâns de persoane poate cauza blocaje;
4. Se pot manifesta tendințe de adoptare comodă (sau, cu orice preț, prin consens) a unor soluții de compromis, care, uneori, nu sunt și de calitate;
5. Supraîncărcarea informațională a participanților poate conduce la pierderea atenției sau la ignorarea aspectelor esențiale;
6. Sunt posibile pierderi de informație cauzate de receptarea greșită a intervențiilor orale, omisiuni și distorsiuni de consemnare în documentele întâlnirii (procese verbale, minute);
7. Se produce un consum exagerat de resurse (timpul pierdut în dezbateri sterile , în divagații, sau în activități sociale conexe, costurile ridicate pentru organizarea și desfășurarea unor întâlniri „față în față”).

3.3. Proiectarea modelului SSD experimental

Pentru proiectarea modelului experimental a fost ales **limbajul UML** (Unified Modeling Language). Prima parte a acestui subcapitol expune **principalele caracteristici** ale acestui limbaj: **aplicabilitatea, arhitectura UML, părțile principale și tipurile de diagrame. Descrierea modelului experimental** începe cu **diagrama arhitecturii generale** a acestuia și continuă cu **arhitectura software organizată pe pachete**. Fiecare pachet software conține **una sau mai multe clase** și sunt grupate în funcție de **modul de susținere al activității decizionale**: activitate principală sau activitate suport. Au fost descrise în continuare **funcțiile fiecărui pachet** software, prin detalierea **funcțiilor claselor componente**.

*Limbajul de modelare vizuală UML*¹ (Unified Modeling Language) se bazează pe un set de notații grafice care ajută la descrierea și proiectarea sistemelor informatice, în particular al celor dezvoltate folosind tehnica programării orientate pe obiecte. UML ajută la specificarea, vizualizarea și documentarea modelelor sistemelor informatice. De asemenea, limbajul UML poate fi folosit la descrierea altor sisteme care nu sunt informatice (de exemplu se poate modela un proces de afaceri, fără a avea ca scop obligatoriu crearea unui sistem informatic).

În Figura 6 este descrisă arhitectura generală a modelului experimental propus, folosind limbajul UML. Acesta este compus dintr-un model al unui sistem de asistare al deciziilor, o interfață web, o bază de date și un sistem de fișiere pentru stocarea documentelor.

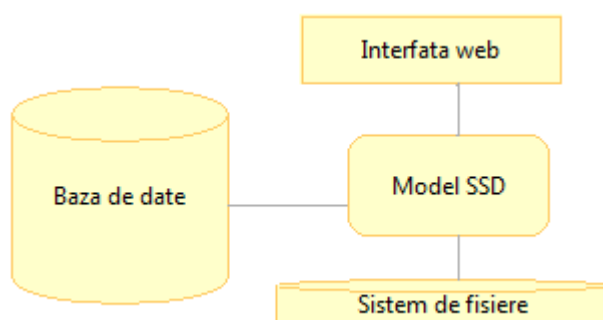


Figura 6. Arhitectura generală a modelului experimental

¹ <http://www.uml.org/>

În continuare se va descrie modelul sistemului suport pentru decizii inspirat din soluția GroupSystems (Filip, 2008). În Figura 7 este prezentată arhitectura modelului experimental organizată pe pachete software.

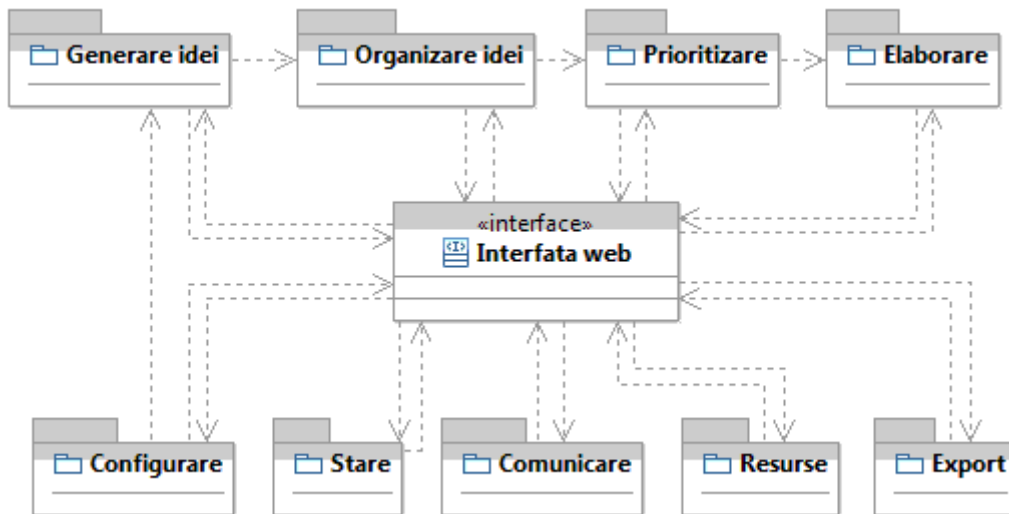


Figura 7. Arhitectura modelului experimental pe pachete

Pachetele software prezentate în arhitectura modelului experimental pot fi grupate astfel: pachete care susțin activitatea decizională de bază (Generarea de idei, Organizarea de idei, Prioritizarea, Elaborarea) și pachete care susțin activitatea suport a activității decizionale (Configurare, Stare, Comunicare, Resurse, Export).

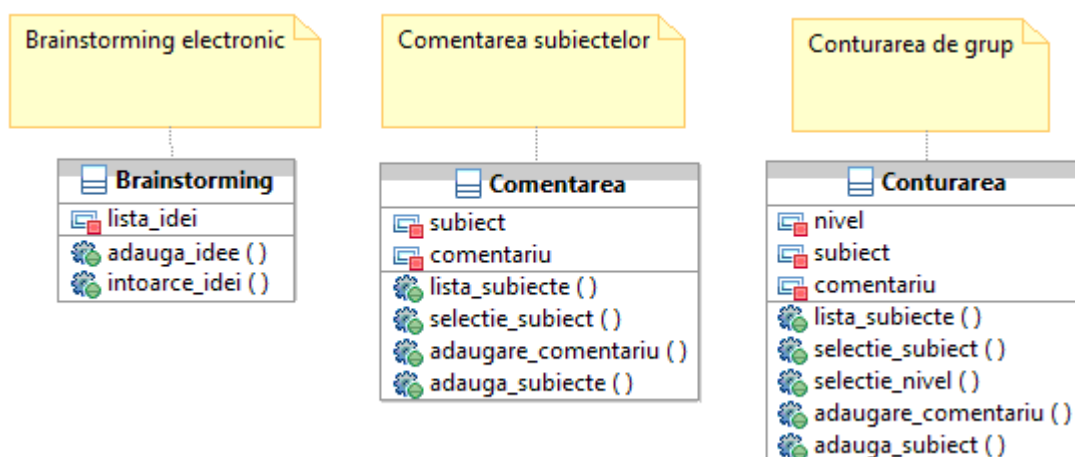


Figura 8. Pachetul pentru generarea ideilor

Figura 8 prezintă pachetul folosit pentru generarea ideilor, prima fază din cadrul procesului decizional. Acest pachet cuprinde trei clase (*Brainstorming*, *Comentarea*, *Conturarea*) ce vor fi utilizate, la alegere, în funcție de configurația sistemului.

Prima clasă este folosită pentru generarea unei liste de idei folosind metoda brainstorming. Astfel, clasa *Brainstorming* va descrie o instanță care va fi folosită pentru a adăuga idei într-o listă de idei. De asemenea, instanța returnează lista ideilor.

Clasa *Comentarea* creează o instanță cu ajutorul căreia fiecare participant are acces la o listă de subiecte în vederea introducerii comentariilor proprii. Instanța va returna o listă de subiecte, va selecta un subiect anume pentru a putea adăuga un comentariu și bineînțeles, va permite adăugarea unui subiect nou.

Clasa *Conturarea* permite crearea unei instanțe care servește prezentarea subiectelor sub formă arborescentă. Structura arborescentă este realizată prin folosirea subiectelor structurate pe baza unor niveluri. La fiecare subiect pot fi adăugate comentariile în mod ordonat.

Ideile generate de pachetul anterior vor fi prelucrate de clasele din pachetul Organizare, prezentat în Figura 9. Acesta conține două clase care vor crea instanțe în funcție de configurația sistemului.

Clasa *Grupare* este folosită pentru a crea o instanță care va permite crearea de grupuri de idei. Acestea vor fi stabilite de moderatorul sistemului. În momentul în care lista grupurilor de idei a fost creată, se pot grupa ideile generate la faza decizională anterioară.

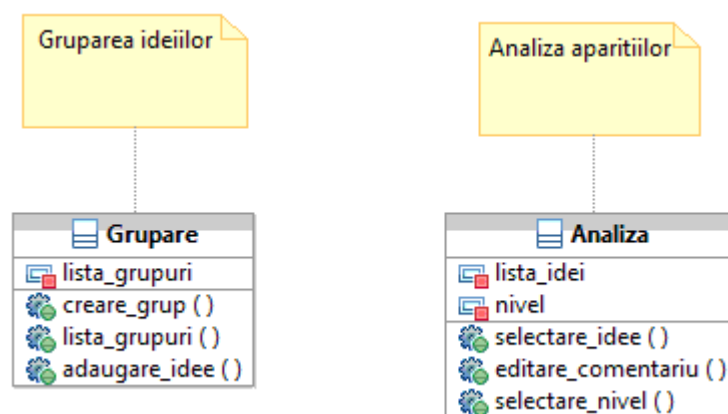


Figura 9. Pachetul pentru organizarea ideilor

Participanții pot să identifice aparițiile celor mai importante idei și să finalizeze comentariile, cu ajutorul clasei *Analiza*.

În Figura 10 sunt prezentate clasele pachetului Prioritizare (*Votare*, *Chestionar*, *Dictionar*). Acestea sunt folosite pentru a obține gradul de importanță al fiecărei idei apărute.

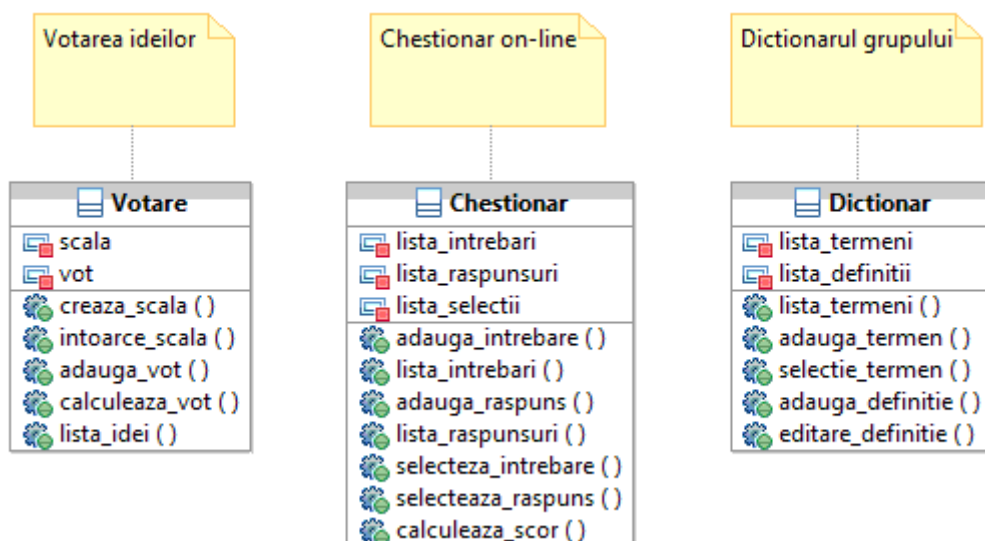


Figura 10. Pachetul pentru prioritizarea ideilor

Clasa *Votare* creează o instanță care permite votarea ideilor. Instanța permite de asemenea stabilirea modului în care se realizează votarea (prin da/nu, note etc.). Clasa *Chestionar* permite moderatorului realizarea chestionarelor on-line pentru sintetizarea răspunsurilor introduse on-line de participanți. Instanța clasei *Dictionar* permite crearea interactivă a definițiilor pentru elementele utilizate în procesul decizional.

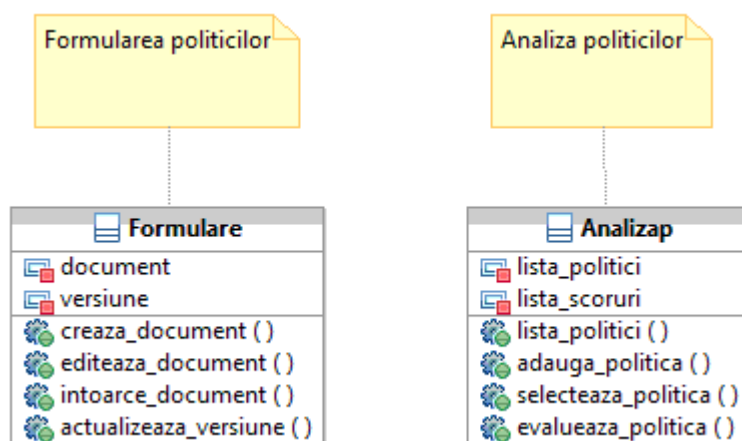


Figura 11. Pachetul pentru formularea politicilor

Clasele care compun pachetul software pentru formularea politicilor sunt prezentate în Figura 11. Cele două clase vor fi folosite alternativ în funcție de configurația sistemului. Instanța clasei *Formulare* permite crearea documentelor referitoare la politici sau misiuni, prin elaborarea acestora în comun de către participanți. Clasa *Analizap* creează o instanță prin care se evaluează în mod sistematic politicile.

Înainte de începerea fazelor decizionale, aplicația necesită anumiți parametri de configurare. Acest pas se realizează prin intermediul instanțelor claselor descrise în pachetul Configurare (Moderator, Participant, Generare), din Figura 12.

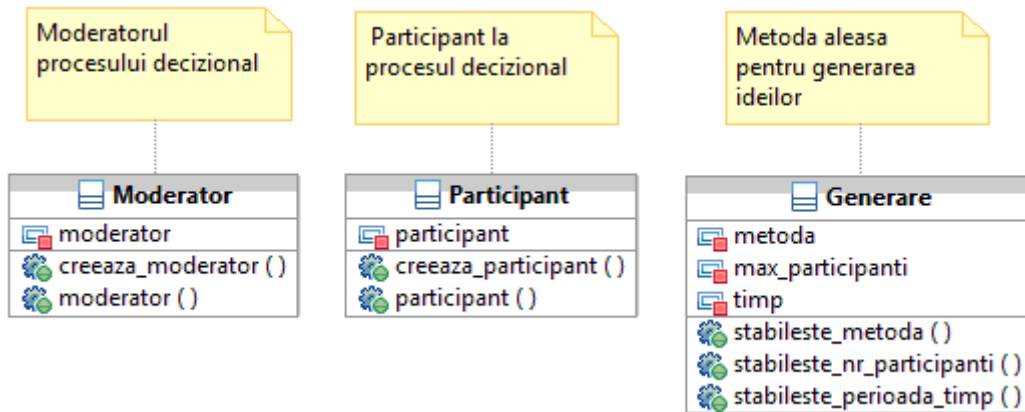


Figura 12. Pachetul pentru configurarea sistemului

Instanțele claselor *Moderator* și *Participant* determină tipul de utilizator care utilizează sistemul. Clasa *Generare* creează o instanță care va stabili parametri generali de sistem (metoda aleasă de generare a ideii, numărul de participanți, timpul pentru anumite faze decizionale).

Capitolul 4. Construirea SSD Bazat pe Comunicații

4.1. Construirea modelului experimental

În vederea construirii modelului experimental a fost propusă **strategia orientată obiect**. Au fost expuse **principiile de bază ale programării orientate pe obiecte** și s-au prezentat o serie de **avantaje** oferite de acestea. Deoarece idea de construire a aplicației presupune o separare a elementelor de programare (aplicația model) de interfață, se prezintă **modelul MVC** (*Model-View-Controller*) și sunt punctate câteva avantaje și dezavantaje în folosirea acestuia. **Limbajul Perl** a fost ales pentru a demonstra funcționalitatea modelului experimental, iar interfața aplicației (bazată pe web) a fost propusă pentru dezvoltare utilizând **standardul CGI** (*Common Gateway Interface*). După stabilirea tehnologiilor propuse pentru construirea SSDBC, se prezintă **metoda iterativă de dezvoltare** a unui Sistem Suport pentru Decizii, prin evidențierea avantajelor alegerii acesteia.

În vederea construirii sistemelor informatice pot fi identificate mai multe strategii de dezvoltare. Printre acestea se numără: strategia descompunerii funcționale, strategia fluxului de date, strategia orientată pe structura datelor și strategia orientată obiect.

Abordarea structurală specifică strategiei orientate obiect presupune un caracter conceptual pronunțat care diminuează distanța semantică dintre modelul sistemului și realitate. Interacțiunea redusă dintre obiecte și coeziunea puternică obținută prin încapsulare și polimorfism permite o localizare mai bună a modificărilor, conducând la un nivel de risc scăzut al efectelor neașteptate.

Unul dintre beneficiile cele mai importante ale programării orientate obiect este reutilizabilitatea. Posibilitatea de a utiliza codul creat de alt programator (mai ales cel din biblioteca de clase) și de a scrie propriile clase reutilizabile, e caracteristica care face programarea orientată obiect mai productivă decât programarea convențională (Filip, 1999).

Modelul MVC

Modelul MVC (*Model-View-Controller*) a fost conceput la sfârșitul anilor 1970 și are cel mai important rol în istoria dezvoltării interfețelor utilizator orientate obiect. Modelul are la bază descompunerea aplicațiilor prin separarea elementelor de programare a interfeței utilizator de cele ale aplicației reale.

Idea de bază a arhitecturii MVC este că interfața cu utilizatorul trebuie să fie separată de aplicația în sine (de funcționalitate). Această premisă permite dezvoltarea lor separată și, cel mai important, permite conectarea cu ușurință a unei noi interfețe la o aplicație existentă. Permite, de asemenea, reutilizarea unor componente ale unei interfețe existente într-o altă aplicație. O aplicație poate folosi și fără interfața sa, eventual de o altă aplicație. Toate aceste aspecte sunt legate de modularitatea, reutilizabilitatea și încapsularea specifice programării orientate obiect.

Limbajul Perl și interfața CGI

Perl (Limbaj Practic pentru Extragere și Rapoarte) este un limbaj interpretat. Interpretorul Perl este un program cărui i se furnizează o listă de comenzi care constituie programul Perl. Deoarece interpretorul citește și execută comenzile Perl, programatorii numesc adeseori programele Perl script-uri (scenarii).

Odată cu nașterea *World Wide Web*, utilizarea Perl a explodat. *Common Gateway Interface* (CGI) furnizează un mecanism simplu de a transfera date de la serverul web la un alt program și să returneze rezultatul interacțiunii programului ca o pagină web. Astfel, Perl a devenit rapid limbajul dominant pentru programarea CGI.

CGI (*Common Gateway Interface*) este un standard care permite programelor diferite de pe site-ul Web să interacționeze cu utilizatorii care vizitează site-ul. Deoarece este un standard, nu este dependent de clientul web, nici de serverul web și poate fi mutat pe o altă mașină în condiții de maximă funcționalitate.

Marele avantaj al folosirii CGI îl reprezintă crearea contextului dinamic, fără ca programele să fie întotdeauna interactive. Se pot folosi scripturi neinteractive pentru a furniza informații dinamice, care nu necesită informații introduse de utilizator (Bîzoi, Suduc și Gorghiu, *Hybrid Method to Design Multi-Language Web Sites*, 2009).

Metoda iterativă de dezvoltare

Metoda iterativă de dezvoltare este o metodă care implică un dialog permanent între dezvoltator și utilizator, utilizatorul fiind implicat în dezvoltarea sistemului, iar dezvoltatorul în utilizarea sistemului. O porțiune din sistemul suport pentru decizii este construit rapid, apoi testat, îmbunătățit și dezvoltat în pași sistematici.

Procesul de elaborare este structurat în mai multe cicluri, câte unul pentru fiecare dezvoltare de sub-sistem. Metoda descrisă de Sprague și Carlson (1982) constă din următorii pași:

- Proiectantul și utilizatorul definesc împreună o sub-problemă care va reprezenta începutul dezvoltării sub-sistemului. Această sub-problemă trebuie să fie mai puțin importantă ca dimensiune, delimitată clar, dar suficient de importantă ca utilitate pentru decident;
- În același timp, problema este analizată și un prototip este elaborat cu ușurință. Acest prototip trebuie să includă funcționalitățile principale ale sistemului;
- Sub-sistemul este utilizat și evaluat adăugând noi reprezentări, modele și structuri de control după fiecare ciclu de dezvoltare.

De principiu, prima fază a procesului de dezvoltare iterativ este similar cu procesul clasic până în momentul în care primul prototip este creat. Pornind din acest moment, prototipul este dezvoltat constant până când sistemul final este creat (Marakas, 2003).

Sistemul de dezvoltare progresiv implică pe lângă avantajele prezentate mai sus și o cooperare strânsă a categoriilor de actori implicați în construcția sistemului suport pentru decizii. Acesta permite utilizatorilor mai activi implicarea în mare măsură în dezvoltarea SSD.

Această metodă are de asemenea, avantajul de a permite o evaluare constantă a sistemului și nu numai o evaluare la sfârșitul realizării sistemului, cum este cazul metodei clasice. Acest tip de sistem orientat pe utilizator este flexibil și permite crearea de noi versiuni pentru a corecta diferite probleme apărute sau pentru a adăuga noi opțiuni (Suduc și Bîzoi, 2008).

4.2. Demonstrarea funcționalității modelului experimental

Scopul acestui subcapitol este de a prezenta elemente care să conducă la concluzia că Sistemul Suport pentru Decizii Bazat pe Comunicații (denumit *Allego*) propus în capitolele anterioare, va fi **funcțional** în urma etapei de construire. Inițial, sunt prezentate o **serie de principii** care trebuie luate în considerare la construirea SSD, după care sunt enumerate principalele **caracteristici ale aplicației Allego**. În continuare, se prezintă **organizarea bazei de date** relaționale și cum s-ar realiza **aplicația model** ținându-se cont de arhitectura descrisă în capitolul 3. În final, se prezintă modul de **generare al interfeței web** prin utilizarea standardul CGI și sunt abordate principalele **aspecte de securitate**.

În vederea construirii unui sistem suport pentru decizii, trebuie urmate o serie de principii (Suduc și Bîzoi, 2008):

- Să existe o abordare globală a problemei care trebuie rezolvată cu ajutorul aplicației ce va fi create;
- Să existe o metodologie unitară pentru proiectarea și dezvoltarea sistemului;
- Să se aplice cele mai noi soluții și tehnici în proiectarea și dezvoltarea sistemelor informatice;
- Sistemul informatic se va realiza în mod independent față de structura organizațională a companiei unde va fi implementat;
- Viitorii beneficiari direcți ai sistemului vor fi implicați în realizarea activităților de analiză, proiectare și implementare a sistemului informatic;
- Dezvoltarea activităților de proiectare se va face în concordanță cu legea și resursele utilizator disponibile;
- Să se prevadă și eventual controla potențialele modificări în cadrul aplicației;
- Să se documenteze eventualele compromisuri moștenite în timpul construirii software.

Principalele caracteristici ale aplicației *Allego* sunt:

- Aplicația este creată pe baza abordării mixte, iar strategia aleasă este cea orientată obiect;
- Metoda de dezvoltare folosită este metoda iterativă;

- Arhitectura aplicației a fost modelată folosind limbajul UML (diagrame de tip clasă). Pentru dezvoltarea aplicației se folosește limbajul Perl în stilul orientat-obiect, tehnologiile software folosite fiind ultimele versiuni stabile la acest moment de timp;
- Aplicația a fost proiectată astfel încât să nu fie specifică unei anumite structuri organizatorice, ci să ofere flexibilitate prin opțiunile de personalizare;
- Datorită faptului că se utilizează metoda iterativă de implementare, potențiali utilizatori sunt implicați în dezvoltarea și testarea aplicației;
- Aplicația a fost proiectată astfel încât în faza de implementare să se folosească numai tehnologii gratuite, iar eventualele implicații juridice să fie cât mai reduse. Utilizatorii au nevoie doar de existența unui calculator și a unui *browser* web, deoarece interfața aplicației va fi web;
- Abordarea, strategia de programare și metoda de implementare au fost alese pentru a putea modifica și extinde cu ușurință aplicația;
- Pentru a funcționa, aplicația necesită existența unor aplicații software, cum ar fi: sistem de operare, server de baze de date, server web, server de poștă electronică, limbajul Perl cu anumite module etc. Lista cerințelor funcționale minime va fi prezentată într-un document separat.

Facilități oferite de aplicația *Allego*:

- Aplicația este multiutilizator și are o interfațată cu web-ul, neexistând nici o restricție conceptuală în ce privește accesul utilizatorilor la aceasta;
- Procesele decizionale ale unor utilizatori diferiți pot fi executate în paralel pe sistem;
- Generarea contextului interfeței se face în mod dinamic;
- Aplicația este interactivă, deoarece folosește formulare web și elemente avansate de interfață;
- Datorită modului de implementare, aplicația poate fi tradusă cu ușurință în alte limbi de circulație internațională, prin folosirea metodei hibride de creare a siturilor web multi-lingvistice (Bîzoi și alții, 2009);
- Poate fi portată cu ușurință pe alte sisteme de operare populare;
- Oferă un grad bun de securitate, care poate fi crescut prin folosirea unor elemente adiționale ale sistemului de operare.

Modelul aplicației *Allego*

Perl este un limbaj de programare care folosește module. Un modul este un pachet (*package*) Perl. Obiectele în Perl se bazează pe referințe la date din cadrul unui pachet. Un obiect în Perl este o simplă referință care cunoaște cărei clasă aparține.

În programarea orientată pe obiecte cu alte limbaje de programare, se declară o clasă și apoi se creează obiecte ale acelei clase. Toate obiectele unei clase particulare se comportă într-un anumit fel, pe baza metodelor din acea clasă. Se pot crea noi clase sau se pot defini altele noi prin moștenirea proprietăților dintr-o clasă existentă.

Există trei afirmații foarte importante pentru a înțelege cum clasele, obiectele și metodele funcționează în Perl: (a) O clasă este un pachet Perl. Acest pachet furnizează metodele pentru obiecte; (b) O metodă este o subrutină Perl. Singurul lucru care trebuie precizat este acela că la aceste metode, primul argument este numele clasei; (c) Un obiect în Perl este pur și simplu o referință la anumite date din cadrul clasei.

Una din facilitățile oferite de programarea orientată obiect o reprezintă moștenirea. Moștenirea oferită de Perl presupune doar moștenirea metodelor, pentru a moșteni date, trebuie să fie create propriile mecanisme.

Deoarece fiecare clasă este un pachet, are propriul spațiu de nume cu propriile tablouri asociative ale numelor de simboluri. Membrii unei clase pot fi adresați începând cu Perl 5 folosind caracterul două puncte dublat (*\$class::\$member*).

La momentul reprezentării modelului folosind limbajul UML, clasele aplicației model au fost grupate în pachete. Deoarece termenul de pachet are altă semnificație pentru limbajul Perl, trebuie precizat că implementarea pachetelor (cu rol de container) din aplicația de modelare s-a făcut sub forma de directoare la nivelul sistemului de fișiere al sistemului de operare.

Astfel, a fost creat directorul *AppModel* care va conține întreaga structură de subdirectoare și fișiere care vor reprezenta modelul aplicației. Pentru a stoca într-un mod organizat clasele (pachetele) Perl au fost create următoarele 9 subdirectoare: *Generare_idei*, *Organizare_idei*, *Prioritizare*, *Elaborare*, *Export*, *Resurse*, *Comunicare*, *Stare*, *Configurare*.

Generarea interfeței web

CGI - *Simple Common Gateway Interface Class* este o bibliotecă care folosește obiectele Perl 5 să creeze formulare web și să prelucreze conținutul lor. Pachetul *CGI.pm* definește obiectele *CGI*, entități care conțin valorile interogării curente sau alte variabile de stare. Folosind metodele *CGI* pot fi examinate cuvinte cheie și parametrii transmiși scriptului și se pot crea formulare ale căror valori inițiale pot fi preluate din interogarea curentă (de altfel și conservarea stării informațiilor).

Modulul furnizează funcții prescurtate care produc cod *HTML*, reducând astfel timpul de editare și apariția erorilor. De altfel, furnizează și anumite funcționalități pentru facilități *CGI* mai avansate, incluzând încărcare de fișiere, cookies, *CSS (Cascading Style Sheets)*, operații cu serverul, cadre etc.

Sunt două stiluri de programare cu *CGI.pm*: un stil orientat obiect și altul orientat pe funcții. În stilul orientat obiect, se pot crea unul sau mai multe obiecte *CGI* și apoi utiliza metode pentru a crea diferite elemente ale paginii. Fiecare obiect *CGI* începe cu lista numelor parametrilor care sunt transmiși scriptului de către server.

Se pot modifica obiectele, se pot salva într-un fișier sau într-o bază de date și se pot recrea mai târziu. Acest lucru este posibil pentru că fiecare obiect corespunde “stării” unui script *CGI* și pentru că lista parametrilor fiecărui obiect este independentă de a altora.

Capitolul 5. Concluzii și perspective

5.1. Contribuții personale

Printre contribuțiile personale ale autorului la tema de cercetare, se pot enumera:

- ✓ Realizarea unui studiu referitor la numărul de publicații din domeniul de cercetare, indexate în patru baze de date internaționale pentru stabilirea contextului activității de cercetare, prezentat în Capitolul 1, subcapitolul 1.3;
- ✓ Realizarea unui studiu sistematic pentru cunoașterea stadiului actual al domeniului de cercetare *Sisteme Suport pentru Decizii*, cu accentuarea clasei *Sisteme Suport pentru Decizii Bazate pe Comunicații*. Astfel, în Capitolul 2, subcapitolul 2.1 se prezintă SSD la modul general, iar în subcapitolul 2.2 modalități de utilizare și implicații ale acestora, în subcapitolul 2.3 tehnologii utilizate de SSD, în subcapitolul 2.4 sunt prezentate metode de proiectare și strategii de construire a SSD;
- ✓ Definierea *Sistemelor Suport pentru Decizii Bazate pe Comunicații* prin prezentarea comparativă a conceptelor: *Sistem Suport pentru Decizii Organizațional*, *Sistem Suport pentru Decizii de Grup*, *Groupware*, *platformă colaborativă*, *videoconferință* (sub-subcapitolul 2.1.5).
- ✓ Reîncadrarea în clasa sistemelor informatice, din punct de vedere tehnologic, a conceptului de *Sistem Suport pentru Decizii Bazate pe Comunicații*, prin prisma apariției și dezvoltării *Sistemelor de Comunicații Unificate* (sub-subcapitolul 2.3.2).
- ✓ Propunerea unei arhitecturi pentru un *Sistem Suport pentru Decizii Bazat pe Comunicații* (prezentată în Capitolul 3) și a unei metode de proiectare a modelului experimental;
- ✓ Propunerea unei strategii de construire a modelului experimental și a unei metode de dezvoltare adaptată aplicației propuse (prezentată în Capitolul 4, subcapitolul 4.1);
- ✓ Demonstrarea funcționalității modelului experimental prin exemplificarea unei porțiuni de cod a aplicației *Allego* (prezentată în Capitolul 4, subcapitolul 4.2);
- ✓ În perioada 2004-2010, autorul a publicat în total peste 50 materiale² care prezintă rezultate ale cercetării din domenii multidisciplinare (articole, studii, cărți) în calitate de prim-autor și coautor. Dintre acestea, relevante pentru domeniul cercetării, se pot enumera:

² <http://ssai.valahia.ro/~bizoi/pub.shtml>

- *Computer Supported Cooperative Work – An Example for Implementing a Web-based Platform Application*, în revista *Studies in Informatics and Control*, volumul 15, numărul 1 din anul 2006 (prim autor);
- *EDUSAN – A Web-based Collaborating Support System For The Institutions Involved In The Health Education And Diseases Prevention*, în *Buletinul Științific al Facultății de Inginerie Electrică*, numărul 1/2006 (coautor);
- *Technical Overview of the VccSSe Web System*, la 8th Conference: Virtual University, Warsaw University of Technology, 18-20 iunie 2008 (prim autor);
- *Hybrid Method to Design Multi-Language Web Sites* - Proceedings of the 1st International Conference on Computer Supported Education, CSEDU 2009, Lisbon (prim autor);
- *Rates on Collaborative Platforms Activity in Multinational Educational Projects* - Proceedings of the 9th WSEAS International Conference on Distance Learning & Web Engineering (DIWEB'09) (prim autor);
- *Riscurile utilizării sistemelor de comunicații*, Editura Academia Română, Seria Probleme Economice, Vol. 327-328, 2008 (prim autor);
- *Riscurile utilizării sistemelor de comunicații unificate*, Editura Academia Română, Seria Probleme Economice, Vol. 372, 2009 (prim autor);
- *An Overview of the DSS Development* - Scientific Bulletin of Electrical Engineering Faculty - 2008, Year 8, No. 1 (coautor);
- *Using Collaborative Platforms for Decision Support* - Proceedings of the 17th International Conference on Control Systems and Computer Science, CSCS-17, Vol. 2, 2009 (prim autor);
- *Interface Architecture for a Web-Based Group Decision Support System* - *Studies in Informatics and Control*, Volume 18, Issue 3, 2009 (coautor);
- *Exploring Multimedia Web Conferencing* - *Informatică Economică*, Volume 13, No. 3, 2009 (coautor);
- *Ethical Aspects on Software Piracy and Information and Communication Technologies Misuse*, *Supplementary Ways for Improving International Stability* - SWIIS, IFAC Workshop, octombrie 28-30, 2009 (coautor);
- *Audit for Information Systems Security*, *Informatică Economică*, Volume 14, No. 1, 2010 (coautor);

- *Risk Assessment of Information and Communication Technology Use in Multinational Educational Projects*, Second World Conference on Educational Sciences (WCES 2010), Istanbul, February 4-8, 2010 (prim autor);
 - *Using Web Conferencing for Disseminating the Educational Projects Results*, Second World Conference on Educational Sciences (WCES 2010), Istanbul, February 4-8, 2010 (coautor).
- ✓ De asemenea, autorul și-a desfășurat activitatea în calitate de specialist în șase proiecte naționale de cercetare, în trei proiecte europene educaționale (având rol de expert tehnic pentru dezvoltarea unor sisteme care facilitau colaborarea și lucrul în echipă) și a participat în anii 2008 - 2009 la *Programul interdisciplinar de prevenire a fenomenelor cu risc major la scară națională* (Program fundamental al Academiei Române), coordonat de domnul acad. Florin Gheorghe Filip.

5.2. Perspective de continuare a cercetărilor

În urma activității de cercetare efectuate, ca urmare a schimbărilor tehnologice rapide și a ritmului de dezvoltare a mediului de afaceri, au fost identificate noi direcții de cercetare, după cum urmează:

- ✓ Nu există o definiție clară a SSD, iar definirea acestei clase de sisteme informatice poate să difere de la autor la autor. Dezvoltarea tehnologică rapidă, apariția unor noi metodologii și abordări ne poate conduce la concluzia că definirea și prezentarea caracteristicilor SSD constituie în continuare un subiect de actualitate;
- ✓ În această lucrare a fost propusă arhitectura unui SSDBC cu interfață web, iar concluzia a fost că încă mai sunt multe teme de studiat. Transformarea acestei arhitecturi într-o arhitectură orientată pe servicii, probabil ar conduce la situații aproape neexplorate în ce privește utilizarea unor astfel de sisteme sau cum influențează activitățile organizațiilor;
- ✓ Din cauza caracterului multidisciplinar al *Sistemelor Suport pentru Decizii*, tehnicile și metodologiile de integrare a tehnologiilor vor avea o influență deosebită asupra acestora. *Sistemele de Comunicații Unificate*, aflate încă la început și prezentate pe scurt în subcapitolul 2.3.2 *Tehnologii pentru SSD Bazate pe Comunicații* vor schimba conceptele referitoare la comunicare și colaborare;

- ✓ Noțiunea de *virtualizare* va schimba infrastructura informațională și de comunicații existentă; în aceste condiții, este necesară o adaptare a sistemelor informatice, inclusiv a SSD;
- ✓ Dezvoltarea rețelelor de socializare, web-ul 2.0 și accesul din ce în ce mai mare al utilizatorilor la Internet au schimbat semnificația noțiunilor de comunicare și colaborare. Se constată o lipsă a datelor concrete cum percep utilizatorii sistemele informatice, cum le influențează stilul de viață și relația de muncă;
- ✓ În ultima perioadă se discută tot mai mult de adoptarea unor măsuri pentru aplicarea unor restricții de bandă în funcție de valoarea abonamentului plătită de client sau de măsuri legislative care să interzică accesul la anumite servicii (de exemplu *Youtube*). Toate aceste măsuri vor avea un impact negativ asupra Internetului și implicit asupra SSDBC și va genera o cerere de noi strategii și abordări.

Concluzii

Sistemele Suport pentru Decizii sunt o clasă a sistemelor informatice care au apărut inițial sub forma unor studii teoretice în anii 1950-1960. Dezvoltarea tehnologică și apariția unor noi paradigme de programare și metodologii a făcut ca *Sistemele Suport pentru Decizii* să fie un domeniu de cercetare activ, care conduce la noi direcții de cercetare neexplorate anterior. După cum a fost prezentat în subcapitolul 1.3. *Contextul activității de cercetare*, cercetătorii publică într-un ritm ascendent articole, cărți, studii și alte materiale care evidențiază rezultate ale cercetării în domeniul SSD.

Dezvoltarea Internetului, a tehnologiilor de comunicații, coroborate cu nevoile mediului de afaceri (mai ales după anii 1990 - 2000) au condus la apariția unor noi clase de *Sisteme Suport pentru Decizii*: *Sistemele Suport pentru Decizii Bazate pe Comunicații (SSDBC)*. SSDBC se regăsesc sub mai multe denumiri, cele mai cunoscute fiind: *Sistemele Suport pentru Decizii de Grup*, *groupware*, *platformele de colaborare* etc. și au ca scop utilizarea rețelei și a tehnologiilor de comunicații pentru a facilita colaborarea, comunicarea și crearea unui suport partajat pentru luarea deciziei.

Studiul prezentat în subcapitolul 1.3. *Contextul activității de cercetare*, ne indică faptul că managerii și cercetătorii sunt în perioada inițială de acumulare a cunoștințelor referitoare la *Sistemele Suport pentru Decizii Bazate pe Comunicații*. Mai sunt multe lucruri

de învățat cum SSDBC interacționează cu echipele de lucru în întâlnirile electronice și cum influențează ieșirile/produsele unei organizații.

SSDBC ajută în comunicarea, colaborarea și coordonarea echipelor de lucru ale organizațiilor. S-a constatat că performanțele grupurilor de lucru și calitatea deciziilor au fost îmbunătățite când la alegerea SSDBC s-a ținut cont de tipul sarcinilor, mărimea grupului și localizarea membrilor acestuia. Rezultate bune au fost obținute și în situațiile în care organizațiile au angajat personal special pentru acest scop sau cei care l-au folosit erau beneficiarii sistemului.

Referințe bibliografice

- ***. (fără an). Preluat pe 2006, de pe 10 Guiding Principles for the Design of Computer-Based Decision-Support Systems: http://www.cadrc.calpoly.edu/pdf/decision_brochure.pdf
- ***. (fără an). *ACM Digital Library*. Preluat pe iulie 26, 2010, de pe The ACM Portal: <http://portal.acm.org/portal.cfm>
- ***. (2009). *Building toward a unified communications strategy*. Preluat pe noiembrie 10, 2009, de pe http://www.adobe.com/products/acrobatconnectpro/webconferencing/pdfs/connectpro_unifiedcommwhitepaper_1_09.pdf
- ***. (fără an). *IEEE Xplore - Home*. Preluat pe iulie 26, 2010, de pe IEEE - The world's largest professional association for the advancement of technology: <http://www.ieee.org/index.html>
- ***. (fără an). *ScienceDirect - Browse Journals and Books*. Preluat pe iulie 26, 2010, de pe ScienceDirect - Home: <http://www.sciencedirect.com/>
- ***. (fără an). *Web of Knowledge - Science - Thomson Reuters*. Preluat pe iulie 26, 2010, de pe ISI Web of Knowledge: <http://www.isiwebofknowledge.com/>
- Airinei, D. (2006). *Sisteme de asistare a deciziilor și depozite de date - notițe de curs*. Preluat pe decembrie 14, 2006, de pe Universitatea A. I. Cuza Iași, Facultatea de Economie și Administrarea Afacerilor: <http://portal.feaa.uaic.ro/C10/Sisteme%20de%20asistare%20a%20deciziil/default.aspx>
- Ancona, D., Chong, C. (1996). Entrainment: Pace, cycle, and rhythm in organizational behavior. *Research in Organizational Behavior*, vol. 18, L. L. Cummings and B. M. Staw (Eds.), 251-284.
- Bailey, S. B. (2007). *Unified Communications: Why and How*. Preluat pe noiembrie 13, 2009, de pe http://www.accenture.com/NR/rdonlyres/48C2F458-D71E-404C-967C-60413BC09C96/56196/114782G_UnifiedComm_PoV_92.pdf
- Banker, R. D., Field, J. M., Schroeder, R. G., Sinha, K. K. (1996). Impact of work teams on manufacturing performance. *Academy of Management Journal*, 29(4), 867-890.
- Beach, L. (1997). *The Psychology of Decision Making, People in Organizations*. California: Sage Publications Inc.
- Bîzoi, M., Gorghiu, G., Suduc, A., Alexandru, A. (2006). Computer Supported Cooperative Work – An Example for Implementing a Web-based Platform Application. *STUDIES IN INFORMATICS AND CONTROL, Volume 15, Number 1*, 71-78.

- Bîzoi, M., Suduc, A. M., Filip, F. G. (2009). Using Collaborative Platforms for Decision Support. *Proceedings of the 17th International Conference on Control Systems and Computer Science (CSCS 17)*, Vol. 2, (pg. 349-352). București.
- Bîzoi, M., Suduc, A. M., Gorghiu, G. (2009). Hybrid Method to Design Multi-Language Web Sites. *Proceedings of the 1st International Conference on Computer Supported Education (CSEDU 2009)*, Vol. 1, (pg. 427-430). Lisabona.
- Bîzoi, M., Suduc, A. M., Gorghiu, G., Gorghiu, L. (2009). Analysis of 1000 Days of Collaborative Activities in Two Multinational Educational Projects. *WSEAS Transactions on Advances in Engineering Education, Issue 10, Volume 6*.
- Bîzoi, M., Suduc, A., Filip, F. G. (2009). *Riscurile utilizării sistemelor de comunicații unificate*. București: Academia Română, Seria "Probleme economice", Vol. 372.
- Bîzoi, M., Suduc, A., Gorghiu, G., Gorghiu, L. M. (2010). Risk Assessment of Information and Communication Technology Use in Multinational Educational Projects. *Second World Conference on Educational Sciences (WCES 2010)* (pg. 2836-2840). Istanbul: Elsevier Ltd. in *Procedia - Social and Behavioral Sciences*, 2(2).
- Brinck, T. (1998). *Groupware: Introduction*. Preluat pe februarie 19, 2009, de pe <http://www.usabilityfirst.com/groupware/intro.txt>
- Chaffey, D. (1998). *Groupware, Workflow and Intranets - Reengineering the Enterprise with Collaborative Software*. Woburn: Butterworth - Heinemann.
- Cioca, M., Cioca, L. (2010). Decision Support Systems used in Disaster Management. În C. S. Jao, *Decision Support Systems* (pg. 371-390). INTECH.
- Clayton, M. J. (1997). Delphi: A technique to harness expert opinion for critical decision-making tasks in education. *Educational Psychology*, 17(4), 373-386.
- Delbecq, A., Van de Ven, A., Gustafson, D. (1986). *Group Techniques for Program Planning: a Guide to Nominal Group and Delphi Processes*. Wisconsin, Greenbriar: Middleton.
- Delbecq, A., Ven, V. d., A., H., Gustafson, D. (1975). *Group Techniques for Program Planning*. Glenview: Foresman and Company.
- Dennis, A., George, J., Jessup, L., Nunamaker, J., Vogel, D. (1998). Information Technology to Support Electronic Meetings. *MIS Quarterly* (12:4), pg. 591-624.
- DeSanctis, G., Gallupe, B. (1987). A foundation for the study of group decision support systems. *Management Science*, 33, 589-609.
- Druzdzel, M. J., Flynn, R. R. (1999). Decision Support Systems. În A. Kent, & M. Dekker, *Encyclopedia of Library and Information Science*.

Druzdzel, M. J., Flynn, R. R. (2002). *Decision Support Systems, Encyclopedia of Library and Information Science, Second Edition*. New York: Editura Allen Kent.

Filip, F. G. (2008). Decision support and control for large-scale complex systems. *Annual Reviews in Control*, 32(1) , 61-70.

Filip, F. G. (2002). *Decizie Asistată de Calculator: Decizii, decidenți - metode și instrumente de bază*. București: Editura Tehnică.

Filip, F. G. (2005). *Decizie asistată de calculator: decizii, decidenți, metode de bază și instrumente informatice asociate. Ediția a II-a*. București: Editura Tehnică.

Filip, F. G. (1999). *Informatică industrială: Noi paradigme și aplicații*. București: Editura Tehnică.

Filip, F. G. (2004). *Sisteme suport pentru decizii*. București: Editura Tehnică.

Fredman, J. H. (1999). *Organizational Decision Support Systems - A Theoretical and Practical Study, Focusing on Group Decision Support, Knowledge Management and Means of Communication in Organizational Decision Support Systems*. Department of Informatics, Göteborg School of Economics, Göteborg University.

George, J. F. (2008). *Handbook on Decision Support Systems 1*. International Handbooks Information System.

George, J. (1991). The Conceptualization and Development of Organizational Decision Support Systems. *Journal of Management Information Systems*, 8(3) , 109-125.

Gorry, G., Scott Morton, M. (1971). A framework for management information systems. *Sloan Management Review*, 13(1) , 55-70.

Gray, P. (1994). *Decision Support and Executive Information Systems*. Englewood Cliffs, New Jersey: Prentice-Hall.

Hackathorn, R., Keen, P. (1981). Organizational Strategies for Personal Computing in Decision Support Systems. *MIS Quarterly*, 5, 3 , 21-26.

Huber, G. P. (1984). Issues in the Design of Group Decision Support Systems. *MIS Quarterly*, Vol. 8, No. 3 , 195-204.

King, J., Star, S. (1990). Conceptual foundations for the development of organizational decision support systems. *Proceedings of the Twenty-third Annual Hawaii International Conference on Systems Sciences*, vol. 3 (pg. 143-151). Los Alamitos, CA: IEEE Computer Society Press.

Kivijärvi, H., Kuula, M. (1996). An Experiment to Apply Some Substance-Theories to the Design and Development of a Corporatewide DSS in a Small Company. *Scandinavian Journal of Information Systems*, 8 .

Little, J. D. (1970). Models and Managers: The Concept of a Decision Calculus. *Management Science*, 16(8) , 35-43.

Marakas, G. (2003). *Decision Support Systems and Megaputer*. Upper Saddle River, New Jersey: Prentice Hall.

Minzberg, H. (1991). Planning on the left side and managing on the right. *Creative Management (Jane Henry, ed.)*. Sage Publications, London , 58-70.

Osborn, A. (1957). *Applied imagination*. New York: Scribner.

Power, D. (2002). *Decision support systems: concepts and resources for managers*. Greenwood Publishing Group.

Sewell, M., Sewell, L. (2002). *The Software Architect's Profession: An Introduction*. Upper Saddle River: NJ: Prentice Hall PTR.

Simon, H., A. (1957). *Administrative behavior: A study of decision-making process in administrative organization*. New York: Free Press.

Simpson, L. (2009). *Is Unified Communications Ready for Prime Time?* Preluat pe noiembrie 13, 2009, de pe <http://www.compucom.com/PublishingImages/whitepapers/UnifiedCommunicationsWPCCv.pdf>

Sprague, R. H., Carlson, E. D. (1982). *Building effective decision support systems*. Englewood Cliffs: N.J., Prentice-Hall.

Suduc, A. M., Bîzoi, M. (2008). An Overview of the DSS Development. *Scientific Bulletin of Electrical Engineering Faculty, Year 8, No. 1* , 99-102.

Suduc, A., Bîzoi, M., Filip, F. (2009). *Aspecte legate de interfețele om-calculator pentru reducerea riscurilor utilizării sistemelor informatice*. București: Academia Română, Seria "Probleme economice", Vol. 369-370.

Suduc, A., Bîzoi, M., Filip, F. (2010). Audit for Information Systems Security. *Informatică Economică, Volume 14, No. 1* , 43-48.

Suduc, A., Bîzoi, M., Filip, F. (2009). Exploring Multimedia Web Conferencing. *Informatica Economica, Volume 13, No. 3* , 5-17.

Suduc, A., Bîzoi, M., Gorghiu, G., Gorghiu, L. M. (2010). Using Web Conferencing for Disseminating the Educational Projects Results. *Second World Conference on Educational Sciences (WCES 2010)* (pg. 2813-2818). Istanbul: Elsevier Ltd. in Procedia - Social and Behavioral Sciences, 2(2).

Turban, E., Aronson, J. (1998). *Decision Support Systems and Intelligent Systems*. London: Prentice-Hall International Inc.

Turban, E., Aronson, J. E. (1995). *Decision Support and Intelligent Systems (5th edition)*. Upper Saddle River: N.J.: Prentice-Hall, Inc.

Turban, E., Aronson, J. E., Liang, T. (2005). *Decision Support Systems and Intelligent Systems, 7th Ed.* USA: Prentice-Hall.

Vogel, D., Nunamaker, J. (1990). Group Decision Support System Impact: Multi-Methodological Exploration. *Information & Management* , 15-28.

Watson, R. (1990). *A Design for Infrastructure to Organizational Decision Making* . Los Alamitos, CA: IEEE Computer Society Press.

Yen, D., Gong, B., Wen, H., Chen, H. (1998). Collaborative Computing: Groupware and Group Decision Support Systems. *Decision Sciences Institute Proceedings, No. 2* , 592-594.