

Modelarea și controlul dialogului om–calculator

Dizertație

ing. Cosmin Munteanu

Departamentul de Calculatoare
Facultatea de Automatică și Calculatoare
Universitatea "Politehnica" din Timișoara
e-mail: cosmin@cs.utt.ro

Conducător: ș.l. ing. **Marian Boldea**

Rezumat

Ca parte a unor cercetări în domeniul sistemelor automate de dialog vocal om-calculator, lucrarea de față realizează o trecere în revistă a stadiului actual în domeniul celei mai importante părți a acestor sisteme - componenta de control al dialogului, precum și o prezentare a unor rezultate personale, cea mai importantă contribuție fiind elaborarea unui limbaj de descriere a dialogului, facilitând atât modelarea cât și controlul lui, și care oferă deci un cadru generic pentru continuarea cercetărilor.

Controlul dialogului om-calculator poate fi considerat o formă de raționament automat care, în contextul unei anumite aplicații și al unei conversații cu un utilizator, urmărește determinarea, la un moment dat, a celei mai adecvate acțiuni a sistemului și anticiparea, sau uneori chiar determinarea, conținutului următoarei replici a utilizatorului. Realizarea acestui calcul presupune existența unui model al dialogului cu ajutorul căruia să poată fi determinată evoluția sa, precum și disponibilitatea și prelucrarea unor date pe baza cărora tipuri generice de modele ale dialogului să poată fi instanțiate pentru sisteme de dialog având domenii de aplicație bine definite.

În consecință, lucrarea prezintă succesiv aspecte legate de modelarea dialogului, analiza datelor și controlul dialogului, exemplificate cu rezultate ale activității personale.

Cuprinsul

Cuprinsul	1
Lista figurilor	3
Lista tabelelor	4
1 Introducere	5
2 Modelarea dialogului	8
2.1 Principii ale modelării dialogului	8
2.1.1 Principii ale limbajului natural	9
2.1.2 Modelarea contextului	10
2.1.3 Perspective ale modelării	10
2.2 Variante de reprezentare	11
2.2.1 Grafuri orientate	11
2.2.2 Limbaje de descriere	12
3 Analiza datelor	15
3.1 Analiza conversației	16
3.1.1 Organizarea generală a dialogului	17
3.1.2 Structura locală a dialogului	19
3.1.3 Localizarea informațiilor utile în dialog	22
3.1.4 Concluzii ale analizei conversaționale	23
3.2 Analiza discursului	24

4	Controlul dialogului	27
4.1	Considerații teoretice	27
4.1.1	Istoria dialogului	29
4.2	Variante de proiectare	29
4.2.1	Modelare prin graf	30
4.2.2	Modelare în limbaj de descriere	31
5	Concluzii și continuări	34
	Referințe	36
A	Gramatica DDL	37
B	Un subdialog în DDL	40

Lista figurilor

1.1	Structura tipică a sistemelor de dialog	6
2.1	Modelarea dialogului din punct de vedere contextual	11
2.2	Exemplu de reprezentare a modelului de dialog prin graf orientat cu două tipuri de noduri	13
3.1	Proiectarea iterativă a modelului de interacțiune .	16
3.2	Organizarea generală a dialogului	17
4.1	Secvența de identificare a utilizatorului - reprezentare prin graf	30
4.2	Structura tipului de date FRAME	33

Lista tabelelor

3.1	Exemplificarea elementelor dialogului, la nivel general	18
3.2	Exemplificarea tipurilor de secvențe ale dialogului, la nivel local	20
3.3	Localizarea informațiilor în dialog	23
3.4	Ierarhia actelor de vorbire identificate	25

Capitolul 1

Introducere

Ultimii ani au adus în domeniul calculatoarelor o adevărată revoluție atât în plan comercial, cât și tehnic, astfel încât actualmente nu mai surprinde prezența lor în cele mai diverse locuri. Această situație nu ar fi fost posibilă, pe lângă avansul tehnologic și scăderea costurilor, fără interfețe utilizator accesibile și non-specialiștilor. În acest domeniu, deși supremația o dețin interfețele grafice, o creștere rapidă a popularității cunosc în ultimul timp interfețele bazate pe vorbire. Cu toate acestea, există încă numeroase constrângeri care împiedică dezvoltarea lor, printre acestea numărându-se costurile ridicate de proiectare și implementare, precum și lipsa experienței [1].

Unele dintre cele mai răspândite interfețe bazate pe vorbire se regăsesc în sistemele de dialog, reprezentate în cele mai multe cazuri de sistemele automate de furnizare a informațiilor, care în unele cazuri posedă chiar interfețe multimodale. Deși sistemele de dialog cunosc o răspândire tot mai largă, nu există încă o metodă științifică de abordare a proiectării lor, unanim acceptată de către cercetători, în ciuda apariției în domeniul comercial a unor astfel de sisteme și a diverselor proiecte de cercetare din acest domeniu. De asemenea, nu este foarte bine conturată nici o structură generală a sistemelor de dialog.

În [2], se încearcă definirea unei structuri tipice a sistemelor de dialog, prezentată în figura 1.1.

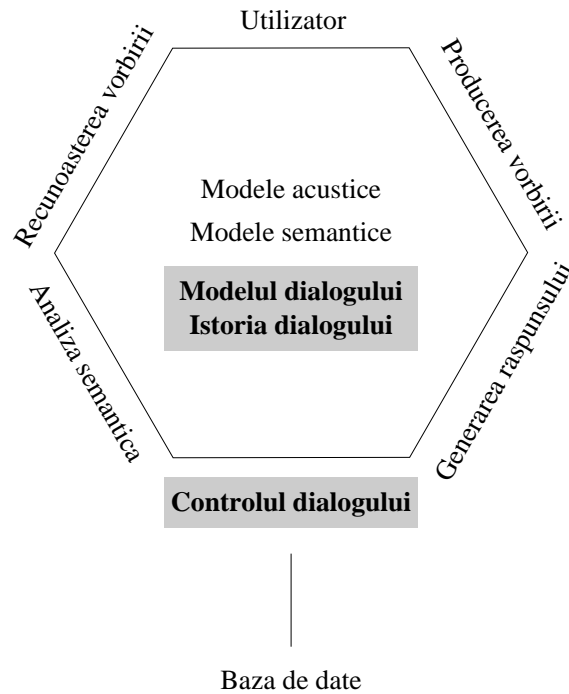


Figura 1.1: Structura tipică a sistemelor de dialog

Una din principalele probleme în proiectarea sistemelor de dialog o constituie controlul dialogului, problemă cauzată în mare parte de dificultatea definirii unui model al dialogului (model de interacțiune). Cauzele principale ale acestei probleme sunt complexitatea dialogului între oameni, particularitățile care apar în cazul dialogului om-calculator, dificultatea analizei acestuia și faptul că orice teorie a dialogului trebuie să fie interdisciplinară, necesitând cunoștințe de lingvistică, psihologie, sociologie [3]. Aceste motive conferă o importanță aparte părții de control al dialogului și modelării lui, context în care analiza datelor colectate din dialoguri între oameni sau pe

parcursul testării și utilizării sistemelor de dialog poate fi utilă pentru construirea unor modele de interacțiune care să permită implementarea unui dialog om-calculator cât mai performant.

Capitolul 2

Modelarea dialogului

2.1 Principii ale modelării dialogului

Dacă în caracterizarea sistemelor de dialog, în ciuda diverselor curente de opinie, se poate identifica o structură generală comună, în ceea ce privește modelarea dialogului părerile sunt mult mai divizate. Se disting două direcții principale, diferențiate de scopul final al modelului de dialog [4]:

1. Dezvoltarea unei teorii a dialogului.

Principalele obiective ale unei astfel de teorii sunt:

- identificarea *formulărilor* care caracterizează dialogul;
- ce presupuneri trebuie făcute legat de cunoștințele utilizatorului și de contextul dialogului pentru a putea aprecia un dialog ca fiind *cooperant*;
- ce *extensii* ale dialogului pot fi introduse pentru a îmbunătăți comportarea sistemului din punct de vedere al naturalității desfășurării dialogului.

2. Dezvoltarea unor tehnici pentru participarea calculatorului într-un dialog.

Adesea, dialogul om-calculator diferă semnificativ de un dialog interuman, care include fenomene complexe (clarificări, confirmări, referiri anterioare, ezitări ș.a.) De aici rezultă o direcție majoră a cercetării spre realizarea unui dialog cât mai natural.

Din punct de vedere al structurării modelului de dialog, unii autori [5] indică o structură pe nivele a modelului: tranzacții, schimburi, intervenții și acțiuni ale dialogului. Această structură se bazează pe trei principii: o compunere ierarhică, o compunere funcțională și recursivitate, ea permițând atingerea unor scopuri fundamentale ale modelului de interacțiune: independența față de limbă, față de domeniu (pentru a păstra generalitatea modelului) și posibilitatea de a identifica diferitele părți ale dialogului.

2.1.1 Principii ale limbajului natural

Deoarece unul dintre idealurile sistemelor de dialog este susținerea unui dialog cât mai natural cu un utilizator (idealul ”suprem” fiind trecerea testului **Turing** [3]), modelarea dialogului trebuie să se ghideze și după principii ale analizei limbajului natural [6]. Printre aceste principii cele mai importante ar fi:

- dezambiguizarea textului produs de utilizator;
- relaxarea constrângerilor;
- întărirea constrângerilor;
- confirmări;
- completări;
- descrierea acțiunilor invizibile semantic;

- detectarea confuziei utilizatorului și corectarea erorii.

O analiză detaliată va fi făcută în paragraful 3.1.

2.1.2 Modelarea contextului

Urmărirea unor aspecte definitorii ale dialogului care să permită construirea unui model de interacțiune adecvat poate creea impresia unor elemente heterogene (vezi capitolul 3). O abordare dintr-o perspectivă mai generală conduce la o analiză mai omogenă a particularităților dialogului. Această perspectivă o reprezintă modelarea contextului [5]. **Contextul** se constituie din patru axe fundamentale: *contextul factual*, reprezentând ansamblul de acțiuni din cadrul unei conversații; *contextul situațional*, care definește un aspect static, general, aparținând domeniului social; *contextul cognitiv*, un context dinamic care surprinde intențiile interlocutorilor; *contextul interacțional* cuprinzând secvențele de formulări din cursul conversației. Din aceste patru axe, trei reprezintă interes din punct de vedere computațional: contextul factual, interacțional și cognitiv.

2.1.3 Perspective ale modelării

Prin prisma celor enunțate anterior, modelarea dialogului poate fi abordată astfel:

- *Lingvistic*: din perspectiva contextului interacțional, dialogul se desfășoară în faze (numite **negocieri**), faze care pot fi caracterizate prin: (1) Deschiderea negocierii → Reacție, sau (2) Deschiderea negocierii → Reacție → Evaluarea negocierii.
- *Computațional*: se încearcă separarea, cât mai mult posibil, a modelului de interacțiune de controlul dialogului;

pentru aceasta, modelul teoretic al dialogului trebuie să fie independent de aplicație, dar fără a fi independent de tipul aplicației.

- *Ierarhic și funcțional*: dialogul este văzut ca fiind constituit din cele patru elemente prezentate în paragraful 2.1: tranzacții, schimburi, intervenții și acțiuni ale dialogului. Regulile de compunere a celor patru elemente în structura dialogului sunt prezentate în figura 2.1.

$T \rightarrow S \mid SS$	T = tranzacție,
$S \rightarrow I \mid IS \mid SS$	S = schimb,
$I \rightarrow A \mid AI$	I = intervenție,
	A = acțiune a dialogului

Figura 2.1: Modelarea dialogului din punct de vedere contextual

2.2 Variante de reprezentare

În construirea modelului de interacțiune, o problemă esențială este cea a modalității de reprezentare a lui. În primele experimente am utilizat o reprezentare prin graf orientat, iar în prezent se urmărește introducerea unei reprezentări prin limbaj de descriere.

2.2.1 Grafuri orientate

Această modalitate de reprezentare, având ca bază teoretică automatele finite, are ca principal avantaj faptul că oferă o percepție facilă asupra dialogului nespecialiștilor în calculatoare. Un alt avantaj îl constituie posibilitatea modificării ”din mers”, pentru inserarea și ștergerea unor stări fiind necesar doar un

editor specializat (primele modele au fost realizate folosind editorul daVinci [7], vezi [8]).

Graful folosit are două tipuri de noduri: sistem și utilizator. Nodurile **sistem** corespund stărilor interne ale sistemului de dialog, reprezentând un posibil punct de decizie, funcție de formularea utilizatorului. Nodurile **utilizator** reprezintă ieșirea produsă de sistem (aceasta putând fi un răspuns la o întrebare a utilizatorului, o întrebare de clarificare, o cerere de reformulare sau o întrebare adresată utilizatorului). Deși flexibilă, soluția bazată pe grafuri nu oferă posibilitatea unei reprezentări eficiente a subdialogurilor și nici reprezentarea unor informații suplimentare referitoare la starea dialogului. În figura 2.2 este prezentată o porțiune a modelului.

Pe parcursul experimentelor, modelul de interacțiune folosit a fost modificat iterativ, obținând astfel o reprezentare mai bună a dialogului. Această dezvoltare incrementală a fost facilitată de ușurința cu care o reprezentare sub formă de graf se poate modifica și adapta unor noi cerințe.

2.2.2 Limbaje de descriere

Deși simplă și intuitivă, reprezentarea prin grafuri orientate are totuși dezavantajul major că nu permite tratarea subdialogurilor, precum și alte dezavantaje legate de limitarea informațiilor și prelucrărilor ce se pot asocia nodurilor.

Eliminarea acestor dezavantaje se poate face prin utilizarea unor limbaje de descriere a dialogului. Dintre limbajele de descriere existente, cel mai reprezentativ pare HDDL din sistemul SpeechMania ([3], [9]). Acesta are un caracter declarativ și este organizat pe secțiuni, cele mai importante fiind cele de reguli și de acțiuni. Caracterul declarativ permite dialoguri cu inițiativă mixtă (sistem și utilizator), iar facilitățile importante oferite sunt: tratarea

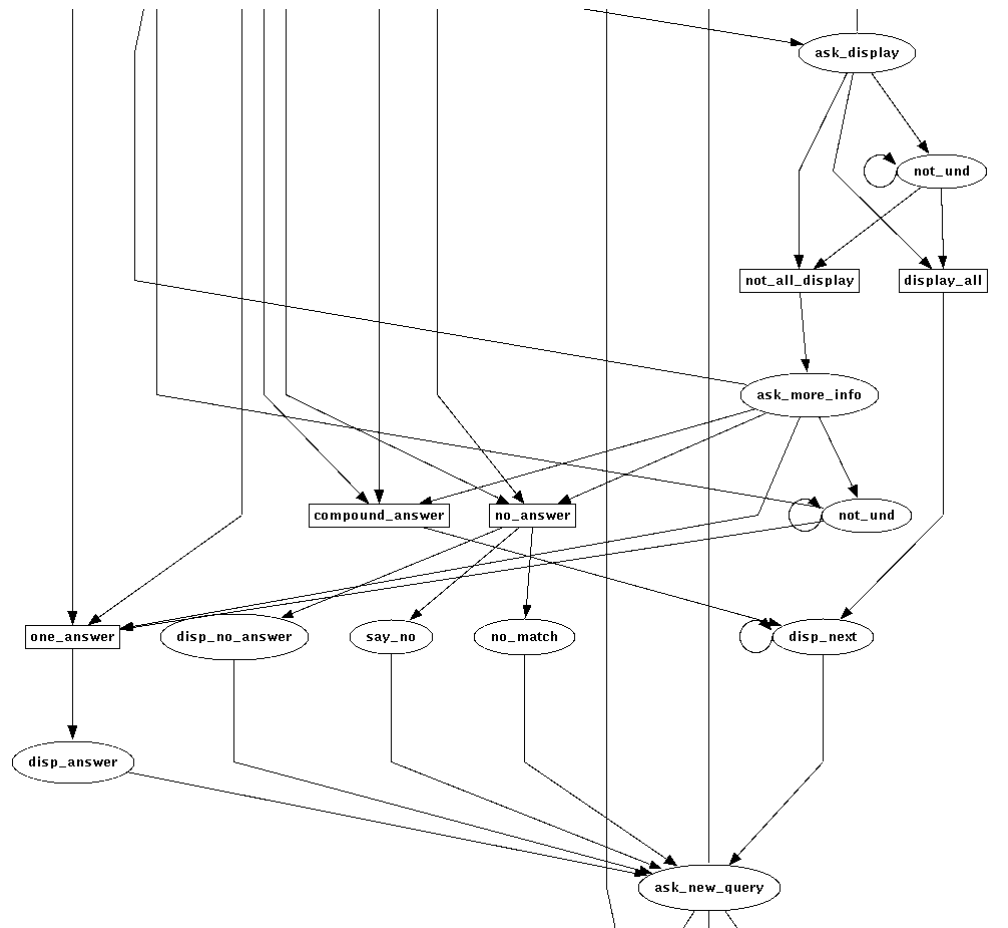


Figura 2.2: Exemplu de reprezentare a modelului de dialog prin graf orientat cu două tipuri de noduri

evenimentelor; posibilitatea modularizării (clase, obiecte grupate în biblioteci); interceptarea unor elemente specifice vorbirii naturale. O caracteristică esențială este modularitatea, modulele reprezentând *subdialoguri*.

Ca o alternativă la reprezentarea prin graf descrisă anterior, este în curs de realizare o reprezentare printr-un limbaj de descriere a dialogului (DDL = *Dialog Description Language*), bazat pe o gramatică independentă de context. Deși un asemenea limbaj necesită o oarecare obișnuință cu programarea,

el oferă avantajul unei flexibilități sporite prin includerea subdialogurilor și creșterea varietății informațiilor care pot fi procesate și a acțiunilor ce pot fi specificate, precum și posibilitatea tratării integrate a problemelor modelării și controlului dialogului.

Alegerea facilităților oferite de un limbaj de descriere a dialogului este condiționată de modul în care modulul de control al dialogului primește informațiile despre acțiunile utilizatorului. Deoarece pentru cercetările în curs a fost aleasă reprezentarea sub formă de cadre [10], principalele facilități specifice ale limbajului se regăsesc în funcțiile de procesare a cadrelor.

Limbajul DDL, propus în cadrul cercetărilor efectuate asupra modelării dialogului [11], este un limbaj procedural, fluxul interpretării unui fișier de descriere DDL fiind echivalent cu parcurgerea automatului finit echivalent dialogului. Un avantaj major al acestui mod de reprezentare este posibilitatea declarării subdialogurilor, structurând astfel dialogul în funcție de context. Gramatica limbajului este prezentată detaliat în anexa A, iar o scurtă descriere a facilităților oferite - în paragraful 4.2.2.

Reprezentarea prin limbaj nu este total disjunctă de cea prin graf, limbajul proiectat fiind structurat pe blocuri echivalente stărilor unui automat finit care reprezintă dialogul.

Capitolul 3

Analiza datelor

Evoluția unui sistem de dialog respectă ciclul de viață tipic al produselor software. În acest context, componenta care suferă cele mai multe modificări este modulul de control al dialogului, deoarece modelul dialogului pe care se bazează poate să sufere schimbări semnificative față de prima variantă folosită, de obicei construită *off-line* [2]. Analiza datelor experimentale poate însă avea o contribuție esențială la construcția rapidă a unui model al dialogului adecvat și reducerea numărului de iterații din ciclul de dezvoltare al sistemului, modelul de interacțiune ajungând mai repede la o formă care să permită angajarea calculatorului într-un dialog cât mai confortabil și mai eficient.

Privind proiectarea sistemelor de dialog din punctul de vedere al ingineriei software, analiza datelor ocupă o poziție importantă în dezvoltarea iterativă, după cum se arată în figura 3.1. O analiză corectă a datelor colectate, fie dintr-un sistem simulat, fie dintr-un sistem real, poate aduce îmbunătățiri considerabile modelului de interacțiune.

Datele colectate în urma experimentelor pot fi analizate din două perspective: analiza conversației și analiza discursului [5].

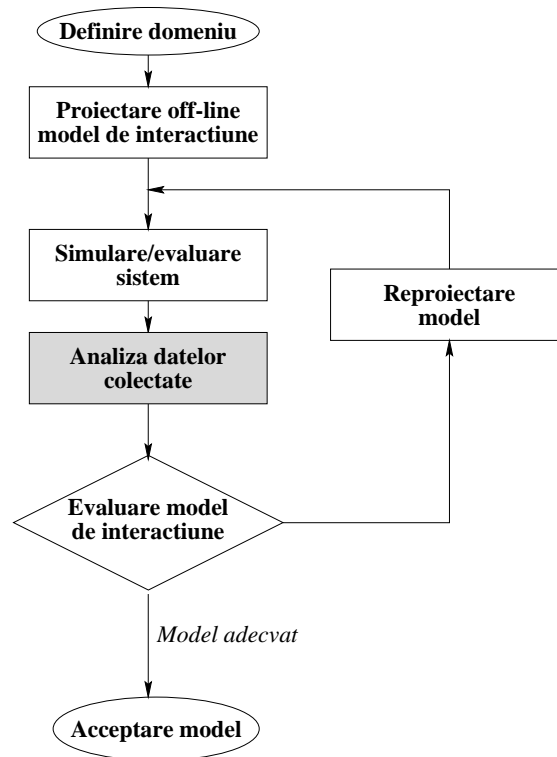


Figura 3.1: Proiectarea iterativă a modelului de interacțiune

3.1 Analiza conversației

O modelare *off-line* a dialogului bazată doar pe intuiție nu poate fi completă și nici conformă cu realitatea. De aici rezultă necesitatea studierii unui corpus de dialoguri cât mai cuprinzător. Acest studiu poate fi util și în obținerea unor clasificări a dialogurilor, identificarea unor caracteristici fundamentale și chiar comparația între diferite limbi.

Prin *analiza conversației* se urmăresc fenomenele care pot fi observate în cadrul dialogului pentru a obține o descriere a sa din punct de vedere al succesiunii evenimentelor care îl compun. Această analiză este structurată pe mai multe nivele:

1. organizarea generală a dialogului;

2. structura locală a dialogului;
3. localizarea informațiilor utile în dialog.

În continuare va fi prezentată o analiză detaliată a dialogurilor colectate în cadrul unui proiect de cercetare în domeniul sistemelor automate de dialog [11], dialoguri specifice unui sistem automat de furnizare a informațiilor despre orar [10].

3.1.1 Organizarea generală a dialogului

Din punct de vedere al organizării generale, dialogul poate fi privit ca un automat finit, cu stări reprezentând elementele structurale ale unui dialog, după cum se arată în figura 3.2. Fluxul dialogului este dat de regula: $(D(F|UF)(R)^+I)^+$.

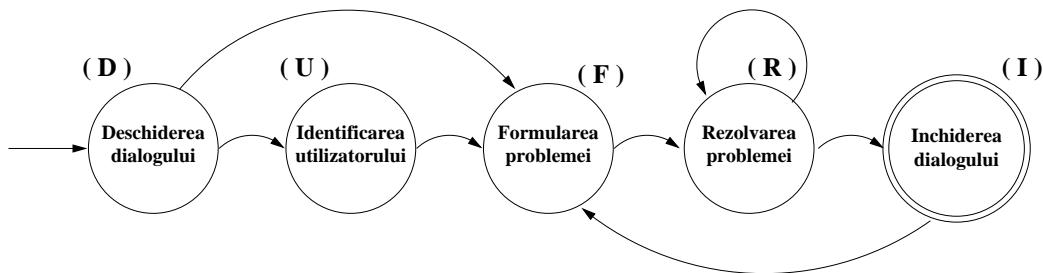


Figura 3.2: Organizarea generală a dialogului

În tabelul 3.1 sunt exemplificate elementele care compun dialogul la nivel general. Menționăm că din perspectiva analizei generale a dialogului, nu interesează unele aspecte, cum ar fi subdialogurile declanșate de erori de recunoaștere, aspecte care vor fi analizate în paragraful următor.

- **Deschiderea dialogului**

Această parte a dialogului diferă în situația unui dialog om-calculator față de un dialog uman. Chiar dacă sistemul deschide dialogul printr-un mesaj de salut, în proporție

Elementul dialogului	Secvența de dialog
Deschidere dialog	S: Bună ziua, ați mai utilizat acest sistem ?
	U: (ăă) Da, am mai utilizat acest sistem.
Identificare utilizator	S: (...) precizați anul, grupa, subgrupa (...)
	U: Sunt student în anul 4, grupa 4 1.
Formulare problemă	S: Vă rog formulați întrebarea dumneavoastră.
	U: Unde am cursul de R.L.C. și când ?
Rezolvare problemă	S: La anul 4, cursul de R.L.C. se ține joi de la 11 la 14 în sala A109.
Închidere dialog	S: Mai doriți alte informații ?
	U: Nu.
	S: Vă mulțumesc că ați folosit acest sistem. La revedere.

Tabelul 3.1: Exemplificarea elementelor dialogului, la nivel general
(U=utilizator, S=sistem)

foarte mare utilizatorii trec direct la formularea problemei, mai ales dacă dialogul este controlat de sistem.

- **Identificarea utilizatorului**

În această secvență, dialogul este în continuare controlat de sistem. Partea de identificare a utilizatorului este specifică sistemului de furnizare a informațiilor despre orar.

- **Formularea problemei**

Utilizatorul este cel care dirijează dialogul în această fază, sistemul așteptând întrebarea sa, ulterior verificând suficiența informațiilor furnizate.

- **Rezolvarea problemei**

În caz că sistemul poate genera un răspuns pe baza informațiilor furnizate, are loc transmiterea acestuia către utilizator; în caz contrar se inițiază subdialoguri pentru clarificări sau introduceri de informații suplimentare.

- **Închiderea dialogului**

De obicei, se oferă utilizatorului opțiunea de a relua dialogul, pentru solicitarea altor informații. În sistemele care oferă și anumite servicii (în general rezervări de bilete la avioane, trenuri), închiderea interactivă a dialogului are loc prin repetarea acțiunii solicitate, așteptându-se confirmarea utilizatorului [12].

3.1.2 Structura locală a dialogului

Analiza structurii locale a dialogului oferă o perspectivă la nivelul succesiunii de formulări a celor doi interlocutori, succesiune care constituie o fază a dialogului [5], [13].

Sucesiunea formulărilor poate fi caracterizată din mai multe puncte de vedere: perechi adiacente, schimbări de vorbitor, întreruperi, rupturi-corectări, confirmări, precizări. Tabelul 3.2 prezintă exemple de formulări, urmărind structura locală a dialogului.

- **Perechi adiacente**

Marea majoritate a intervențiilor celor doi interlocutori pot fi văzute ca perechi adiacente de formulări (mesaj de salut - răspuns la salut, întrebare - răspuns, întrebare - negare etc.). Caracteristica principală a perechilor adiacente este dependența logică și secvențială a celei de-a doua formulări de prima formulare (fiecare formulare aparținând unuia din cei doi interlocutori), determinând o configurație de tip cauză - efect.

Deși această perspectivă pare simplă și eficientă, nu toate secvențele de dialog pot fi structurate astfel (în unele cazuri, din punct de vedere logic, se impune înlănțuirea unei a treia formulări, în alte situații o formulare poate fi singulară.)

Tipul secvenței	Secvența de dialog
Perechi adiacente	U: (ăă) Cu cine am laboratorul de I.P. ?
	S: Ați spus laboratorul de I.P. ?
Schimbări de vorbitor (obstrucționare)	U: Poți să-mi spui, (ăă), când am laborator de I.A., te rog ?
Înteruperi (1)	S: Dacă sunteți student, precizați ■ anul (...)
	U: ▷ Aș vrea să știu când (...)
Înteruperi (2)	S: Dacă sunteți student, precizați ■ anul (...)
	U: ▷ Sunt în anul (...)
Rupturi-corectări (1)	U: (...) grupa 2 1 (ăă) grupa 2 subgrupa 1.
Rupturi-corectări (2)	U: Când pot recupera laboratorul de P.C. ?
	S: Îmi pare rău, informația solicitată nu există în orar.
Rupturi-corectări (3)	U: Unde se ține cursul de S.M. ?
	S: Nu există ore conform criteriilor specificate.
Confirmări	U: Când am curs de I.P. ?
	S: La anul 4, cursul de I.P. se ține (...)
Precizări	U: La anul 1, când este laboratorul de P.C. ?
	S: (...) doriți să precizați criterii suplimentare (...)
	U: Da.
	S: Precizați grupa și subgrupa.

Tabelul 3.2: Exemplificarea tipurilor de secvențe ale dialogului, la nivel local (U=utilizator, S=sistem)

• Schimbări de vorbitor

În cadrul unui dialog între doi interlocutori umani, au loc schimbări frecvente de vorbitor. Pentru a realiza schimbarea efectivă a persoanei care vorbește, în cadrul vorbirii naturale apar elemente specifice care declanșează procedura de schimbare (în general, aceste elemente se manifestă prin intonație sau unele expresii). Într-un dialog om-calculator aceste elemente sunt dificil de surprins, schimbul de vorbitor având loc pe principii acustice (pe partea sistemului) și pe principii logice și acustice (pe

partea utilizatorului).

Utilizatorii umani încearcă, în cadrul dialogului, ”obstrucționarea” celuilalt vorbitor de a lua cuvântul. Aceste elemente (de obicei ezitări, repetări sau introducerea unor cuvinte suplimentare, de politețe) pot fi detectate în cadrul dialogului om-calculator.

- **Întreruperi**

Atât în dialogurile inter-umane, cât și în cele om-calculator, unul din interlocutori (utilizatorul) întrerupe interlocutorul care vorbește (sistemul). Aceste întreruperi pot avea două cauze: (1) neatenția utilizatorului la formularea sistemului (primul caz din tabelul 3.2), sau (2) o formulare a sistemului pe care utilizatorul o anticipează sau o cunoaște deja.

- **Rupturi-corectări**

Rupturile, denumite și *eșecuri ale dialogului*, sunt situații care pot fi caracterizate ca ”anormale” în raport cu un model de desfășurare corectă a dialogului. De obicei, rupturile sunt urmate de corectări, prin care se încearcă restabilirea desfășurării normale a dialogului. Rupturile urmate de corectare pot fi: (1) auto-corectări (un enunț incoerent urmat imediat de unul coerent), (2) determinate de domeniu sau de scopul dialogului (rupturi pe care le semnaleză sistemul) și (3) eșecuri la nivel de recunoaștere (în unele cazuri, pot fi detectate de către sistem).

- **Confirmări**

În cursul desfășurării dialogului, apare necesară o sincronizare a celor doi interlocutori, pentru a verifica păstrarea consistenței subiectului. Într-un dialog om-calculator, aceasta poate avea loc prin repetarea de către sistem a informației furnizate de utilizator.

- **Precizări**

Precizările sunt solicitate de sistem, în cazul în care utilizatorul a furnizat insuficiente informații sau formularea utilizatorului este ambiguă (conține contradicții sau duce la prea multe răspunsuri în urma interogării bazei de date).

3.1.3 Localizarea informațiilor utile în dialog

În acest paragraf va fi analizată maniera în care informația este vehiculată în cadrul dialogului. Această analiză are la bază observația că subiectul și structurarea tematică a unui dialog influențează modul în care sunt structurate formulările. Din această perspectivă, se pot identifica trei elemente fundamentale: referințe, elipse și anafore. Transcrieri ale dialogurilor care conțin astfel de elemente se găsesc în tabelul 3.3 (în paranteze se indică informația vehiculată). Analiza acestor elemente are o importanță deosebită atât în faza de modelare, cât și în cea de proiectare a controlului dialogului (vezi paragraful 4.1.1).

- **Referințe**

Prin referințe interlocutorul indică în mod implicit entități sau obiecte anterioare în cadrul dialogului. Existența referințelor induce un grad de ambiguitate în dialog.

- **Anafore**

Anafora este foarte similară cu referința, fiind identificată prin folosirea unor pronume. De multe ori, o referință este dezambiguizată prin folosirea anaforei, astfel că unii cercetători tratează în mod unitar anaforele și referințele (vezi [13]) sau descriu anafora ca un caz particular de referință (vezi [5]).

- **Elipse**

Elipsele descriu entități subînțelese în contextul momentan al dialogului. Într-un dialog, pot fi identificate trei tipuri de elipse:

- *Elipse conversaționale* (legate de domeniul dialogului);
- *Elipse lingvistice* (descriu reutilizarea informației, prin păstrarea contextului la care se referă o formulare);
- *Elipse pragmatice* (utilizarea unei informații implicite).

Categoria localizării	Secvența de dialog
Referință	U: Sunt în anul 2, grupa 3.
	S: Vă rog formulați întrebarea dumneavoastră.
	U: Unde am [<i>grupa 3</i>]laborator de S.D.A. ?
Anaforă	U: Cu cine se ține cursul de S.I.E. ?
	S: La anul 4, cursul de S.I.E. se ține cu profesorul Strugaru.
	U: Ce ore mai am tot cu el [<i>cu prof. Strugaru</i>]?
Elipse conversaționale	U: Când am cursul de I.A. [<i>Inteligență Artificială</i>]?
Elipse lingvistice	U: Ce ore am marți ?
	S: [<i>Marți</i>] De la 8 la 11, curs de I.P.
Elipse pragmatice	U: Ce laborator se ține în [<i>sala</i>] B 529 luna?

Tabelul 3.3: Localizarea informațiilor în dialog
(U=utilizator, S=sistem)

3.1.4 Concluzii ale analizei conversaționale

În urma analizei dialogului din punct de vedere conversațional, se poate concluziona că, pentru a se comporta adecvat în orice situație, un sistem de dialog trebuie să identifice

toate elementele enunțate anterior. Practic, în acest stadiu al dezvoltării sistemelor de dialog, acest lucru nu este posibil, putându-se încerca rezolvarea doar a unor situații [5]:

- respectarea structurii globale a dialogului;
- rezolvarea unor eșecuri legate de domeniu;
- rezolvarea unor eșecuri legate de recunoaștere;
- furnizarea unor răspunsuri cuprinzătoare;
- interpretarea referințelor și a elipselor;
- generarea elipselor și a referințelor.

3.2 Analiza discursului

Deși analiza conversației poate oferi informații importante asupra dialogului, unele fenomene nu pot fi surprinse în acest mod: nici un element nu poate indica ce anume motivează cei doi interlocutori; nu se poate urmări progresul dialogului (ce elemente fac ca un dialog coerent să poată fi definit și ca eficient). Din acest motiv, dialogul trebuie analizat nu doar în cursul și după desfășurarea sa, ci și înainte de momentul începerii.

Analiza discursului se bazează pe elemente ale teoriei comunicării și are ca unități fundamentale actele de vorbire (speech acts), entități minimale ale unui dialog care reprezintă acțiuni întreprinse de interlocutor prin intermediul formulărilor sale. Aceasta presupune o analiză interdisciplinară, implicând elemente de lingvistică, psihologie, sociologie, lingvistică computațională, științe cognitive, regăsindu-se ca disciplină independentă în cadrul pragmaticii, unde are la bază studiul fenomenelor de "deixis", ca legătură între limbaj și context [13].

Scopul analizei discursului este de a găsi o reprezentare a stării mentale, intenționale, a utilizatorului, prin identificarea

cunoștințelor și a inferențelor folosite de utilizator în cadrul dialogului. O asemenea reprezentare poate oferi informații despre ce așteaptă utilizatorul de la sistem.

În analiza discursului există două principii fundamentale [5]: *principiul cooperării*, enunțat de H.P. Grice, prin care se consideră că fiecare participant la dialog oferă cât mai multe informații interlocutorului pentru a realiza un dialog cooperant, și *principiul pertinentei*, introdus de D. Wilson și D. Sperber, contrar celui anterior, și conform căruia un dialog este cu atât mai pertinent cu cât obține un grad mai mare de interes din partea interlocutorului prin transmiterea unei cantități reduse de informații.

	Nivel	Categorie	Tip
Acte de vorbire	sistem	cereri	de informații suplimentare
			de confirmare
			de identificare
		mesaj	de deschidere
			de introducere
			de închidere
	utilizator	răspuns	negativ
			pozitiv
		confirmare	pozitivă
			negativă
		identificare	precizare
			refuz
cerere	întrebare		

Tabelul 3.4: Ierarhia actelor de vorbire identificate

Pentru a putea analiza dialogul din perspectiva discursului, trebuie deci identificate actele de vorbire care intervin în cadrul lui. În general, actele de vorbire sunt în corespondență cu stările automatului finit care reprezintă dialogul, fiind dependente de domeniul sistemului de dialog. Pentru exemplificare, în tabelul 3.4 sunt prezentate ierarhic actele de vorbire care pot apare

în cadrul dialogului în cazul sistemului pilot de furnizare a informațiilor despre orar.

Recunoașterea de către sistem a actelor de vorbire în timpul desfășurării dialogului este o sarcină dificilă, de rezolvarea căreia este responsabil controlul dialogului, pe baza informațiilor furnizate de modulele de recunoaștere automată a vorbirii și analiză semantică.

Nu în ultimul rând, ca și în cazul analizei conversației, analiza discursului este importantă și pentru a obține un model de dialog care să permită sistemului o comportare ”umană” și din punct de vedere al ieșirii, prin folosirea a cât mai mult posibil a unor elemente din dialogurile inter-umane.

Capitolul 4

Controlul dialogului

4.1 Considerații teoretice

Modulul de control al dialogului este partea centrală a unui sistem de dialog, de el depinzând modul în care sistemul poate susține un dialog cât mai natural. Chiar dacă și asupra proiectării părții de control a sistemelor de dialog există păreri împărțite, în general se pleacă de la ideea că dialogul poate fi văzut ca un automat finit determinist, care se află în orice moment într-o stare bine definită, având ca intrări acțiunile utilizatorului [14].

Controlul dialogului se află în strânsă legătură cu modelul de interacțiune. O implementare a părții de control independentă de domeniul de aplicație, și care folosește modelul de interacțiune ca suport, oferă o flexibilitate și portabilitate sporită sistemului de dialog, precum și independență față de domeniul de aplicație.

Principala funcție a modulului de control al dialogului este determinarea la un moment dat, pe baza caracteristicilor domeniului de aplicație, a contextului conversației cu un utilizator și a ultimei lui formulări, a următoarei acțiuni a sistemului, și anticiparea, eventual chiar determinarea, conținutului următoarei formulări utilizator. Pe această bază,

modulul de control al dialogului efectuează următoarele acțiuni principale: [3]

- determină dacă s-au strâns suficiente informații de la utilizator pentru a putea realiza comunicarea cu o aplicație externă (în cazul sistemelor de informare, o bază de date);
- realizează comunicarea cu aplicația externă;
- realizează comunicarea (transmiterea informației) către utilizator.

Controlul dialogului este îngreunat de o comportare "naturală" a utilizatorului, sistemul trebuind să decidă ce acțiuni să întreprindă când formulările utilizatorului sunt ambigue sau nu conțin suficiente informații.

Proiectarea modulului de control al dialogului este dependentă, pe lângă modelul de interacțiune, și de analizorul semantic (vezi figura 3.1) (pentru prezentarea analizei semantice a se vedea [10]). Chiar dacă analizorul semantic oferă o interfață convenabilă, rămân două sarcini importante care trebuie rezolvate la nivelul controlului dialogului: identificarea acțiunilor utilizatorului și generarea formulărilor către utilizator (gestionarea ieșirii) [6].

- **Identificarea acțiunilor utilizatorului** are ca scop extragerea informației utile din formulările utilizatorului și luarea unor decizii privind suficiența sau insuficiența acestor informații. Un model de interacțiune adecvat oferă controlului dialogului posibilitatea identificării rapide și eficiente a informației. Pentru extragerea informației ținând cont de caracteristicile limbajului natural (descrise în capitolul 3), o soluție o reprezintă gestionarea unei *istorii a dialogului* (vezi paragraful următor).

- **Gestionarea ieșirii** poate îmbunătăți în mod hotărâtor ”naturațea” dialogului om-calculator. Pentru aceasta, formulările sistemului trebuie să fie *informative* (să ofere cât mai multe informații posibil), *eficiente* (să conducă la o rezolvare cât mai rapidă a problemei) și *computațional necostisitoare* (să solicite cât mai puține resurse).

4.1.1 Istoria dialogului

Istoria dialogului reprezintă o înlanțuire a formulărilor utilizatorului și ale sistemului, prin care se permite în orice moment al dialogului accesarea oricăror formulări anterioare. Scopul menținerii unei istorii a dialogului este de a rezolva referințele și elipsele (definite în paragraful 3.1.3).

Din punct de vedere practic, istoria dialogului se implementează cel mai ușor sub forma unei stive conținând toate formulările anterioare. Această soluție, deși simplu de implementat, este ineficientă din punct de vedere al vitezei, datorită necesității prelucrării semantice a fiecărei formulări, în vederea localizării informației. O soluție eficientă este păstrarea unei stive cu reprezentările semantice, o asemenea stivă fiind utilă și în momentul analizei semantice, pentru rezolvarea unor ambiguități [10].

4.2 Variante de proiectare

După cum s-a menționat în paragraful anterior, controlul dialogului este strâns legat de modelul de dialog ales. Plecând de la cele două variante principale de descriere a dialogului, managerul de dialog poate fi proiectat folosind un model reprezentat prin graf sau unul reprezentat printr-un limbaj de descriere specializat.

4.2.1 Modelare prin graf

În cazul utilizării acestei reprezentări, controlul dialogului consta în parcurgerea grafului și efectuarea a două tipuri de acțiuni: procesarea formulărilor utilizatorului, pe baza cărora se alege starea următoare, și generarea unui răspuns sau a unei întrebări către utilizator [15].

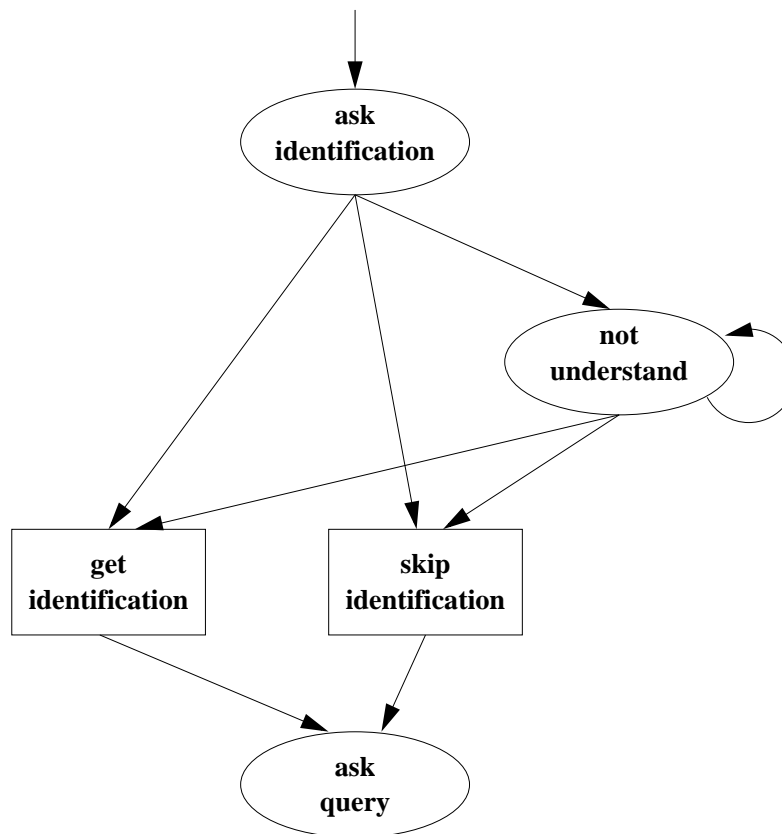


Figura 4.1: Secvența de identificare a utilizatorului - reprezentare prin graf

În figura 4.1 se prezintă un exemplu de secvență a reprezentării prin graf. Secvența corespunde cererii sistemului de identificare a utilizatorului. Din figură se observă unul din neajunsurile reprezentării prin graf: în cazul unei formulări greșite a utilizatorului, sistemul solicită o reformulare. Această

situație, deși se repetă în toate momentele în care se așteaptă formulări de la utilizator, trebuie descrisă de fiecare dată, deoarece modelarea prin graf nu permite reprezentarea subdialogurilor. Un alt dezavantaj al acestei soluții îl constituie necesitatea unui mecanism suplimentar de memorare (de obicei, accese suplimentare la istoria dialogurilor) (în exemplul din figura 4.1, informația de identificare a utilizatorului va fi folosită în interogarea bazei de date).

4.2.2 Modelare în limbaj de descriere

Această modalitate de modelare a dialogului oferă o mai mare flexibilitate și pentru controlul lui, limbajul de descriere permițând specificarea, pe lângă informațiile legate de modelarea dialogului, și a acțiunilor necesare în fiecare stare. Astfel, modulul de control al dialogului poate fi construit în jurul unui interpretor DDL.

În actualul stadiu de dezvoltare al sistemului, interpretorul DDL este folosit doar pentru a evalua gramatica și facilitățile limbajului. Această situație este justificată de faptul că asemenea limbaje există din păcate doar în număr redus, în sisteme comerciale (e.g. HDDL, folosit în sistemul SpeechMania [9]) sau nedisponibile public (e.g. DDL-Tool, care se prezintă sub forma unui mediu CAD [14]), ceea ce duce la lipsa de informații și de experiență în acest domeniu, astfel încât sunt necesare numeroase teste pentru a se putea stabili și evalua funcționalitatea sa.

Avantajele descrierii prin limbaj pledează pentru folosirea unei astfel de reprezentări în locul uneia prin graf. Astfel, principalul avantaj îl constituie posibilitatea structurării dialogului în subdialoguri, ceea ce oferă o soluționare globală a unor sarcini locale. Alt avantaj îl constituie posibilitatea reprezentării explicite a informațiilor (sub forma variabilelor),

ceea ce poate reduce la minim accesul la istoria dialogului.

În urma acestei analize, a rezultat ca necesară și fezabilă, într-o primă etapă, punerea la punct a unui limbaj procedural, care să ofere o libertate cât mai mare în proiectarea părții de control al dialogului. În urma analizei datelor colectate (vezi capitolul 3) și a evidențierii neajunsurilor reprezentării prin graf, s-a definit un limbaj DDL (*Dialog Description Language*). Acesta se află în prezent în stadiul de evaluare în contextul unor cercetări asupra sistemelor de dialog vocal om-calculator [11]. Gramatica limbajului DDL este prezentată integral în anexa A.

Facilitățile oferite de DDL cuprind:

- declararea și manipularea unor structuri de date specifice: tipul de date `FRAME` (cadru), cu structura din figura 4.2, și a variabilelor de tip numeric și șir de caractere. Tipul de date `FRAME` este impus de analizorul semantic bazat pe cadre [10], controlul dialogului preluând de la acesta informația extrasă din formularea utilizatorului sub formă de cadre;
- apelarea și procesarea subdialogurilor prin instrucțiuni specifice (`CALL nume_nod`);
- controlul tranzițiilor între stări (instrucțiunea `JUMP nume_nod`);
- controlul fluxului dialogului (instrucțiuni de salt și condiționale: `IF - ELSE` și `GOTO etichetă`);
- facilități specifice acțiunilor sistemului:
 - preluarea cadrelor de la analizorul semantic (funcția `get_frame (decl_frame)`);
 - accesarea istoriei dialogului, organizat ca listă de cadre (funcțiile `next_frame (decl_frame)` și `prev_frame (decl_frame)`);

- generarea răspunsului (instrucțiunea `print_output (lista_parametri)`);
- interogarea bazei de date (instrucțiunea `print_query (expresie_sir)`).

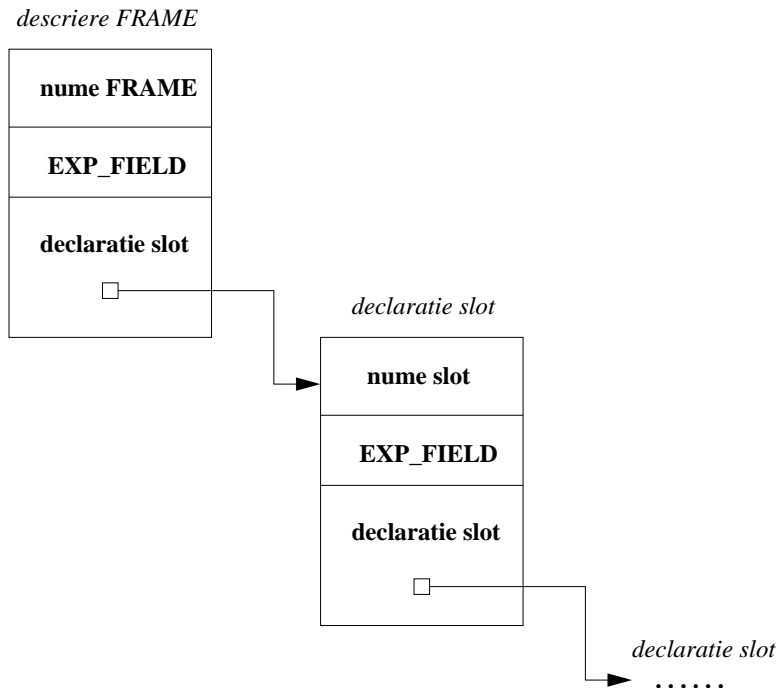


Figura 4.2: Structura tipului de date FRAME

În anexa B se prezintă un exemplu de subdialog scris în limbajul DDL, subdialog care corespunde porțiunii de graf din figura 4.1. Se observă posibilitatea utilizării de informații suplimentare în cursul dialogului, fără restricții asupra momentului folosirii lor.

Capitolul 5

Concluzii și continuări

Construirea unui sistem de dialog presupune un efort considerabil, care poate fi însă diminuat prin proiectarea corespunzătoare a elementelor sale fundamentale, context în care cea mai importantă componentă, modulul de control al dialogului, necesită o atenție deosebită. Din această perspectivă, un model de interacțiune adecvat este hotărâtor pentru performanța sistemului final, iar o modelare cât mai bună a dialogului impune o analiză corespunzătoare a datelor colectate în timpul experimentelor.

În cadrul lucrării prezentate am urmărit, pentru început, obținerea unui model de interacțiune adecvat unui sistem de dialog din categoria sistemelor automate de furnizare a informațiilor. În urma experimentelor realizate pe durata mai multor luni și a analizei datelor astfel acumulate, modelarea inițială prin graf și-a demonstrat neajunsurile, astfel încât s-a impus trecerea la modelarea printr-un limbaj de descriere a dialogului. S-a definit un limbaj adecvat (DDL) care să permită atât modelarea cât și controlul dialogului, oferind astfel o mai mare flexibilitate și în proiectarea părții de control a dialogului.

Cercetările viitoare vor urmări testarea și îmbunătățirea limbajului prin construirea unui modul de control al dialogului bazat pe acesta și evaluarea lui experimentală.

Bibliografie

- [1] S. Sutton et al. Building 10,000 Spoken Dialogue Systems. În *Proceedings ICSLP'96*, 1996.
- [2] N. O. Bernsen, H. Dybkjaer, și L. Dybkjaer. *Designing Interactive Speech Systems*. Springer-Verlag, Londra, 1998.
- [3] M. McTear. Spoken Dialogue Technology: Enabling the Conversational User Interface. *ACM Computing Surveys (în curs de apariție)*, 2000.
- [4] P. Cohen. Dialogue Modeling. În *Survey of the State of the Art in Human Language Technology*, chapter 6: Discourse and Dialogue. Cambridge University Press, Cambridge, Marea Britanie, 1998.
- [5] E. Bilange. *Dialogue personne-machine*. Hermès, Paris, 1992.
- [6] A. Abella, M.K. Brown, și B. Buntschuh. Development Principles for Dialog-Based Interfaces. În *Dialogue Processing in Spoken Language Systems*. Springer-Verlag, Londra, 1996.
- [7] *daVinci Online Documentation V2.1*. www.tzi.de/~davinci, 1998.
- [8] C. Munteanu. Mediu pentru dezvoltarea sistemelor de dialog prin metoda Vrajitorului din Oz. Lucrare de

- diplomă, Departamentul de Calculatoare, Universitatea "Politehnica" din Timișoara, iunie 1999.
- [9] H. Aust și O. Schröer. An Overview of the Philips Dialog System. În *Proceedings DARPA Broadcast News Transcription and Understanding Workshop*, Landsowne, USA, 1998.
- [10] D. Bohuș. Analiza semantică stocastică în dialogul om-calculator. Lucrare de diplomă, Departamentul de Calculatoare, Universitatea "Politehnica" din Timișoara, iunie 2000.
- [11] M. Boldea. SADVOC: Sisteme Automate de Dialog Vocal Om-Calculator. Propunere de proiect major de cercetare înaintată CNCSIS, 1999.
- [12] E. Giachin și S. McGlashan. Spoken Language Dialogue Systems. În *Corpus-Based Methods in Language and Speech Processing*. Kluwer Academic Publishers, Dordrecht, Olanda, 1997.
- [13] S. C. Levinson. *Pragmatics*. Cambridge University Press, Cambridge, Marea Britanie, 1992.
- [14] B. Lindberg et al. An Integrated Dialogue Design and Continuous Speech Recognition System Environment. În *Proceedings ICSLP'92*, Banff, Canada, 1992.
- [15] C. Munteanu și M. Boldea. MDWOZ: A Wizard of Oz Environment for Dialog Systems Development. În *Proceedings Second International Conference on Language Resources and Evaluation*, Atena, 2000.

Anexa A

Gramatica DDL

Cuvinte rezervate :

var, MANDATORY, OPTIONAL, FORBIDDEN, if, fi, else,
print_query, print_output, jump, goto, NULL, TYPE,
get_frame, prev_frame, next_frame, NODE, LABEL

Terminale (token-uri):

a...z A...Z 0...9 NL () < > [] , ; ! == != <=
>= = " // (NL=newline)

Comentarii (linia curentă): //

start_ddl	→ decl_var noduri
noduri	→ nod nod noduri
decl_var	→ <u>VAR</u> declaratii NL
declaratii	→ identif identif declaratii
identif	→ litera litera rest_identif
rest_identif	→ litera cifra litera rest_identif cifra rest_identif
litera	→ a b ... z A ... Z
cifra	→ 0 1 ... 9
nod	→ <u>NODE</u> nume_nod NL bloc_nod
nume_nod	→ identif

bloc_nod	→ statement statement bloc_nod
statement	→ <u>LABEL</u> eticheta NL instruction function
eticheta	→ identif
instruction	→ instr_if instr_pr_q instr_pr_o instr_jump instr_go instr_atrib instr_call
instr_if	→ <u>if</u> conditie NL if_statement NL <u>fi</u>
if_statement	→ bloc bloc NL <u>else</u> NL bloc
conditie	→ conditie_logica conditie , conditie conditie ; conditie !conditie
conditie_logica	→ function relatie
relatie	→ operand op_rel operand
operand	→ expresie expresie_sir constanta
op_rel	→ == != < > <= >=
instr_pr_q	→ <u>print_query</u> (expresie_sir)
instr_pr_o	→ <u>print_output</u> lista_param_pr
instr_jump	→ <u>jump</u> nume_nod
instr_go	→ <u>goto</u> eticheta
instr_call	→ <u>call</u> nume_nod
instr_atrib	→ ref_variabila = expresie
function	→ funct_null funct_type func_get_frame func_next_frame func_prev_frame
funct_null	→ <u>NULL</u> (variabila)
funct_type	→ <u>TYPE</u> (variabila)
func_get_frame	→ <u>get_frame</u> (decl_frame)
func_next_frame	→ <u>next_frame</u> (decl_frame)
func_prev_frame	→ <u>prev_frame</u> (decl_frame)
decl_frame	→ λ descr_frame decl_frame

descr_frame	→ (<nume_frame> [EXP_FIELD] slot_decl)
nume_frame	→ identif
slot_decl	→ slot_descr slot_descr slot_decl
slot_descr	→ (slot_name [EXP_FIELD])
slot_name	→ identif
EXP_FIELD	→ <u>MANDATORY</u> <u>OPTIONAL</u> <u>FORBIDDEN</u>
constanta	→ semn const_num const_sir
semn	→ + - λ
const_num	→ cifra const_num
const_sir	→ " sir_caract "
sir_caract	→ litera sir_caract
ref_variabila	→ identif
variabila	→ user_var frame_var
user_var	→ identif
frame_var	→ nume_frame nume_frame.slot_acces
slot_acces	→ nume_frame slot_name slot_name.slot_index slot_acces
slot_index	→ const_num user_var
expresie	→ operand op_aritm operand
op_aritm	→ + - * /
expresie_sir	→ operand_sir operand_sir + expresie_sir
operand_sir	→ variabila const_sir

Anexa B

Un subdialog în DDL

```
VAR frame_curent, fara_identif, an, grupa, subgrupa,
    profesor

(.....)

NODE identificare
// declaratia de inceput a nodului

LABEL identificare_input

frame_curent = get_frame ((<identificare> [MANDATORY]
    (<an> [OPTIONAL]) (<grupa> [OPTIONAL]) (<subgrupa>
    [OPTIONAL])) (<identificare> [MANDATORY] (<profesor>
    [MANDATORY])) (<negativ> [MANDATORY] <nu> [MANDATORY]))
//extragere cadru curent

if NULL(frame_curent)
//cadru neidentificabil
    call not_understand
    goto identificare_input
fi

if frame_curent == <identificare>
//cadrul reprezinta o identificare
    fara_identif=0
    if ! NULL(frame_curent.identificare.profesor)
        //utilizatorul este un profesor
        profesor = frame_curent.identificare.profesor
```

```
        else
            //utilizatorul este student
            if ! NULL(frame_curent.identificare.an)
                an = frame_curent.identificare.an
            if ! NULL(frame_curent.identificare.grupa)
                grupa = frame_curent.identificare.grupa
            if ! NULL(frame_curent.identificare.subgrupa)
                subgrupa = frame_curent.identificare.subgrupa
        fi
    else
        if frame_curent == <negativ>
            //utilizatorul nu s-a identificat
            fara_identif=1
        else
            //formulare corecta, dar nepotrivita in context
            call not_understand
            goto identificare_input
        fi
    fi

    jump cerere_intrebare
    //salt la urmatorul nod

    (.....)
```