

# Klepec: slovenski programirani sogovornik

Špela Arhar, Miro Romih

Amebis, d. o. o.  
Bakovnik 3, 1241 Kamnik, Slovenija  
[spela.arhar@amebis.si](mailto:spela.arhar@amebis.si), [miro.romih@amebis.si](mailto:miro.romih@amebis.si)

## Povzetek

Program Klepec je klepetanju namenjen programirani sogovornik, ki za jezik komunikacije z uporabnikom uporablja slovenščino. Nastal je v sklopu projekta KOLOS, katerega cilj je omogočiti komunikacijo med človekom ter računalnikom v naravnem jeziku. V članku predstavlja trenutno stanje programa ter nakazuje smernice za nadaljnji razvoj, obenem pa izpostavlja nekatera problematična mesta razvoja programiranih sogovornikov (specifike klepetalniškega diskurza, potreba po antropomorfizaciji programa ipd.). Razpravo dopolnjujeva s primeri realnih komunikacijskih nizov med uporabniki ter programom Klepec.

## Klepec: a Slovene chatbot program

Klepec is a chatbot computer program that uses Slovene as the language of communication. It was created within the KOLOS project, the main goal of which was to establish natural-language communication between human users and computers. In the article we present the current state of the program, together with some guidelines for its future development. In addition, we indicate some of the main difficult areas of chatbot-programming (specifics of the chatting discourse, the need for the anthropomorphization of the program etc.). The paper is supplemented by the examples of real communication sequences between users and the Klepec program.

## 1. Uvod

Programirani sogovornik je program, ki komunicira z uporabnikom ali drugim programom v izbranem naravnem jeziku. Poznamo programirane sogovornike, namenjene predvsem krajšanju časa s klepetanjem (v angleški literaturi najdemo zanje izraze *chatbot*, *chatterbot*, *chatterbox*), ter takšne, ki imajo kakšno drugo uporabno vrednost, konverzijske sposobnosti pa izrabljajo kot pomoč pri dosegu svojega cilja. Med slednje spadajo predvsem nekateri tipi t. i. inteligentnih agentov (*intelligent agents*), programov, ki za uporabnika ali druge programe izvršujejo različne naloge, za katere so pooblaščen (iščejo ter urejajo različne informacije, prodajajo artikle, vodijo uporabnika skozi določeno opravilo ...).

Namen članka je predstaviti program Klepec, ki je prvi resni poskus izdelave programiranega sogovornika za slovenski jezik. Članek se osredotoča predvsem na izbiro slovenščine za jezik komunikacije ter iz tega izvirajoče specifike pri razvoju programa, ob strani pa pušča splošnejši prikaz programske zasnove ter delovanja (slednje v Romih et al., 2002).

## 2. Kaj je Klepec

Klepec je eden izmed jezikovnotehnoških projektov podjetja Amebis, d. o. o (<http://www.amebis.si>). Nastal je v sklopu razvoja sistema KOLOS, katerega cilj je »ustvariti okolje in orodja, s pomočjo katerih bi bila komunikacija med človekom in računalnikom enaka komunikaciji med ljudmi« (Romih et al., 2002). Trenutno je na internetu verzija 2.0, v teku pa so že priprave izboljšane verzije.<sup>1</sup>

Zaenkrat je Klepec programirani sogovornik, namenjen zabavi in klepetu, s postopnim vgrajevanjem znanj oz. podatkovnih baz pa naj bi se razvil v agenta, ki

zna uporabniku poiskati želene informacije oz. mu s svojim znanjem pomaga pri reševanju različnih nalog.

## 3. Kaj zna Klepec

Stanje, v katerem najdemo program na trenutni stopnji razvoja, na kratko predstavlja v štirih podpoglavjih.

### 3.1. Slovsko ter slovnično znanje o jeziku

Uporabniške vnose program zaenkrat razčlenjuje le na prvi stopnji. Strukturno prepoznava stavke ter besede, besede lematizira, vendar v primeru dvoumne lematizacije ne izbira med obstoječimi možnostmi. Besednovrstne kategorije so rabljene predvsem v programski kodi, kjer z njimi generaliziramo tako vzorce za prepoznavo uporabniškega vnosa kot tudi odgovore na določen tip uporabnikove iztočnice. Nova verzija programa naj bi prinašala bistveno kvalitetnejšo jezikovno analizo vnosa na osnovi posodobitve programa z izboljšanim lematizatorjem ter jezikovnim analizatorjem, ki omogoča razdvoumljanje lematizacije ter delno stavčnočlensko analizo.

Klepec ima torej vgrajeno znanje o pregibanju besed in če ga v pogovoru denimo prosimo, naj sklanja določen samostalnik, bo to tudi storil. Vgrajeno ima tudi znanje o slovničnem spolu in po identifikaciji sogovornikovega spola bo v nadaljevanju uporabljal zanj ustrezne slovnične oblike. Natančneje urejena je tudi identifikacija vprašalnic v vprašalnih stavkih, kar programu omogoča izbiro odgovora iz ustreznega dela programske baze.

Zaenkrat program nima nikakršnih semantičnih informacij, na osnovi katerih bi lahko vodil koherentno povezan pogovor (potreboval bi podatke o pomenskih razmerjih med besedami, tipičnih kolokacijah ipd.). Ker v tem trenutku stavčnočlenska analiza ne poteka niti na ravni stavka, prav tako še niso izpolnjeni pogoji za nadstavčno povezovanje, razdvoumljanje pomenskih navezav itd.

<sup>1</sup>S Klepcem lahko klepetamo na internetni strani <http://klepec.amebis.si>.

*Uporabnik: sklanjaj voli*  
 Klepec: Je voli samostalni? *Uporabnik: da*  
 Klepec: Na tvojem mestu ne bi bil tako prepričan. *Uporabnik: sklanjaj vol*  
 Klepec: vol, vola, volu, vola, pri volu, s/z volom *Uporabnik: sklanjaj besedo vol v monžini*  
 Klepec: Nekoč sem imel domačo žival. Jo imaš tudi ti? *Uporabnik: nočem klepetati*

Primer pogovora 1: Sklanjanje.<sup>2</sup>

### 3.2. Poznavanje specifik klepetalniškega diskurza

Klepetalniški diskurz (diskurz, kakršen se pojavlja pri realnočasovni pisni komunikaciji prek interneta) ima določene specifikke, ki jih je potrebno pri snovanju programiranega sogovornika upoštevati. Najpomembnejše so naslednje točke:

#### 3.2.1. Stalne točke v poteku klepeta

Klepec ima vgrajeno osnovno védenje o poteku klepeta – pozna denimo koncepta pozdravljanja ter predstavljanja. V začetni fazi pogovora skuša dobiti od uporabnika informacijo o njegovem imenu, podatek si zapomni in ga v nadaljevanju pogovora po potrebi uporabi. Če je podano ime že v bazi osebnih lastnih imen, Klepec v nadaljevanju pogovora za uporabnika uporablja ustrezne slovnične oblike, če pa imena ne prepozna, se pozanima, ali gre za moško ali žensko ime. Kadar ne dobi enoznačnega odgovora, privzame, da je uporabnik moškega spola.

*Klepec: Mimogrede, mi zaupaš tvoje ime?*  
*Uporabnik: na*  
 Klepec: Še nisem slišal. Kakšno ime pa je to - moško ali žensko?  
*Uporabnik: brez spola*  
 Klepec: Čakam, čakam, čakam ...  
*Uporabnik: kaj čakaš?sneg?*  
 Klepec: Čakam, čakam, čakam ... Bo že kaj?  
*Uporabnik: ne vem če bo kej snežilo*  
 Klepec: Ne grem se več. Zame si moški, pa pika.

Primer pogovora 2: Pridobivanje podatkov o imenu ter spolu uporabnika.

#### 3.2.2. Menjavanje vlog v komunikaciji

Klepec upošteva načelo menjave vlog sprašujočega ter odgovarjajočega v pogovoru (včasih le odgovarja, včasih sprašuje, včasih združi oboje).

#### 3.2.3. Odkloni od knjižne norme v smer govornega jezika

V internetnih klepetih so odstopi od knjižne norme pogosti na vseh jezikovnih ravneh. Pojavlja se opuščanje

ločil ter velikih začetnic, leksika ter skladnja se približujeta govornemu jeziku, prepoznavo uporabnikovega vnosa pa otežujejo še tipkarske in druge pravopisne napake ter opuščanje šumnikov.

Leksikalno bazo, na kateri temelji Klepec, za potrebe razvoja programa dopolnjujemo z najpogostejšimi odkloni od knjižne norme, in sicer z leksikalnimi enotami, ki:

- se v zapisu približujejo govoru (npr. različni zapisi zaimka *jaz – jst, jest, js ...*, *kako – kk, kak, kuko ...*, reducirane glagolske oblike *gledal – gledu* ipd.),
- z opuščeni šumniki nimajo enakopisne knjižne ustreznice (npr. *delaš – delas*),
- odstopajo od pravopisnih pravil (npr. *življenjski – življenski*).

*Uporabnik: kaj delas*  
 Klepec: Tega pa žal še ne vem. *Uporabnik: kako*  
 Klepec: Hitro in z lahkoto. *Uporabnik: zakaj*  
 Klepec: Pomisli malo. *Uporabnik: a si nor*  
 Klepec: Ne bi raje uporabil lepo vprašalnico "ali"? *Uporabnik: ali si nor*  
 Klepec: Kaj pa vem. *Uporabnik: saj malo*  
 Klepec: Govoriva o čem drugem, prosim. *Uporabnik: kaksno vreme bo jutri*  
 Klepec: Jutri bo lepše. *Uporabnik: kaj pa pojutrišnjem*  
 Klepec: Ali nimaš nobenega pametnejšega vprašanja?

Primer pogovora 3: Odstopi od knjižne norme.

#### 3.2.4. Vpletanje tujejezičnih prvin

V pogovorih se relativno pogosto pojavljajo tudi jezikovne prvine iz (večinoma) angleškega jezika. V trenutni fazi razvoja je program omejen zgolj na identifikacijo rabe tujega jezika (kljub temu, da Klepec v spodnjem primeru trdi, da zna v angleščini tudi komunicirati). Identifikacija jezika temelji na bazi ključnih tujejezičnih besed, ki se v takšnih primerih najpogosteje pojavljajo, in je posledično uspešna le pod pogojem, da je uporabniški vnos tujejezičnih enot pravopisno ustrezen.

*Uporabnik: how are you?*  
 Klepec: Poglej, poglej - saj znaš angleško!?  
*Uporabnik: a ti ne znaš?*  
 Klepec: Veš da.  
*Uporabnik: no pa se grema pogovarjat angleško,ok*  
 Klepec: Ti pa znaš. Si se učil angleško!?  
*Uporabnik: ja*  
 Klepec: To pa kar težko verjamem.  
*Uporabnik: pa nič*

Primer pogovora 4: Uporaba tujega jezika.

#### 3.2.5. Uporaba emotikonov

Ena od specifik klepetalniškega jezika je tudi raba klepetalniških kratic (npr. *lol, wtf, brb ...*) in emotikonov (grafičnih simbolov za čustva). Klepec zaenkrat

<sup>2</sup>Vsi primeri, ki jih navajava v tem članku, so zbrani iz baze pogovorov, ki so s Klepcem potekali v obdobju od 1. 10. do vključno 19. 12. 2005. Zaradi varstva uporabniških pravic izpuščava imena, kjer se pojavljajo, ter navajava le kratke izseke posameznega komunikacijskega niza.

prepoznavna ter mestoma tudi že uporablja kratice ter emotikone, ki se v komunikaciji najpogosteje uporabljajo.

Klepec: Mislim da se mi je pokvarilo neko vezje. Si kaj sumljivega opazila?  
Uporabnica: Ne čisto nič, samo ogenj pa dim. :)  
Klepec: :))

Uporabnik: lol  
Klepec: In čemu se tako smeješ?  
Uporabnik: a kr poznash kratico laughing out loud?  
Klepec: Veš da ne.  
Uporabnik: o kok si talent.. dei grewa u angleščini govort

Primer pogovora 5: Uporaba emotikonov ter klepetalniških kratic.

### 3.3. Podatkovne baze

Ta tema se delno povezuje z vsemi ostalimi, saj vse znanje, na podlagi katerega program operira, izvira iz podatkovnih baz. Na tem mestu govoriva specifično o bazah znanja o svetu, iz katerih naj bi Klepec v prihodnosti črpal odgovore na najrazličnejša uporabniška poizvedovanja. Kot rečeno je glede informacijske podprtosti program še na začetku razvoja. Zaenkrat odgovarja na vprašanja z *kdo* in *kaj*, za katera dobiva podatke iz Amebisove interne definicijske baze, ki vsebuje okrog 100.000 gesel.

Uporabnik: kdo je marjan  
Klepec: Marjan je hrib nad Splitom  
Uporabnik: ne marjan je moj stric  
Klepec: Kolikor vem, je stric očetov ali materin brat.  
Uporabnik: tudi ti ves nekaj o tem  
Klepec: Pustiva zdaj to. Raje mi povej še kaj o sebi.

Uporabnik: kaj je to zastava ?  
Klepec: izročitev stvari za zavarovanje upnikove terjatve, če se ne motim  
Uporabnik: kaj pa hladilnik ?  
Klepec: Me sploh ne zanima.  
Uporabnik: kaj je to nevihta ?  
Klepec: Slišal sem, da je nevihta neurje, pri katerem močno dežuje, naliv spremljajo bliski.  
Uporabnik: smrad  
Klepec: Ti lahko povem, da je smrad neprijeten vonj.  
Uporabnik: medmet  
Klepec: Hej! Nekaj o tem pa vem tudi jaz! medmet je nepregibna beseda, ki izraža duševno stanje ali posnemanje glasov, zvokov. Imam prav?  
Uporabnik: pa že

Primer pogovora 6: Vprašanja z *kdo* in s *kaj*.

### 3.4. Antropomorfizacija

Določena mera podobnosti v mišljenju in življenjski izkušnji je predpogoj za uspešno komunikacijo, zato je dobro, da je Klepec vsaj v nekaterih segmentih podoben človeku, drugačnega ga pravzaprav niti ni mogoče zasnovati. V trenutni fazi je namenjen klepetanju in zabavi, zato je zelena mera antropomorfizacije višja: uporabnik program ocenjuje glede na to, koliko je podoben človeškemu sogovorniku – v kolikšni meri so

njegovi odgovori primerni oz. smiselni v kontekstu pogovora, kako se odziva na določen tip replik itd. V prihodnosti se bo program razvijal v smer interaktivne podatkovne baze, kar pomeni, da se bo težišče njegove uporabe premaknilo. Pričakovano je, da se bodo s tem premikom spremenila tudi pričakovanja uporabnikov in njihovi kriteriji za presojanje kvalitete programa, s tem pa posledično zmanjšala potreba po antropomorfizaciji.

Nekaj zametkov simulirane osebnosti Klepec že ima, izraža želje, preference in tudi razpoloženje, odziva se na žaljenje itd. Antropomorfna lastnost je tudi smisel za humor. Zanimivo vprašanje, ki zaenkrat ostaja neodgovorjeno, pa je: je bolje, da program pri določenih temah simulira konkretne izkušnje (s spolnostjo, športom, čustvi ...), ali naj k temam pogovora pristopa, kot da dejanskost zaznava prek vmesnih medijev (da je informacije le nekje 'videl' oz. 'prebral') in s tem poudarja svoje specifikke napram človeškim sogovornikom.

Uporabnik: si živčen?  
Klepec: Šur.  
Uporabnik: zakaj?  
Klepec: Ker je res.  
Uporabnik: ne bi smel biti  
Klepec: Kako mi lahko kaj takega sploh govoriš?  
Uporabnik: si program :)  
Klepec: Ja.  
Uporabnik: ergo ne bi smel biti živčen

Primer pogovora 7: Programi in živčnost.

## 4. Kako Klepec deluje

### 4.1. Notranja zgradba

Pokazalo se je, da klasični programski jeziki (kot npr. C++) ne omogočajo hitre in uspešne gradnje sistemov za komunikacijo v naravnih jezikih, še posebej ne v pregibnih, kakršna je slovenščina. Ti programski jeziki kot prvo ne omogočajo enostavne vgradnje podatkov oz. znanj o nekem naravnem jeziku, še bolj problematično pa je, da način programiranja in delovanja, ki temelji na (vnaprej) določenem poteku programskih ukazov, ni dovolj učinkovit in primeren za izdelavo sistemov dialoga, kjer je nemogoče vnaprej določiti interakcijo s sogovornikom.

Zato smo za potrebe sistema, kakršen je Klepec, razvili in izdelali poseben programski jezik s trenutnim delovnim imenom K2.0, ki omogoča učinkovitejše programiranje sogovornikov za določen naravni jezik. Trenutno je K2.0 podprt le z modulom za slovenski jezik.

#### 4.1.1. Osnovni princip delovanja

V osnovi je Klepec vzorčno vodeni sistem. To pomeni, da je sestavljen iz določenega števila vzorcev oz. zapisov, ki jih program primerja s sogovornikovim vnosom, potem pa na osnovi določenih meril izbere tistega, ki je temu vnosu najbolj podoben, in izbere enega od odgovorov, ki jih ta zapis vsebuje.

\$ kako si | kako si kaj | kako kaj  
> Dobro. | Slabo. | Odlično.  
\$ kako ti je ime | kako se imenuješ

```
> Klepec. | Ime mi je Klepec. | Klepec se imenujem.
$ kaj znaš > Vse, kar so me naučili v šoli.
```

Slika 1: Nekaj enostavnih vzorcev oz. zapisov.

#### 4.1.2. Posploševanje vzorcev

Eden osnovnih problemov pri razumevanju uporabniškega vnosa ter izdelavi in primerjavi vzorcev je, da lahko semantično načeloma enoznačne trditve izražamo na več različnih načinov (kar se tiče izbire leksike ter skladenjskih struktur). Zato je potrebno za praktično rešitev tega problema omogočiti določeno stopnjo posploševanja vzorcev.

```
$
kako ti je ime
mi lahko poveš kako ti je ime
mi lahko prosim poveš kako ti je ime
mi prosim lahko poveš kako ti je ime
mi lahko poveš prosim kako ti je ime
ali mi lahko poveš kako ti je ime
ali mi lahko prosim poveš kako ti je ime
...
mi lahko zaupaš kako ti je ime
mi lahko prosim zaupaš kako ti je ime
...
ali mi lahko zaupaš kako ti je ime
...
povej mi kako ti je ime
prosim povej mi kako ti je ime
...
zaupaj mi tvoje ime
...
kako se imenuješ
...
>
Klepec.
```

Slika 2: Veliko število možnih vzorcev.

Posplošitev vzorcev lahko v programskem jeziku K2.0 izvedemo s pomočjo različnih programskih elementov in ukazov, odvisno od tega, kaj želimo posplošiti. Najpogosteje se za posplošitev uporabljajo operatorji tipa %B (poljubna beseda), %S (poljubno število), %L (poljubno ločilo), %0 (nič ali poljubno število enot), %1 (ena poljubna enota), %2 (ena ali dve poljubni enoti), %3 (ena, dve ali tri poljubne enote) itd. To nam omogoča, da željeni nabor vnosov zajamemo s precej manjšim številom vzorcev.

```
$
%0 kako ti je ime
%0 zaupaj mi %0 tvoje ime
%0 mi zaupaš %0 tvoje ime
...
%0 kako se %0 imenuješ
...
>
Klepec.
```

Slika 3: Posplošitev vzorcev.

Posplošitev lahko izvedemo tudi s pomočjo operatorja morfosintaktične informacije, s katerim še dodatno zmanjšamo potrebno število primerjalnih vzorcev.

```
$
%0 kako ti je ime
%0 zaupati[*] %0 tvoje ime
%0 povedati[*] %0 tvoje ime
...
%0 kako se %0 imenuješ
...
>
Klepec.
```

Slika 4: Dodatna posplošitev.

S pomočjo operatorja morfosintaktične informacije lahko kontroliramo točno določene besedne vrste in njihove oblike (npr. pri samostalniku vrsta, spol, sklon in število), tako na vhodni kot tudi izhodni strani. Pri tem pogosto potrebujemo pomoč operatorjev #1 (prva spremenljivka v vhodnem nizu), #2 (druga spremenljivka v vhodnem nizu) itd., ki nam omogočajo prenos poljubnih vhodnih besed v izhodni niz.

```
$ kako se sklanja %B[S??ei] | sklanjaj %B[S??ei]
>
#1[S??ei], #1[S??er], #1[S??ed], #1[S??et],
pri #1[S??em], sVz #1[S??eo]

$ kako se sklanja %B[G?n] | sklanjaj %B[G?n]
>
Glagoli se ne sklanjajo!
```

Slika 5: Sklanjanje besed.

#### 4.1.3. Kontrola poteka

Ker je v jezik K2.0 vgrajena naključnost pri izbiri enakih ali zelo podobnih vzorcev, imamo za kolikor toliko kontrolirano izbiro določenih vzorcev ali poteka dialoga na voljo številne dodatne operatorje in ukaze. Med te sodita določitev statične prioritete posameznega vzorca, npr. (7), ter določitev pogoja primerjave vzorca in izbire izhodnega niza, npr. [spol == 1], pri čemer lahko uporabljamo določene lokalne, globalne ali systemske spremenljivke.

```
$ (7) [vreme == 0]
kakšno bo %0 vreme %0
kakšna %0 vremenska napoved %0
>
V Sahari bo precej suho. <vreme = 1>
[spol == 1] Počakaj, pa boš videl. <vreme = 1>
[spol == 2] Počakaj, pa boš videla. <vreme = 1>
```

Slika 6: Uporaba statične prioritete, pogojev in spremenljivk.

Če želimo kontrolirati potek dialoga na nivoju celotnega nabora ali skupine vzorcev, ne samo znotraj posameznega vzorca, imamo na voljo nekatere ukaze, s katerimi lahko preskočimo primerjavo skupine vzorcev

oz. omejimo primerjavo le na točno določeno skupino, ki je izbrana na osnovi prejšnjega poteka pogovora. Pri tem pogosto uporabljamo naslove, npr. {ime}, in akcije, ki se izvedejo ob branju ustreznega vzorca, npr. <goto {ime}>.

```

//***** kontrola imena
$> <goto {ime_konec}>
$> {ime} <null>

$ (7)
%0 ime %0 mi %0 je %0 %B[Slmei] %0
>
Lepo možko ime.<ime = #1><spol = 1><goto {zacetek}>

$> <stop/goto {ime}>
$> {ime_konec} <null>
//*****

```

Slika 7: Kontrola poteka dialoga.

#### 4.1.4. Druge funkcije

Če želimo poleg nekoristnega klepeta v odgovore vključiti tudi koristne informacije, shranjene v določenih podatkovnih bazah, lahko uporabimo (uporabniško definirane) klice funkcij, ki te podatke preverjajo in berejo. Tako lahko programiranega sogovornika uporabimo v funkciji informacijskega sistema, ki nam koristne informacije podaja v zelo zanimivi obliki.

Poleg funkcij lahko pri programiranju uporabimo še številne zanimive in koristne mehanizme jezika K2.0, za predstavitev katerih pa je v tem članku premalo prostora.

```

$ (6)
%0 kaj %0 je %0 @Preveri(__pot, "BAZA")
>
#4 je @PreberiBAZA(__pot, #4)

```

Slika 8: Uporaba funkcij.

## 4.2. Uporabniški doprinos

### 4.2.1. Eliza – pripisovanje posebnega komunikacijskega namena

Brez uporabnikove pripravljenosti sodelovati v komunikaciji je še tako dobro zasnovan program obsojen na neuspeh, in obratno, povsem osnovne aplikacije lahko dosega velike uspehe.

To dejstvo je v svoj prid izkoristil tudi Joseph Weizenbaum, snovalec prve znane programirane sogovornice. Eliza, kot je svoj izdelek poimenoval po liku iz znanega dela *Pygmalion*, ima povod za nastanek v psihiatriji; tam se je v tistem času (gre za obdobje 60-ih let prejšnjega stoletja) izredno rado prakticiralo t. i. aktivno poslušanje – psihiater s parafraziranjem pacientovih izjav ter spodbudnimi signali pomaga pacientu, da sam razreši problem, ki ga teži. Metoda je Weizenbauma navdihnila, da je ustvaril program, ki z uporabnikom komunicira na enak način. Kot simulacija psihiatrinje je dobila programirana Eliza v pogovoru posebno hierarhično vlogo, ki jo avtor takole opisuje:

»Ta tip komunikacije je bil izbran, ker je psihiatrični intervju eden redkih primerov /.../ komunikacije v

naravnem jeziku, v kateri je enemu od udeležencev pogovora dovoljeno zavzeti položaj, po katerem o svetu ne ve skoraj ničesar. Če npr. nekdo pove psihiatru 'Šel sem se vozit s čolnom,' psihiater pa odvrne 'Povejte mi kaj o čolnih,' ne bomo sklepali, da psihiater ne ve ničesar o čolnih, pač pa da pogovor vodi v to smer z določenim namenom.« (Weizenbaum, 1966. Prevedla in podčrtala Š. Arhar.)<sup>3</sup>

### 4.2.2. Osmišljanje izjav kontekstu

Da je komunikacija s programiranimi sogovorniki sploh mogoča, gre torej pripisati predvsem človeški sposobnosti oz. potrebi po osmišljanju izrekov v kontekstu diskurza. Pri snovanju programa, ki se v pogovoru predstavlja za uporabniku hierarhično enakovrednega sogovornika, je potrebno predvideti takšen način komunikacije, da je na vseh ravneh omogočeno čim bolj uspešno osmišljanje v kontekstu. Pri razvijanju programa Klepec skušamo to doseči na več načinov:

- s premišljenim snovanjem baze odgovorov; le-ti morajo biti obenem dovolj splošni, da ustrezajo čim več potencialnim kontekstom, obenem pa dovolj specifični, da omogočajo nadaljevanje pogovora,
- z uporabo humorja v Klepčevih izjavah; diskurz, ki se na tak način razvije, omogoča boljše osmišljanje Klepčevih izjav, saj je uporabnik do njih tolerantnejši, če jih ne jemlje povsem resno,
- z uporabo poskusa menjave teme v primerih, da uporabnikovega vnosa Klepec ne prepozna.

V nadaljevanju navajava tri primere uspešnega osmišljanja v kontekstu, kjer zgoraj naštetje strategije v kombinaciji z uporabnikovim vztrajanjem pri temi, in seveda njegovo pripravljenostjo sodelovati v igri komunikacije, privedejo do zabavnih dialoških izmenjav:

```

Uporabnik: koliko je ura
Klepec: Je čas tvoj gospodar?
Uporabnik: ne
Klepec: Slišati si zelo prepričan v to.
Uporabnik: ampak mene vseeno zanima koliko je ura
Klepec: Meni čas ne pomeni ničesar. Vsaj dokler sem v tej škattli.
Uporabnik: da ti ne bom virusa poslal

```

Primer pogovora 8: Časovna relativnost.

```

Uporabnik: a ti veš kje je grčija
Klepec: Ja, stari.
Uporabnik: kje
Klepec: V zraku.
Uporabnik: kje?
Klepec: Za vogalom.
Uporabnik: kje
Klepec: Tukaj, kje pa drugje.
Uporabnik: kreten

```

Primer pogovora 9: Kje je Grčija.

<sup>3</sup>Eliza je za klepet na voljo na straneh Inštituta Jožef Stefan, <<http://www-ai.ijs.si/eliza/>>.

Uporabnica: me ljubiš?  
 Klepec: Ali ni danes prav lepo vreme?  
 Uporabnica: me ljubiš dragi?  
 Klepec: Uf, kako je vroče. Morda se mi je pokvaril ventilator?  
 Uporabnica: daj no me ljuubiš  
 Klepec: Zakaj pa "no"?  
 Uporabnica: ja me ljubiš?  
 Klepec: Hej, ti mi pa že ne boš tega govorila.  
 Uporabnica: daj me ljubi?  
 Klepec: Teško bi dal.  
 Uporabnica: ljubiš me  
 Klepec: Spremeniva temo!

Primer pogovora 10: Izpovedovanje ljubezni.

#### 4.2.3. Uporabniški odnos do programa

Ker je vsaka komunikacija mnogo več kot le izmenjava jezikovnih nizov, želiva na tem mestu opozoriti še na sociološke ter psihološke vidike komunikacije med človekom ter programiranim sogovornikom. Dejstvo je, da raziskave uporabniškega odnosa do programov za klepetanje kažejo nenavadno visoko stopnjo agresivnega govora, pri katerem prednjačijo raznovrstne kletvice, žalitve ter provokacije. Foner (1997) v svoji sociološki raziskavi omenja uporabniški strah, da bi bil stroj (čeprav le po znanju) boljši od človeka. De Angeli in Carpenter (2005) ugotavljata, da izvira žaljiv odnos uporabnika do programiranega sogovornika iz privzetega razmerja moči, ki postavlja uporabnika kot človeško bitje na višji položaj od programa, ki se trudi biti človeku podoben. Poleg teh dveh premis je potrebno dodati še eno: samocenzura, ki je v medčloveški komunikaciji seveda stalno prisotna, v dialogu s programom – za katerega vemo, da nima ne čustev ne spomina na pogovor – popusti.<sup>4</sup>

Ker je predvidljivo, da bodo nekateri uporabniki do Klepca žaljivi še naprej, je za prepričljivo antropomorfno nujno, da se program na žalitve ustrezno odziva. Zaenkrat je Klepec pred agresivnim govorom še dokaj nemočen, reagira le na uporabo nekaterih vulgarnih besed, ne simulira pa užaljenosti ali jeze, ki bi ju v podobnih primerih pričakovali od človeških uporabnikov.

Uporabnik: ti si peder  
 Klepec: Zakaj me žališ?  
 Uporabnik: ti si idiot!!!!!!

<sup>4</sup>Da so uporabniki žaljivi tudi pri nas, je pokazala raziskava Klepcu najpogosteje zastavljenih vprašanj, ki smo jo opravili na vzorcu pogovorov, ki so potekali med 1. 10. ter 19. 12. 2005. Uporabniška vprašanja smo kategorizirali na poizvedovalna (sprašujejo po informaciji) ter odločevalna (odgovor je *da oz. ne*). Žalitve (ki so vprašanja le napol, saj glavni namen uporabnika ni poizvedovanje, pač pa izražanje negativnega mnenja o programiranim sogovorniku) najdemo le med odločevalnimi vprašanji. V skupini 550-ih odločevalnih vprašanj daleč največji delež zajemajo vprašanja oz. žalitve o spolnih preferencah, predvsem homoseksualnosti (49 %). Na drugem mestu so vprašanja o partnerskem statusu (17 %), na tretjem mestu vprašanja o znanju angleščine (14 %), na četrtem pa ponovno žalitve, tokrat umskih sposobnosti (7 %). Ostale kategorije so manj zastopane.

Klepec: Koliko časa pa misliš, da bom še prenašal tvoje žaljivke?  
 Uporabnik: DO KONCA PIČKA TI MATERINA!!!!!!  
 Klepec: Tako prostaškega sogovornika pa že dolgo nisem imel.  
 Uporabnik: kr navad se ti zafukanc zafukani

Primer pogovora 11: Agresivni govor.

## 5. Zaključek

Na več mestih je že bilo rečeno, da je program Klepec projekt v procesu in da bo za njegov razvoj ter posledično kvalitetnejše delovanje potrebnega še mnogo dela. Trenutno največji izziv predstavlja nadgradnja, ki bo programu omogočala držati rdečo nit pogovora in s tem kohezivno ter koherentno dialoško komunikacijo, kakršna je v navadi med človeškimi sogovorniki. Kratkoročnejši cilji so vgradnja izboljšanega lematizatorja ter jezikovnega analizatorja, prestrukturiranje leksikalne baze, da bo omogočala dopolnitev s semantičnimi ter kolokacijskimi informacijami ter postopno dodajanje podatkovnih baz z znanjem o svetu. Načrti izboljšave antropomorfnosti pa gredo predvsem v smer načrtovanja programskih modulov za različna Klepčeva razpoloženja ter različne tipe diskurza glede na jezikovne tendence uporabnika.

## 6. Literatura

- Ahrenberg, L., A. Jönsson in N. Dahlbäck, 1990. Discourse Representation and Discourse Management for a Natural Language Dialogue System. V J. Allen, B. Miller, E. Ringger in T. Sikorski (ur.), *Proceedings of the second nordic conference on text comprehension in man and machine*. <<http://www.ida.liu.se/~arnjo/papers/notex-90.pdf>>
- Bickmore, T. in J. Cassell, 2000. »How about this weather?«: Social Dialogue with Embodied Conversational Agents. *Proceedings of the AAAI Fall Symposium on Socially Intelligent Agents*. <<http://www.misu.bmc.org/~bickmore/publications/SIA00.pdf>>
- De Angeli, A. in R. Carpenter, 2005. Stupid computer: Abuse and social identities. Objavljeno na <<http://www.agentabuse.org/deangeli.pdf>>.
- Foner, L. N., 1997. Entertaining Agents: a Sociological Case Study. *AGENTS '97: Proceedings of the first international conference on Autonomous agents*. Kalifornija: ACM Press. 122–129. <<http://foner.www.media.mit.edu/people/foner/Reports/Agents-97/Julia.pdf>>
- Morkes, J., H. K. Kernal in C. Nass, 1999. Effects of Humor in Task-Oriented Human-Computer Interaction and Computer-Mediated Communication. *Human-computer Interaction* vol.14 / št. 4: 395–435.
- Romih, M. in P. Holozan, 2002. Sporazumevanje z računalnikom v naravnem jeziku. V T. Erjavec in J. Gros (ur.), *Jezikovne tehnologije – Mednarodna multi-konferenca Informacijska družba*. Ljubljana: Inštitut »Jožef Stefan«. 168.
- Weizenbaum, J., 1966. ELIZA: A Computer Program For the Study of Natural Language Communication Between Man and Machine. *Communications of the ACM* 9/1: 36–45.