

# Uporaba programskega jezika K2.0 v sistemih dialoga

**Miro Romih**

Amebis d.o.o.  
Bakovnik 3, 1241 Kamnik, Slovenija  
miro.romih@amebis.si

Vzporedno z razvojem informacijskih in telekomunikacijskih tehnologij v ospredje prihajajo tudi same vsebine oz. informacije, do katerih želimo dostopati ali jih posredovati drugim. Čeprav ta ponudba zaostaja za tehnologijo, pa je še večji zastoj opazen na področju razvoja novih komunikacijskih vmesnikov. Sistemi menijev in ikon, ki jih na ekranu krmilimo s pomočjo tipkovnice ali miške, že vrsto let ostajajo glavni način upravljanja aplikacij. Za večino naših zahtev je tak način komunikacije z napravami povsem zadosten oz. celo najustrežnejši. Je pa vrsta situacij, ko bi nam klasičen način sporazumevanja s pomočjo naravnega (slovenskega) jezika bistveno olajšal in pohitril delo. Ne glede na to, ali bi bil način komuniciranja pisen ali govoren, bi problem lahko rešili z razvojem ustreznega sistema dialoga.

Že pri zelo enostavnih primerih dialoga se pokaže, da je tak sistem težko napisati v kakem znanem programskem jeziku. Že sam način klasičnega programiranja ni najprimernejši za razvoj vzorčno vodenih sistemov, poleg tega pa je ob vsem tem potrebno vzporedno poskrbeti za vso jezikovno infrastrukturo, ki naj bi jo nek dialog vseboval. Kaj to pomeni, ve vsak, ki se je že kdaj poskusil spopasti s tem problemom - glavnina dela je namenjena obvladovanju osnovnih elementov jezika. V podjetju Amebis smo rešitev videli v razvoju posebnega programskega jezika, s pomočjo katerega bi lažje obvladovali probleme komunikacije v naravnem (slovenskem) jeziku. Zato smo v okviru projekta Kolos, katerega cilj je izgradnja celovitega informacijskega sistema za komunikacijo v naravnem jeziku, razvili poseben programski jezik z delovnim imenom K in trenutno verzijo 2.0. Pri tem smo se le delno zgledovali po nekaterih rešitvah v tujini, v glavnem pa je rešitev zasnovana na podlagi lastnih idej in prilagojena našim jezikovnim virom.

Glavna prednost, ki jo ponuja programski jezik K, je integriranost slovenske jezikovne baze (trenutno okrog 160.000 osnovnih besed s pripadajočimi besednimi oblikami) v njegovo sintakso. Zaradi tega je primeren zgolj za pisanje dialoga v slovenskem jeziku, kar pa je tako ali tako naš cilj, zato ta omejitev ni moteča. S to izgubo jezikovne neodvisnosti pa na drugi strani pridobimo celo vrsto prednosti. Programiranje v programskem jeziku K se razlikuje od načina programiranja v klasičnih programskih jezikih. Predvsem zato, ker gre za izrazito vzorčno voden sistem, kjer potek dogodkov ni vnaprej določen. Delno se ta potek sicer lahko kontrolira, ker pa je eden od glavnih elementov jezika ravno vgrajena naključnost, enako zaporedje akcij skoraj vedno prinese drugačen rezultat oz. potek dialoga.

Uporaba sistema je zelo enostavna. Poleg definicije programskega jezika K, ki se ga v kratkem času lahko nauči uporabljati tudi tisti, ki nikoli prej ni programiral v kakšnem drugem programskem jeziku, sistem sestavljajo štiri osnovne datoteke. Poleg datoteke, ki vsebuje jezikovne podatke o slovenskem jeziku, je tu še program Kolos.exe, namenjen testiranju modulov v jeziku K, programsko jedro Sukus.dll, ki interpretira oz. prevaja in izvaja programsko kodo, napisano v jeziku K, ter knjižnica funkcij Suplus.dll, v kateri so že vsebovane nekatere dodatne osnovne uporabniške funkcije, s pomočjo katerih je izvajanje kode še bolj učinkovito.

Kot primer vzemimo uporabo baze podatkov, v kateri imamo navedene definicije različnih pojmov (miza – kos pohištva, noga – spodnja okončina...). Recimo, da želimo uporabiti te definicije v odgovorih sistema dialoga. Če bi pokrili le tip vprašanja »Kaj je X?« (»Kaj je miza?«), bi to enostavno rešili v vsakem navadnem programskem jeziku. Odgovor bi generirali tako, da bi X-u (mizi) enostavno prilepili ustrezno definicijo: »"X" je "definicija".« (»Miza je kos pohištva«). Če pa bi želeli odgovoriti na vprašanja, kot so »Kaj so X(mize)?«, »Kaj več o X(mizi)?«, »Kaj več o X(mizah)?« in podobno, pa bi morali za vsako geslo v bazi podatkov poznati ustrezne besedne oblike. V jeziku K z vgrajeno jezikovno bazo vsako od teh vprašanj pokrijemo z enim samim splošnim vzorcem, npr. »Kaj so %B[So?mi]?«, pri čemer lahko kot odgovor generiramo različne odgovore: »#1 je @definicija(\_\_pot, "#1").«, »O #1[???mm] ne vem kaj dosti.« itd. Pri tem %B[So?mi] predstavlja obči samostalnik poljubnega spola v imenovalniku množine, #1 konkretno uporabljeno besedo v vprašanju (X), @definicija(\_\_pot, "#1") klic funkcije na imeniku \_\_pot, ki vrne definicijo v vprašanju uporabljenega samostalnika, #1[???mm] pa obliko uporabljenega besede v mestniku množine.

Že v osnovi ni bilo zamišljeno, da bi rešitve, ki bi vključevale dialog v slovenskem jeziku, v celoti programirali v jeziku K. Moduli v tem jeziku so le dopolnilo oz. dodatni vgradni elementi aplikacijam, napisanim v drugih programskih jezikih. S tem je razvijalcem dana večja svoboda pri izbiri ustreznih orodij in s tem večja fleksibilnost pri izvedbi različnih funkcij programskih rešitev. Tipična uporaba bi bili npr. razni avtomatizirani agenti, informacijski portali, dopolnilo komunikacijskih modulov v sistemih umetne inteligence itd.

Sedanja verzija programskega jezika K že sedaj vsebuje celo vrsto koristnih elementov, s katerimi si lahko pohitrilo in olajšamo gradnjo sistemov dialoga. Seveda so se ob njegovi uporabi pokazale tudi nekatere pomanjkljivosti in nerodnosti, ki jih bomo upoštevali pri razvoju novih verzij. Prav tako imamo že zdaj cel kup idej in načrtov za nadaljnje izboljševanje in dopolnjevanje, ki jih zaradi časovnih omejitev nismo uspeli realizirati v dosedanjih verzijah. Ena takih izboljšav, ki bo pomenila velik korak naprej v smeri večje uporabnosti, bo zagotovo vgradnja stavčnega analizatorja skupaj s pomensko bazo besed. S tem bodo razvijalci sistemov dialoga dobili še močnejše orodje za realizacijo konkretnih aplikacij, ki bodo uporabnikom omogočile lažji in hitrejši dostop do želenih informacij. Za zdaj razvojni sistem deluje le v okolju Windows in še ni na voljo zunanjim uporabnikom oz. razvijalcem, računamo pa, da bo z nekaterimi izboljšavami in dopolnitvami prej ali slej pod ustreznimi pogoji na voljo tudi tem.