

TEHNOLOGIJE AGENTOV

– UPORABA V ELEKTRONSKEM POSLOVANJU

Denis Trček

Institut "Jožef Stefan", Jamova 39, 1000 Ljubljana

E-pošta: denis.trcek@ijs.si

Povzetek

Elektronsko poslovanje je prineslo precej novih paradigem v poslovanje in ena od njih so tehnologije agentov. Tovrstne tehnologije združujejo več področij, ki segajo od objektnega programiranja kot takega prek komunikacij, umetne inteligence pa do varnosti. V prispevku je podan pregled omenjenega področja ter primer uporabe v nabavni poslovni funkciji.

Ključne besede: elektronsko poslovanje, tehnologije agentov.

Abstract

E-BUSINESS APPLICATIONS OF AGENT TECHNOLOGIES

Many new paradigms have been introduced recently with the proliferation of e-business applications, and agent technologies are among the most important ones. They join complementary areas that include object programming, communications, artificial intelligence and security. In the article an overview of this field is given with the emphasis on use of agent technologies in supply chain management.



Uvod

Danes poznamo kar nekaj področij, kjer se srečujemo s tehnologijami agentov, npr. upravljanje omrežij in izobraževanje. Na področju elektronskega poslovanja so agenti postali dejstvo predvsem pri dejavnostih trženja (npr. pri iskanju in določanju ciljnih skupin prek svetovnega spleta), nabave (npr. pri iskanju najugodnejših ponudnikov), vse bolj prodirajoč segment mobilnih komunikacij pa širi uporabo teh tehnologij tudi v sektor turizma. Tak primer je projekt CRUMPET (Creation of User Friendly Mobile Services Personalized for Tourism) [FIPO1a], kjer bodo podprte najnovejše prilagodljive nomadske storitve preko različnih vrst infrastrukture (IP, WLAN, GSM, GPRS in UMTS).

Najbolj poznana okolja za upravljanje agentov (OUA) so:

- Grasshopper (podrobnosti so dosegljive na naslovu <http://www.ikv.de>).
- FIPA-OS (podrobnosti so dosegljive na naslovu <http://fipa-os.sourceforge.net>).
- JADE (podrobnosti so dosegljive na naslovu <http://sharon.csel.it/projects/jade>).
- ZEUS (podrobnosti so dosegljive na naslovu <http://www.labs.bt.com>).

Koncepti vseh teh agentnih okolij so si podobni. Namen tega članka je podati tipično strukturo in delovanje agentov, nato pregled okolij za upravljanje agentov in možnost njihove uporabe pri modeliranju ter podpori poslovnih procesov.

Tehnologije agentov – lastnosti in arhitekture

Evolucija na področju agentov ima korenine v začetkih objektnega programiranja, ko smo pri programski realizaciji reševanja problemov iz realnega sveta začeli uporabljati koncept objektov, to je zaključenih enot programske kode, katerim so dodeljeni določeni podatki in pa pripadajoče metode. Razvoj globalnega omrežja in mobilnih komunikacij pa je pripeljal do tega, da so objekti postali funkcionalno vse bolj sposobni. Tiste, ki jih opredeljujejo lastnosti, ki so imitacija človeškega obnašanja, imenujemo agenti. Ti se pri realizaciji internih izračunov poleg lastnih metod poslužujejo še metod preostalih objektov, interne izračune pa vršijo na osnovi prepričanj, sposobnosti in izbir ter komunicirajo med sabo z visokonivojskim jezikom, ki ni neposredno povezan z izračuni.

Definicij agentov je več, ena najpogosteje uporabljenih in najbolj generičnih pa je podana v [Gri01]. Ta pravi, da so agenti programske rešitve, ki izkazujejo naslednje lastnosti:

- a) Avtonomnost – agent samodejno začne dejavnosti skladno s svojimi cilji, ima lasten nadzor in lahko deluje v imenu uporabnika ter je v znatni meri neodvisen od sporočil preostalih agentov.
- b) Prilagodljivost – agent je sposoben prilagajati svoje obnašanje, bodisi na podlagi lastnega učenja, prilagajanja uporabnika ali pa z naknadnim nalaganjem ustreznih kode iz okolja.

- c) Mobilnost – agent se je sposoben premestiti iz enega okolja v drugo, naj si bo s prenosom same kode in zagonom na novi lokaciji ali pa s t.i. serializacijo, kjer se preneseta koda in stanje agenta in se nadaljuje izvajanje v novem okolju.
- d) Inteligenca – agent je sposoben razmišljati o svojih ciljih, pridobljenih informacijah in ostalih agentih ter uporabnikih.
- e) Sposobnost sodelovanja – agent zna komunicirati in sodelovati z drugimi agenti v okolju pri realizaciji določenega cilja, pa naj si bo to v dinamičnih ali pa statičnih okoljih.
- f) Obstojnost (persistenca) – agenti so ob podpori infrastrukture sposobni ohranjati stanje in znanje na daljši rok.

Pred nadaljnjo obravnavo podajmo definicijo združbe agentov. Združba agentov je definirana kot skupek agentov, kjer posamezni agenti nimajo vseh potrebnih sposobnosti za rešitev problema, kjer ni nekega centraliziranega nadzora, kjer so podatki decentralizirani in kjer je računanje asinhrono [Flo99].

Iz definicije lahko neposredno ugotovimo, da je internet naravno okolje za združbe agentov, ki so opredeljene z naslednjimi lastnostmi:

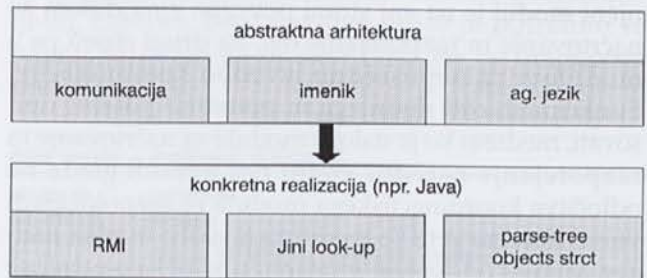
- Arhitekturo agentov kot takih, izhajajoč iz zgornje definicije, od katerih so eni agenti neposredno opravilni, drugi pa infrastrukturne narave.
- Arhitekturo združbe same, ki zahteva tudi definicijo sledečih elementov:
 - Jezika, ki predstavlja bistvo komunikacije med agenti in je deklarativne, to je pojasnjevalne narave (definicije, predpostavke), kjer sta najpogostejša predstavnik Knowledge Query and Manipulation Language [Fin97] in FIPA ACL [FIP01b].
 - Komunikacijskih protokolov, ki opredeljujejo načine izmenjave sporočil v omenjenem jeziku in format njihove predstavitve, kar je v najpreprostejši obliki lahko izvedba prek vtičnic in sklada TCP/IP ali pa elektronske pošte.
 - Ontologij, ki pomenijo način soglašanja o pomenu konceptov. Definirane so kot sheme za opis konceptov in njihovih relacij v nekem komunikacijskem kontekstu. Običajno pri tem izhajamo iz predikatnega računa prvega reda. Primer ontologije je Ontolingua [Gru93].
 - Izvajanja registracije imen in storitev, s čimer je omogočena identifikacija obstoječih agentov in vzpostavitev komunikacije z njimi ob poznavanju lastnosti (sposobnosti) le-teh.

Ena najbolj uveljavljenih skupin standardov, ki pokriva do sedaj omenjena področja za zagotovitev interoperabilnosti heterogenih agentov, je tista, ki jo sprejema FIPA (The Foundation for Intelligent Phys-

icall Agents). Specifikacije FIPA zajemajo abstraktno arhitekturo, komunikacijo agentov, upravljanje agentov, transport sporočil in pa aplikacije.

Abstraktna arhitektura

Spodnja slika prikazuje abstraktno arhitekturo tehnologij agentov.

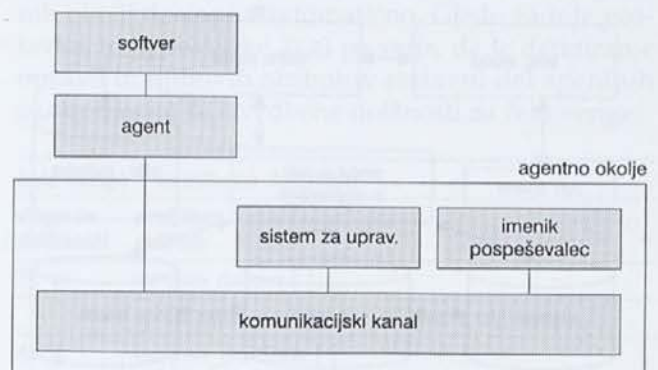


Slika 1: Abstraktna arhitektura tehnologij agentov.

Abstraktna arhitektura se sestoji iz agentov, opredelitve komunikacije, imenika in jezika, ki ga agenti uporabljajo za komunikacijo. Na sliki vidimo primer realizacije v javanskem okolju, kjer komunikacija poteka prek javi lastnega protokola RMI, za imenik pa služi storitev Jini look-up.

Agentno okolje

Drug zorni kot na tehnologije agentov nudi arhitektura OUA, ki je podana na spodnji sliki. Agent komunicira z aplikacijskim softverom ali pa neposredno z uporabnikom. Prek komunikacijskega kanala je povezan s sklopom za upravljanje in pa dvema vrstama infrastrukturnih agentov. Prva opravlja funkcijo klasičnega imenika, ki omogoča identifikacijo agentov in njihovo lokacijo, druga pa hrani podatke o funkcionalnosti (lastnostih, sposobnostih) posameznih agentov.



Slika 2: Okolje za upravljanje agentov.

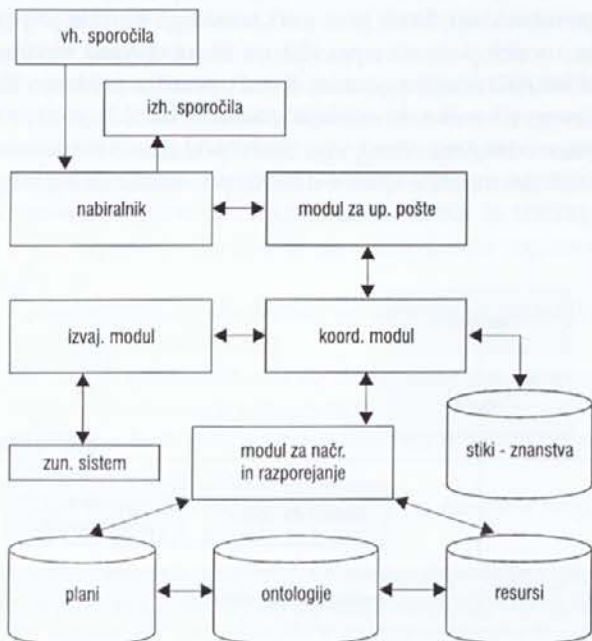
Poleg teh ključnih komponent OUA vsebujejo še knjižnice z ustreznimi algoritmi, protokoli in podatkovnimi strukturami ter orodja za prikazovanje (npr. raznih statističnih veličin).

Predstavitve tipične arhitekture samega agenta povzemamo po [Col99]. V osrčju agenta je koordinacijski modul. Ta sprejema odločitve glede ciljev agenta, npr. kdaj neki cilj realizirati ali opustiti, in pa koordinira komunikacijo s preostalimi agenti. Omenjeni modul je na eni strani povezan z modulom za načrtovanje in razporejanje del, na drugi strani pa z modulom za neposredno izvedbo komunikacije. Funkcionalnosti slednjega ni potrebno posebej opisovati, medtem ko je naloga modula za načrtovanje in razporejanje narediti vrstni red opravil glede na odločitve koordinacijskega modula in razpoložljivih virov. Končno je tu še modul za izvedbo in njen nadzor. Moduli pri svojem izvajanju uporabljajo ustrezne notranje in zunanje zbirke podatkov, kot je to razvidno s slike 3. Med osrednjimi je zbirka ontologij.

Modeliranje poslovnih procesov

V tem razdelku bomo podali primer uporabe tehnologij agentov za reševanje problemov na področju poslovanja. Konkretno bo to modeliranje obvladovanja verige dobav.

Ko s pomočjo tehnologij agentov rešujemo določen problem, je pristop analogen pristopom, ki jih poznamo s področja systemske analize in načrtovanja. Po analizi začnemo z izdelavo konceptualnega modela, ki ga vse bolj razgrajujemo in formaliziramo,



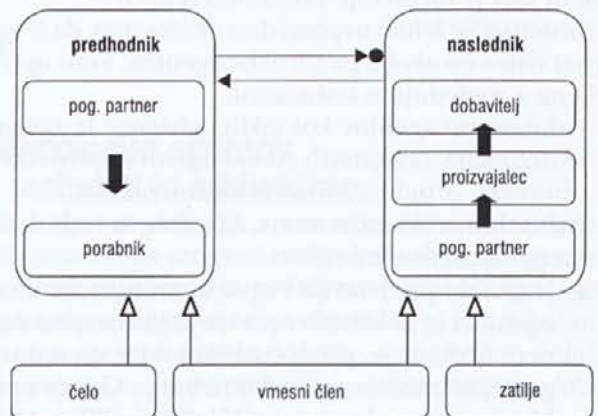
Slika 3: Primer arhitekture agenta.

tako da dobimo logični model. Tega kodiramo v ustreznem programskem jeziku in ga končno preslikamo v fizični / izvedbeni model. Torej:

- V določeni domeni (poslovnem okolju) opravimo analizo poslovnega procesa, kjer opredelimo probleme, želene rešitve ter koncept vlog udeležencev.
- V drugem koraku identificiramo potrebne agente, storitve in ontologije, kar predstavlja dizajniranje.
- Sledi realizacija v okviru agentnega okolja, kjer kreiramo agente, jih ustrezno organiziramo in koordiniramo.

Modeliranje vlog

Za naš primer bomo vzeli enostaven primer verige dobav pri izdelavi pametnih kartic. Dobavitelja v verigi sta izdelovalec mikroprocesorjev (IMP) in dobavitelj plastičnih kartic (DPK), ki dostavljata komponente izdelovalcu pametnih kartic (IPK). Slednji vstavi mikroprocesor s pomočjo epoksi rezine v ustrezen utor na kartici in opravi posebitev le-te.



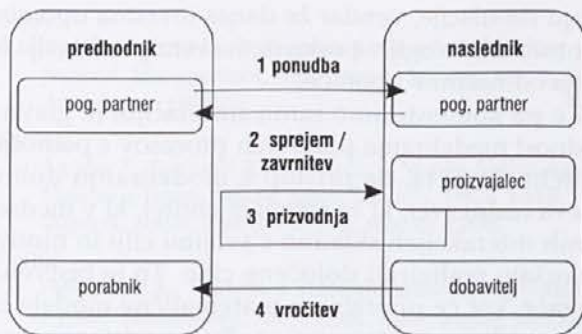
LEGENDA:

- interakcija → vsebovanje
- večkratnost ● dedovanje

Slika 4:

Vloge v verigi dobav (veriga je gledana s strani končnega porabnika).

Model vsebuje dve vloge: predhodnik in naslednik. Modelirane vloge prikazemo z diagramom, ki izhaja iz jezika UML (Universal Modelling Language [Alh98]). Zgornja slika je osnovni gradnik poljubne verige dobav, ki prikazuje ključne vloge in relacije. Gledana je s strani končnega uporabnika, ki je torej v interakciji s čelom verige. Rečeno drugače, čelo verige se imenuje tista točka, ki je v interakciji s končnim uporabnikom. Na spodnji sliki je tako prikazan poteka interakcije s končnim porabnikom.



Slika 5: Potek interakcije.

Zgornji potek interakcije predstavlja prav tako osnovni vzorec vzdolž verige dobav. Na tej osnovi lahko podrobno definiramo vsako vlogo, kar je nujno pred dokončno implementacijo. Definicija vsake vloge poleg splošnega opisa zahteva opredelitev relacij le-te do ostalih vlog, odgovornosti, zunanje vmesnike in predpogoje. Očitno v našem primeru IPK nastopa na čelu verige, vmesnih členov nimamo, zatilje pa predstavljata IMP in DPK. Naš model vlog podaja spodnja tabela:

agenti	vloge
IPK	čelo verige – začetnik pogajanj, porabnik
IMP	zatilje verige – pogajalski partner, dobavitelj, proizvajalec
DPK	zatilje verige – pogajalski partner, dobavitelj, proizvajalec

V naslednjem koraku je potrebno doreči, kako bodo agenti realizirali zadane vloge. Vsaka vloga potegne za sabo določene dolžnosti, ki so po naravi družbeno (socialno) in neposredno (izvedbeno) naravnane. Navedimo obe skupini odgovornosti najprej za udeleženca, ki je na čelu verige.

čelo verige – družbene dolžnosti

vloge	dolžnosti
1) začetnik pogajanj	a) Agent mora poznati in se zavedati lastnosti / sposobnosti preostalih agentov. b) Agent mora znati izvesti povpraševanje za dobavo materiala. c) Agent se mora znati pogajati glede cene materiala. d) Agent mora znati posredovati informacije o naročilih lastni vlogi porabnika.
2) porabnik	a) Agent mora znati prejeti material.

čelo verige – izvedbene dolžnosti

vloge	dolžnosti
3) začetnik pogajanj	a) Agent mora podati zahtevo za nov material.
4) porabnik	a) Agent mora obdelati material.

Podobno opredelimo dolžnosti za zatilje verige. Da pa ne bi preseгли okvirov tega članka, smo primer napravili samo za čelo verige.

Snovanje aplikacije

S konceptualnega nivoja se v tej fazi selimo proti logičnemu nivoju. Dolžnosti, povezane s posameznimi vlogami, moramo preslikati na način, kot jih zahteva primer, ki ga proučujemo in OUA, v okviru katerega bo potekala realizacija. Rečeno poenostavljeno, človeku predstavljive opise je potrebno še naprej razgrajevati in sicer tako, da bo omogočen prenos v OUA.

čelo verige – družbene dolžnosti

vloge in dolžnosti	problemi rešitve
1) a)	znanje [dobavitelj, zahtevan_material] pridobivanje podatkov o zmogljivostih ostalih agentov
1) b)	začetek_dialoga [dobavitelj, zahtevan_material] omogočiti agentu uporabo ustreznega koordinacijskega protokola
1) c)	vrednotenje [material, cena] omogočiti agentu uporabo pogajalskih strategij
1) d)	pošiljanje_sporočila [vloga_porabnika, podrobnosti_dobave] omogočiti agentu pošiljanje sporočil
2) a)	prevzem_materiala [] definiranje izvedbe opravila in njegovih atributov

Komentirajmo najprej postavko 1) a). Da agent pridobi ustrezno znanje, je moč definirati statične povezave pri generiranju agentov, lahko pa agenti uporabijo infrastrukturne možnosti agentnega okolja in sami vzpostavijo ustrezne povezave. Glede postavke 1) b) – agentu moramo omogočiti dostop do ustreznega protokola za začetek in izvrševanje dialoga, npr. Contract Net [Smi80]. Podobno velja glede pogajalskega protokola, kjer lahko uporabimo npr. protokol GrowthStrategy [Col99]. Predzadnja postavka (to je postavka 1) d), pošiljanje sporočila) je v večini agentnih okolij dostopna avtomatično. Glede zadnje postavke, to je postavke 2) a) pa velja, da je definiranje opravil in njihovih atributov sestavni del agentnih okolij. Sledijo še izvedbene dolžnosti za čelo verige:

čelo verige – izvedbene dolžnosti

vloge in dolžnosti	problemi rešitve
3) a)	zahteva_dobave [] sproži zahtevo za dobavo skladno s cilji agenta
4) a)	izdelava_proizvoda [] porabi material in naredi proizvode skladno s cilji agenta

V naslednjem koraku moramo pri pretvarjanju v logični model modelirati koncepte dane domene (ontologije), kar izvedemo prek poimenovanja vsakega koncepta, pripadajočih atributov in ustreznih omejitvev. V našem primeru obvladovanja verige dobav je ontologija naslednja:

koncepti	atributi (tipi)	omejitve
procesor	proizvajalec (alfanumeric)	[intel motorola]
	model (alfanumeric)	
plastična_kartica	material (character)	[ABS PETP]
	standard (alfanumeric)	

V zgornji tabeli atribut proizvajalec označuje proizvajalca mikroprocesorjev, atribut model konkreten model mikroprocesorjev, material označuje vrsto plastike za izdelavo pametnih kartic, standard pa skladnost kartic z določenimi standardi.

Realizacija aplikacije

Na tej točki lahko pristopimo k realizaciji aplikacije. Problem je obdelan do nivoja, ki omogoča neposredno uporabo algoritmov. Teh večinoma ni potrebno pisati samemu razvijalcu, ker so vgrajeni v OUA. Z ustreznimi orodji v okviru konkretnega OUA naredimo naslednje:

- kreiramo ontologijo;
- kreiramo osnovne agente, opis opravil ter organiziramo in koordiniramo agente;
- kreiramo infrastrukturne agente (agenta, ki opravlja funkcijo imenika in agenta, ki vodi evidenco o sposobnostih preostalih agentov).

Ta opravila opravimo kar prek ustreznih grafičnih vmesnikov, podobno kot to počnemo pri programiranju zbirk podatkov (npr. v MS Access).

Sedaj poženemo aplikacijo, s čimer združba agentov "oživi". Dogajanja tekom življenjskega ciklusa združbe agentov lahko opazujemo s pomočjo ustreznega orodja, t.i. vizualizatorja, ki je pravzaprav specializiran agent. Predmet opazovanja je lahko posamezen agent, nadalje dinamika vzpostavljanja odnosov med agenti v realnem času, kaj se dogaja v verigi dobav s surovinami itd. Vizualizator skratka omogoča opazovanje in zajem raznih statistik.

Zaključek

V pričujočem prispevku smo se osredotočili na možnost realizacije poslovnih procesov s pomočjo tehnologij agentov. Resnici na ljubo smo ostali na

nivoju simulacije, vendar že danes tovrstna uporaba agentnih tehnologij v poslovnem svetu predstavlja le enega od načinov uporabe.

Ce pa komentiramo samo simulacijo, je glavna prednost modeliranja poslovnih procesov s pomočjo teh tehnologij ta, da pristop k modeliranju dobro odseva realni svet, ki se sestoji iz entitet, ki v medsebojnih interakcijah skladno s svojimi cilji in motivi poskušajo realizirati določene cilje. To je bistveno drugače, kot če uporabimo matematične modele in proces podamo z ustreznimi enačbami, odstopanja pa potem naknadno vnašamo npr. s stohastičnimi elementi. V primeru tehnologij agentov bi lahko rekli, da imamo opravka z reševanjem od dna k vrhu, v klasičnem primeru pa z reševanjem od vrha k dnu. S spremembami poljubnih karakteristik vzpostavljenega sistema lahko opazujemo učinke in poiščemo ustrezne rešitve za naše poslovno okolje. S tovrstnimi orodji lahko razvijemo realen občutek za upravljanje proizvodnih in storitvenih sistemov, kar je s klasičnimi analitičnimi pristopi težje.

Je pa seveda na področju tehnologij agentov še precej odprtih vprašanj. Eno glavnih je vprašanje varnosti. Namreč agenti, še zlasti mobilni, so izpostavljeni novim vrstam napadov (tako metode kot njihovi podatki). Do sedaj je bil koncept varovanja v omrežju tak, da smo morali ščititi kodo, vezano na nek procesor. Ščitili smo "matično kodo", to je tisto kodo, ki je imela domicil v določenem strojnem okolju. Pri mobilnih agentih ni več moč govoriti o matičnem procesorju in jih je pravzaprav potrebno ščititi pred okoljem, kjer gostujejo. Klasična kriptografija v primeru mobilnih agentov odpove. Precej je obetal pristop s t.i. mobilno kriptografijo [San98], vendar pa kaj več od grobe teoretične osnove na tem področju ni bilo narejenega. Bi bila pa to generična rešitev za probleme, povezane z varnostjo agentov.

Ne glede na to, da so trenutne rešitve še okorne in je marsikaj nedorečenega (zlasti na področju varnosti), so tehnologije agentov obetavno področje s široko namembnostjo uporabe. V bližnji bodočnosti bodo verjetno botrovale tudi novim poslovnim modelom in ne samo izboljševanju obstoječih.

Literatura

[Gri01]

Griss L.M., Accelerating Development with Agent Components, Computer, vol. 34, št. 5, IEEE, maj 2001, str. 37-43.

[Tec98]

Tecuci G., Building Intelligent Agents, Academic Press, New York 1998.

[Gue99] Guessoum Z., Briot J.P., From Active Objects to

Autonomous Agents, IEEE Concurrency, vol. 7, št. 3, marec 1999, str. 68-81.

- [Bie98] Bieszcad A., White T., Pagurek B., Mobile Agents for Network Management, IEEE Comm. Surveys, vol. 1, št. 1, 4th quarter 1998.
- [San98] Sander T., Tschudin C.F., Protecting Mobile Agents Against Malicious Hosts, Mobile Agent Security, LNCS 1419, Vigna G. editor, Springer Verlag, februar 1998.
- [Flo99] Florez M.R.A., Towards a Standardisation of Multi-Agent System Frameworks, Crossroads, ACM, avgust 1999.
- [Fin97] Finin T., Laboru Y., Mayfield J., KQML as an Agent Communication Language, Software Agents, AAAI Press, Menlo Park 1997.
- [Gru93] Gruber T.R., A Translation Approach to Portable Ontology Specifications, Proc. Of KAW '93, Banf 1993, str. 199-200.
- [FIPO1a] Makelainen M., CRUMPET – Tourism and Open Source for Small Devices, Inform!, FIPA, vol. 2, št. 1, marec 2001.
- [FIPO1b] FIPA, ACL Message Structure Specification, FIPA XC00016E, Geneva, avgust 2001.
- [Col99] Collis J., Ndumu D., The Zeus Agent Building Toolkit, Parts 1 thru 4, BT, november 1999.
- [Alh98] Alhir S.S., UML in a Nutshell, O' Reilly, 1998.
- [Smi80] Smith G.R., The contract net protocol: High-level communication and control in distributed problem solver, IEEE Transactions on Computers, Vol. 29, št. 12, december 1980, str.1104—1113.

◆
Denis Trček se ukvarja s področjem e-poslovanja s poudarkom na varnosti že dobrih deset let. S tega področja ima prek šestdeset objav, v glavnem v tujini, reference pa vključujejo tudi sodelovanje na aplikativnih projektih doma: od vzpostavitve informacijske infrastrukture Narodne galerije v Ljubljani do arhitekture varovanja komunikacij pri projektu Kartice zdravstvenega zavarovanja ZZZS. Avtor je docent za področje informatike, je zaposlen na Institutu "Jožef Stefan" in je sodelavec Fakultete za računalništvo in informatiko Univerze v Ljubljani.
 ◆