Mi egy ágens kommunikációs nyelv?

Kommunikáció eszköze (mint embereknél)
– független, autonóm egyedek között
Funkció: információ átadás, cselekvés-koordinálás, szociális manipulálás, jeladás.

Ágens programozásának eszköze
– lehetővé tenni az IT rendszer egyedeinek, hogy érjék el a céljaikat, pl. kapcsolatot létesítsenek más ágensekkel

Software engineering eszköze
– lehetővé tenni IT mérnököknek, hogy érjék el a céljaikat, pl. az ágensei kapcsolatot létesítsenek más ágensekkel

Formális nyelvek
– Definiált szintaxis
– Definiált szemantika
– Tulajdonságainak megértése az üzemeltetésük megelőzően.

Agent Communication Languages (ACLs)

Két fő javaslat az ACL-re:

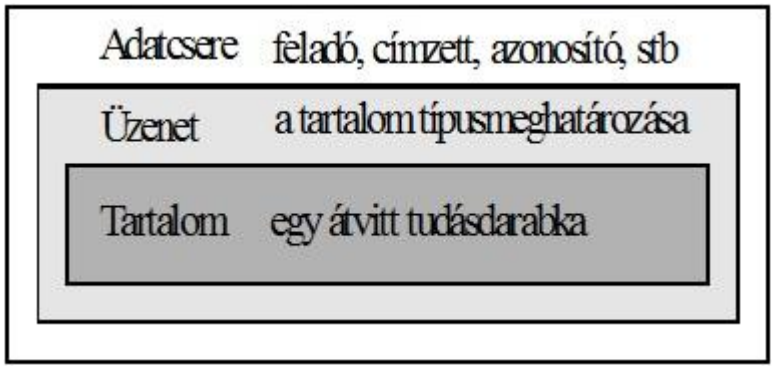
USA DARPA's
Knowledge Query and Manipulation Language (KQML)
(több tudásbázis fuzionálása)

Foundation for Intelligent Physical Agents ACL (FIPA ACL)
(kifejezetten ágensek közötti kommunikáció)

Mindkettő: a kommunikáció (üzenetek) 2 szintje:
– a párbeszéd témája – valamilyen logikai nyelvben
– a témákra vonatkozó illokuciók

Pl.
- query (Esik az eső) „Esik az eső?"
- inform (Esik az eső) „Esik az eső!"

KQML Knowledge Query and Manipulation Language



A kommunikáció szintjei

Adat csere szint (kommunikáció mechanizmusa):

állomások között közlekedő üzenet csomagok legkülső rétege, a kommunikáció legalsó protokoll szintje:

üzenet feladója, *címzettje*, *azonosítója*, … egyéb *kommunikációs paraméterek*

Üzenet szint (kommunikáció logikája):

az üzenet *tartalmának* azonosítása, *típus-meghatározása*

Tartalom szint (kommunikáció tartalma):

Az *átadott információ k*özvetítése. Bármit tartalmazhat, amiben a kommunikáló ágensek megegyeznek.

Példa: A küldi B-nak B küldi A-nak

```
(standby
   :language KQML
   :ontology K10
   :reply-with g1
   :content
   (stream-about   :language KIF
                   :ontology motors
                   :reply-with q3
                   :content motor1))
```

```
(ready  :reply-with 2F0B
        :in-reply-to g1)
```

```
(next  :in-reply-to 2F0B)
```

```
(tell   :language KIF
        :ontology motors
        :in-reply-to q3
        :content
        (= (val (torque motor1)
                (sim-time 5))
                (scalar 12 kgf))
```

…

```
(discard  :in-reply-to 2F0B)
```

Példa: A küldi B-nak B küldi A-nak

```
(advertise
   :language KQML
   :ontology K10
   :content
   (subscribe
      :language KQML
      :ontology K10
      :content
         (stream-about  :language KIF :ontology motors :content motor1)))
```

```
                                    (subscribe  :reply-with s1
                                                :language KQML :ontology K10
                                                :content
                                                     (stream-about  :language KIF
                                                                    :ontology motors
                                                                     :content motor1))
```

```
(tell  :language KIF
        :ontology motors
        :in-reply-to s1
       :content  (= (val (torque motor1) (sim-time 5))
                          (scalar 12 kgf))
……
```

Szemantikailag (jobban) specifikált KQML (1997)

performatívum

⇑ ⇑

min. elő-feltételhalmaz — min. utó-feltételhalmaz

sikeresen elvégzett performatívum által előidézett végállapot

Advertise (A, B, M)

1. A azt állítja B-nek, hogy az M üzenetét elfogadja (ha megkapja) és processzálja (elkötelezettség)
2. Int (A, Proc (A, M)) M = performatívum-név (B, A, X)
3. Pre(A): Int (A, Proc (A, M))
   Pre(B): -
4. Post(A): Know (A, Know (B, Int (A, Proc (A, M))))
   Post(B): Know (B, Int (A, Proc (A, M)))
5. Compl: Know (B, Int (A, Proc (A, M)))

FIPA – Foundation of Intelligent Physical Agents

Indulás – 1995/6, több tíz európói telecom cég, egyetem, ...
2002 (félig-meddig) szabvány
2005-től része IEEE Computer Society, IEEE CS Standard Group on …
IEEE FIPA Standard Committee www.fipa.org

Szabvány kérdése - “gyors szabvány”
- semmi megkötés ágensek belsejére, de
- megkötés a közösség építésére
  alapvető közösségi struktúra szervezet = közösség
  (belépés, kilépés, normatívák, specifikált viselkedések, …)
  alapvető kommunikáció

FIPA konzisztencia – HA alkalmaznánk, akkor előírás szerint viselkedjen.
FIPA szabvány - normatív (formális, formálisan verifikálható modellek)
- illusztratív (leíró informális modellek, alkalmazások)

Szabvány által lefedett témák:

Abstract Architecture

Agent/ Software Integration (örökölt rendszerek)

Agent Message Transport

Transport Protocols, Envelope Representations .... String, XML, Efficient Binary, ...

ACL Representations - ACL Message Structure

Agent Management (platform, kötelező ágensek)

Agent Management Support For Mobility

Agent Communication Languages

Interaction Protocols - Interaction Protocol Library Specification (AUML)

Communicative Acts Library (beszéd aktusok)

Content Languages - KIF (Knowledge Interchange Format)

- CCL (Constraint Choice Language)
- SL (Semantic Language)
- RDF (Resource Description Framework)
- Content Language Library Specification

Device Ontology Specification

Message Buffering Service Specification

Messaging Interoperability Service Specification

Ontology Service Specification

Reference Applications: Personel Travel Assistance, Nomadic Application Support, ...

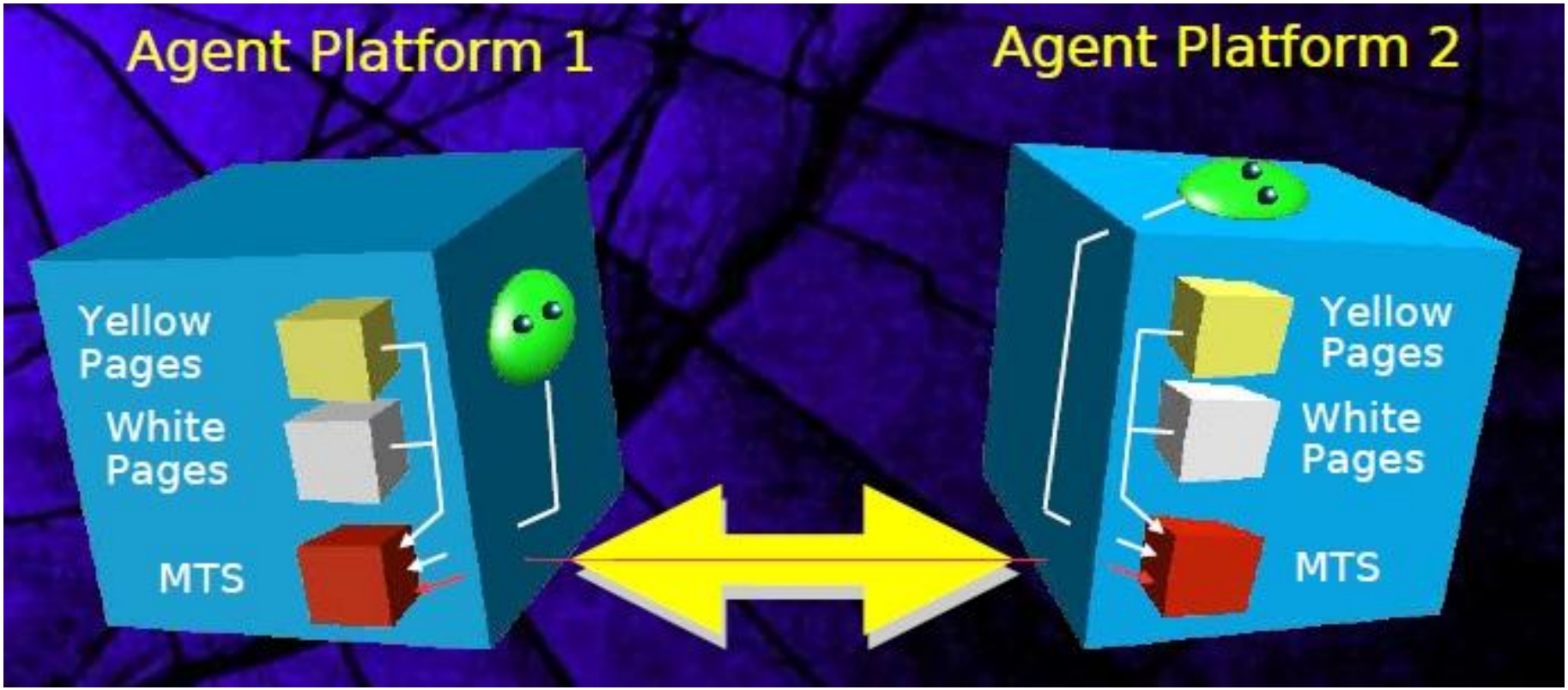
Agent Management

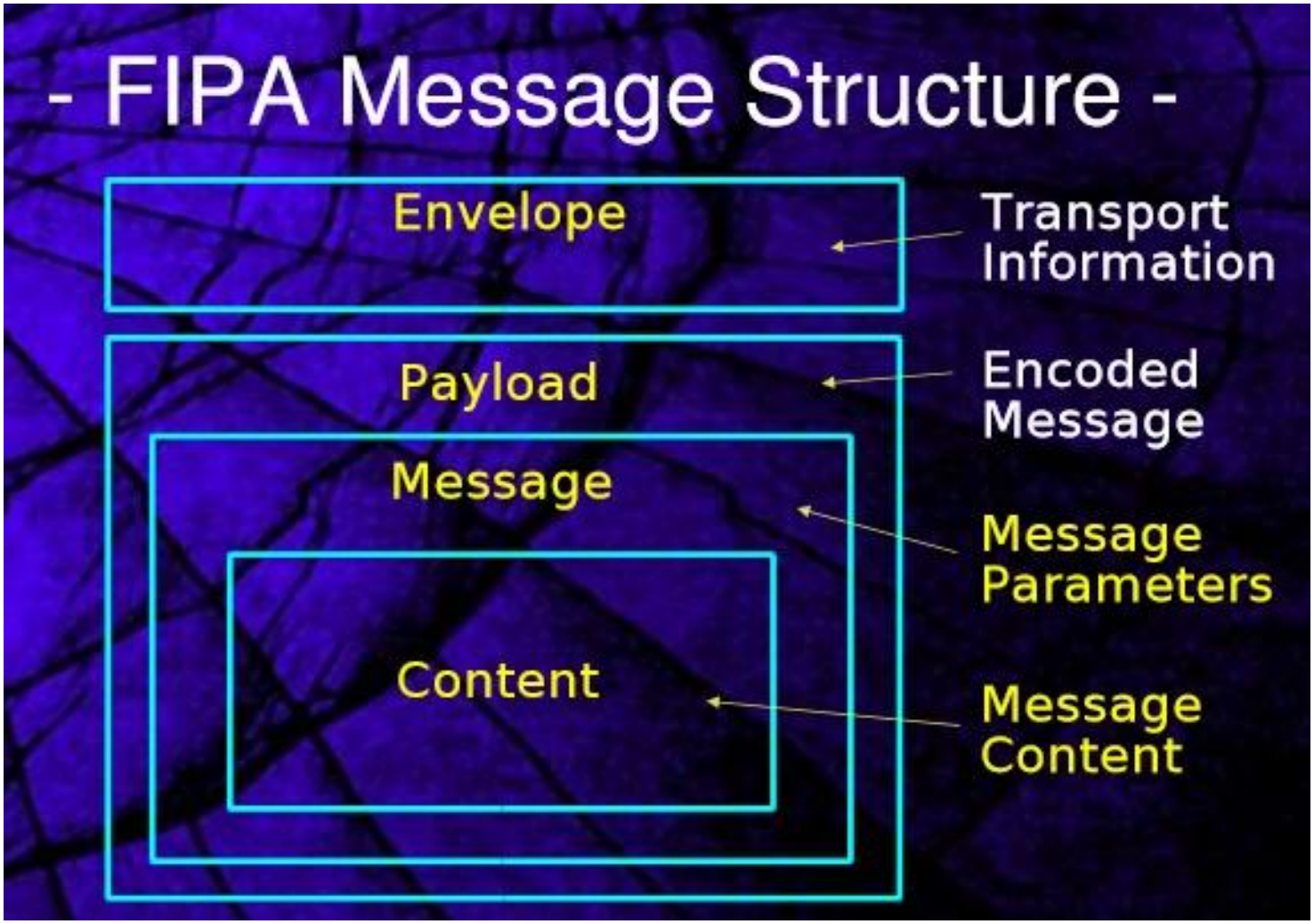
Közösségdefiníció = **Agent Platform** – fizikai infrastruktúra
(1 v. több hoszton)

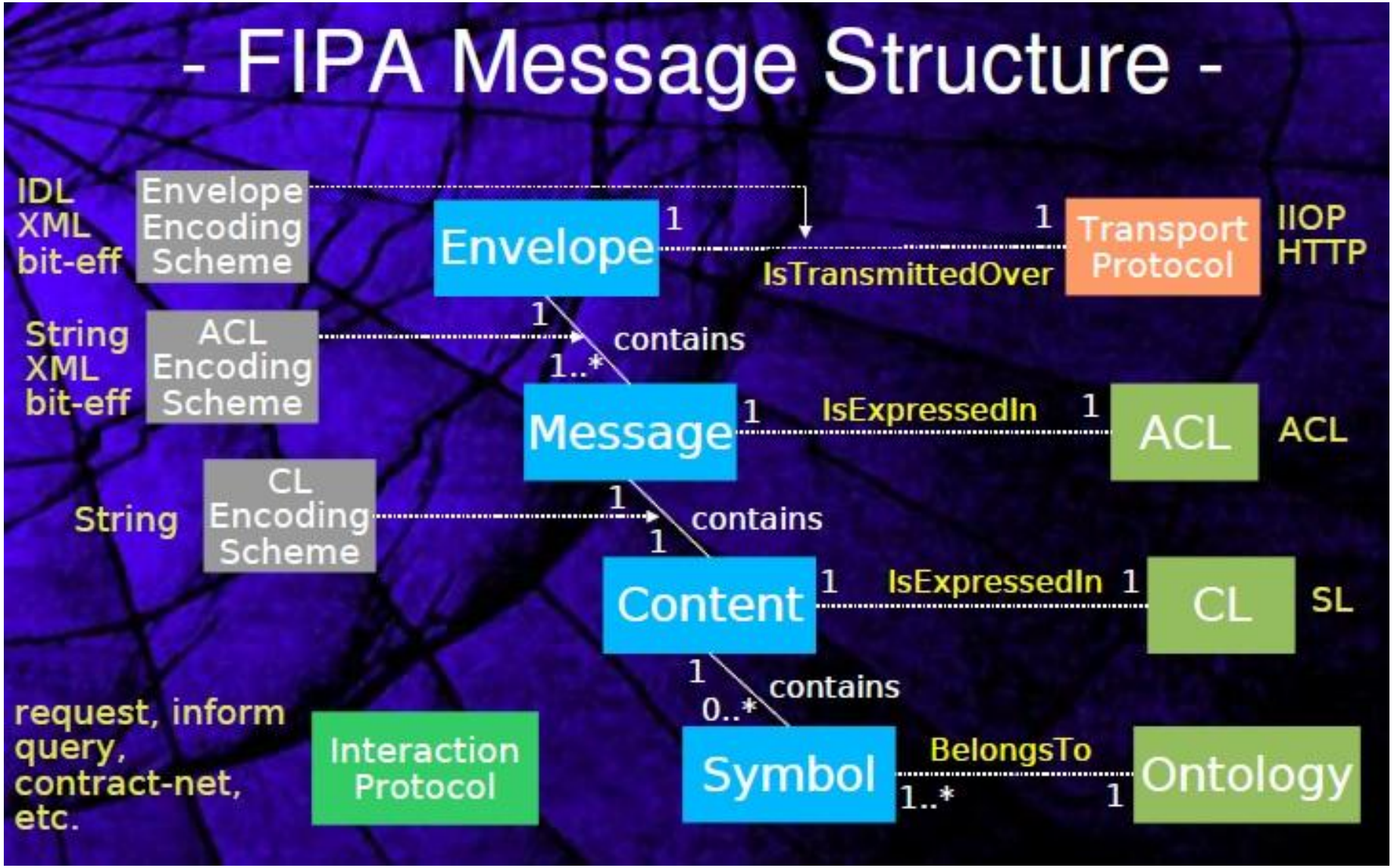
- belépés/ kilépés
- találkozás (hirdetés, erőforrás lokálizálás)
- közösség belüli kommunikáció

DF: Directory Facilitator, 1 vagy több (DF Federation)
AMS: Agent Management System, csak 1/ platform
MTS: Message Transport Service

DF: register, deregister, modify, search
(ágens nem köteles bejelentkezni)
AMS: register, deregister, modify, search, get-description
(ágens köteles bejelentkezni)


Agent Platform 1
Agent Platform 2
Yellow Pages
White Pages
MTS
Yellow Pages
White Pages
MTS

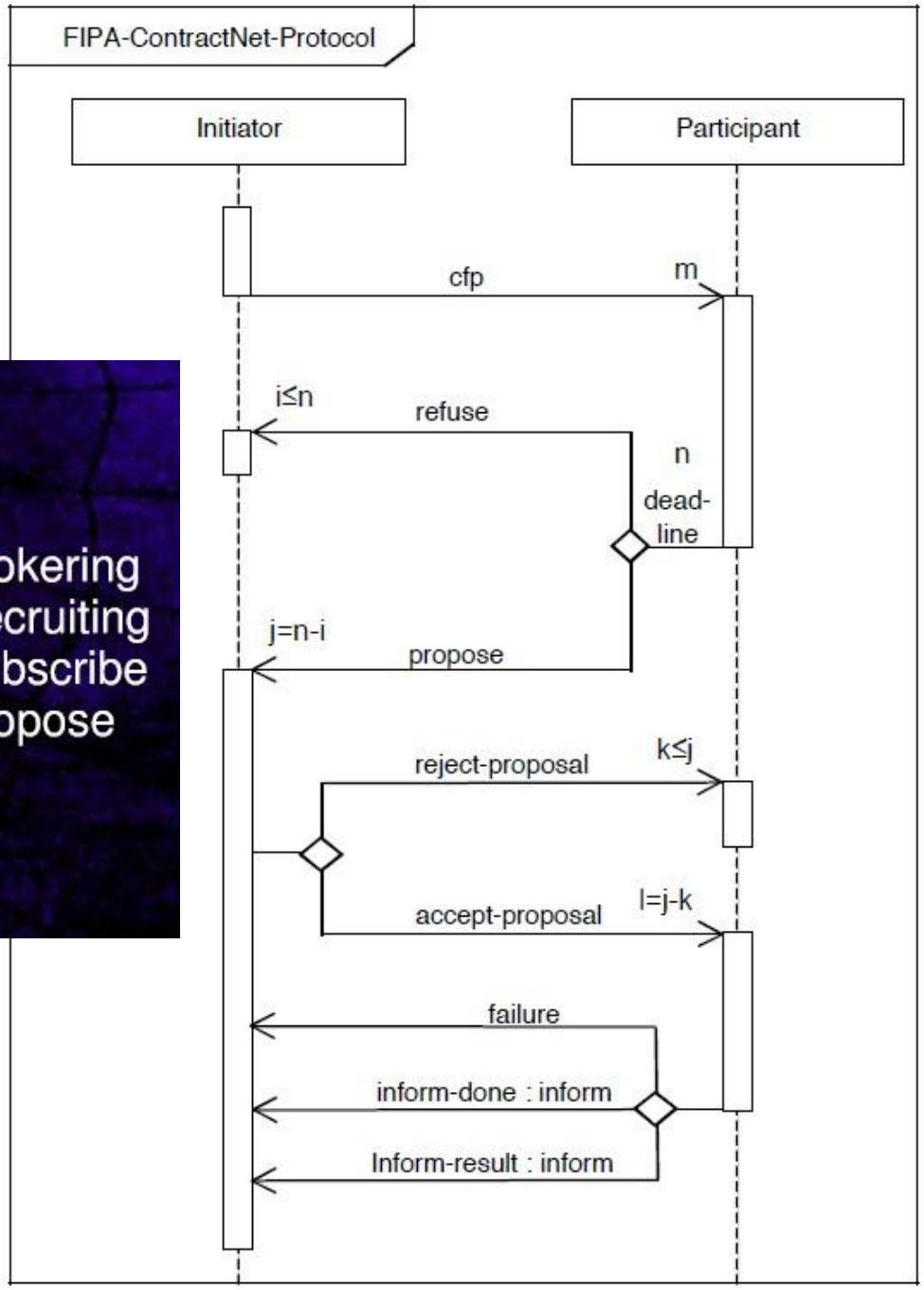

- FIPA Message Structure -
Envelope
Payload
Message
Content
Transport Information
Encoded Message
Message Parameters
Message Content


- FIPA Message Structure -
IDL
XML
bit-eff
Envelope Encoding Scheme
Envelope
1
1
Transport Protocol
IIOP
HTTP
IsTransmittedOver
String
XML
bit-eff
ACL Encoding Scheme
1
contains
1..*
Message
1
IsExpressedIn
1
ACL
ACL
String
CL Encoding Scheme
1
contains
1
Content
1
IsExpressedIn
1
CL
SL
1
contains
0..*
request, inform
query,
contract-net,
etc.
Interaction Protocol
Symbol
BelongsTo
1..*
1
Ontology

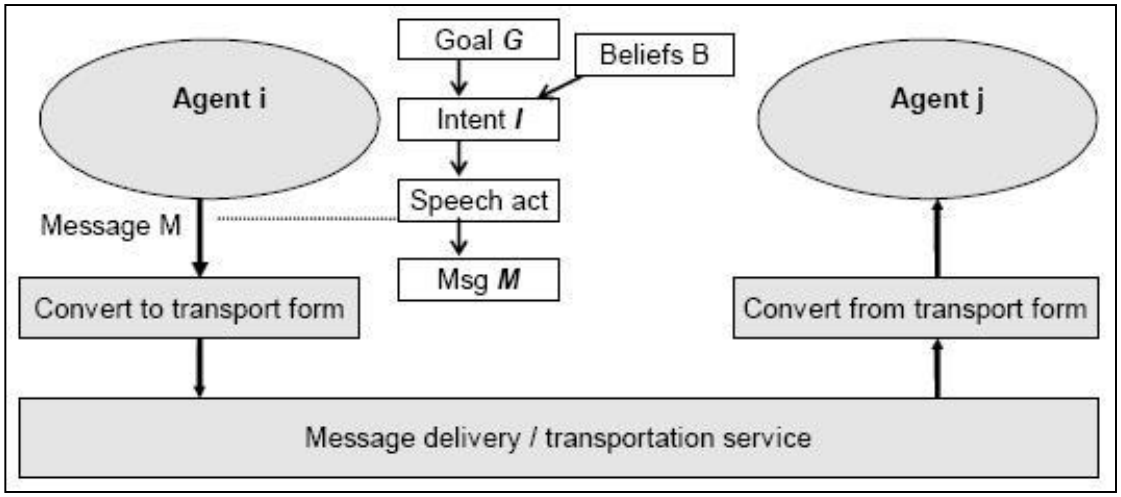
- Interaction Protocols -

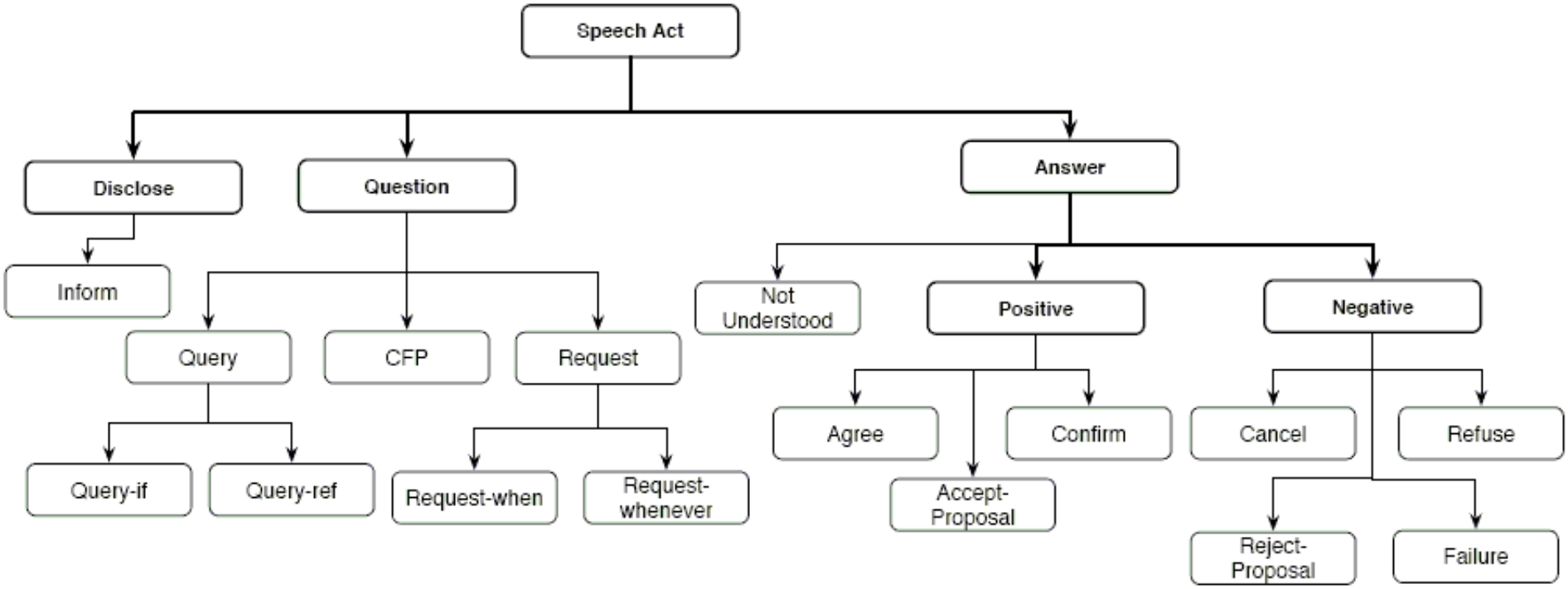
FIPA defined IPs are:

- FIPA-Request
- FIPA-Query
- FIPA-Request-When
- FIPA-Contract-Net
- FIPA-Iterated-Contract-Net
- FIPA-Auction-English
- FIPA-Auction-Dutch
- FIPA-Brokering
- FIPA-Recruiting
- FIPA-Subscribe
- FIPA-Propose



Communicative Act Library





Semantic Language SL

(not) (and) (or) (implies) (equiv) (forall) (exists)

B <agent> <expr>
U <agent> <expr>
I <agent> <expr> szándéka van és tervez
PG <agent> <expr> tartós célja van, de nem szükségk. tervez

(feasible <actexpr> <Wff>) igaz, hogy cselekvés megtörténthet közvetlenül utana Wff igaz lesz
(done <actexpr> <Wff>) igaz, hogy cselekvés épp megtörtént és előtte Wff igaz volt
(iota x (P x)) pont olyan x, amire igaz P(x)
(any <term> <formula>) akármilyen objektum, ami a formulát teljesíti
(all <term> <formula>) minden olyan objektum, ami …

$$\text{Bifi } \phi = \text{Bi } \phi \vee \text{Bi } \neg\phi, \qquad \text{Abnij } \phi = \text{Bi Bj Bi ..... } \phi$$

Confirm <i, confirm (j, ϕ)>
FP: Bi $\phi \wedge$ Bi Uj ϕ
RE: Bj ϕ

FIPA ACL problémái

Implicit feltételezések:

- Ágensek örökké kapcsolatban vannak egymással
- Ágensek hivatalból őszinték, jó indulatúak, igazmondók.

Beszédaktus hiányosságok

- Párbeszédhez való csatlakozás, párbeszéd „elhagyása”?
- Részvétel engedélyezése, tiltása?
- Hiedelemkonfliktusok kezelése?
- Mi van, ha az őszinteség nem vállalható fel (pl. tárgyalások)
- Állítások megkérdőjelezése?
- Dialektikus kötelezettség (aki állít, nem köteles előállni annak indoklásával)
- Explicit érvelési lokuciók hiánya.
- Bármely lokuciók követhetik egymást. Bomlasztó viselkedés nehezen hárítható el.
- Önalakítás kifejezése – hogyan fejezzük ki a hiedelmeink megváltozását hogyan ismerhető fel, hogy ez nem hibás, rosszindulatú, … viselkedés?

- Magán axiomatikus szemantika (nyílt szervezetben!) nem verifikálható.

– Egy nyílt többágenses rendszerben egy ágens belső állapotai mások számára általában nem hozzáférhetőek.

- Egy okos ágens mindig szimulálhatja, hogy egy szükséges belső állapottal rendelkezik.

- Egy okos ágens vállalhat bizonyos hiedelmeket csakis a lokució kimondásának idejére (és akkor (pillanatnyilag) őszinte).

ACL területén mire való egy szemantika?

1. A kimondások, kimondássorozatok, párbeszédek közös jelentésének a biztosítása a kommunikációban részt vevő szoftver ágensek számára.
2. Közös jelentés biztosítása az ágensek emberi felelősei számára (ha ilyenek vannak).
3. Közös jelentés biztosítása az ágenseket és a protokollokat tervező emberi tervezők számára.
4. Közös jelentés biztosítása más emberi érintettek számára (pl. szabályozók).
5. Ágens nyelvek és protokollok tulajdonságainak precíz tanulmányozása, választ keresve pl.
   - Terminálódik-e mindig egy legális párbeszéd?
   - Biztosítható-e a sikeres terminálódás?
   - Mik a terminált párbeszéd eredményállapota?
6. Különböző ágensnyelvek és protokollok összehasonlítása.
7. Ágensnyelvek és protokollok könnyű implementálhatósága produkciós (szabályalapú) rendszerekben.
8. Biztosítani, hogy a kommunikáció implementálása nyílt elosztott rendszerekben egységes (nyílt MAS-ban ACL olyan, mint egy API)

Párbeszédjáték (Dialogue Game) protokollok MAS számára

– Szabályozzák a kimondások kapcsolatát és kombinálását.
– Ágens nem mondhat akármit, akármikor.
– Zavaró viselkedés általában ki van szűrve.
- Protokoll több információt szállít (a kimondások alátámasztásai).
- Mások kimondásait lehet kérdőre vonni.
- Komplex kötelezettségek felépíthetők kimondások révén.

DG Protokollok specifikálása

Indítási szabályok
Legális lokuciók
Lokuciók kombinálásának szabályai
Kötelezettség-létesítés és kombinálásának szabályai
Beszélőváltás szabályai
Párbeszédterminálási szabályok

Kötelezettség tárak A párbeszéd résztvevői által vállalt kötelezettségek nyomon követése

Dialektikus kötelezettségek – pl. kötelezettség egy állítást megindokolni
Szemantikus kötelezettségek – pl. kötelezettség egy cselekvés végrehajtására

Fatio Protokoll/1

További lokuciók az érvelések megvalósítására.

assert(A, θ) – A állít θ-t és dialektikus kötelezettséget vállal, hogy az állítását megindokolja, ha erre megkérik.

question(B, A, θ) – B megkéri A-t a korábbi θ állításának megindoklására. B-re a dialektikus kötelezettség nem vonatkozik.

challenge(B, A, θ) – B megkéri A-t a korábbi θ állításának megindoklására. B dialektikus kötelezettséget vállal, hogy megindokolja, miért ellenzi a θ-t.

justify(A, Δ |- θ) – A, akit θ-ről megkérdeztek, vagy megtámadtak, megadhatja a θ (vagy θ negálásának) indoklását.

retract(A, θ) – A a θ-ra vonatkozó korábbi állítását, vagy indoklását visszavonja.

Fatio protokoll/2 A lokuciók kapcsolata

CR1: *assert* bármikor alkalmazható.

CR2: egy *assert*, vagy *justify* követően *question* és *challenge* bármikor alkalmazható.

CR3: ha egy *assert*-et, vagy *justify*-t egy *question, vagy challenge* lokucióval megkérdőjeleznek, az *assert*-et kimondó ágensnek azonnal kell *justify*-al válaszolnia.

CR4: *challenge-t* követően bármikor alkalmazható a *question* és a *challenge*.

CR5: ha egy *challenge*-t *question* vagy challenge lokucióval kérdőre vonjuk, az eredeti *challenge*-et kimondó ágensnek azonnal kell *justify*-al válaszolnia.

CR6: *retract* bármikor alkalmazható az ugyanannak az ágensnek *assert*, vagy *justify* lokuciója után.

Fatio protokoll/3 A szemantika

Axiomatikus szemantika
- A résztvevő ágensek hiedelmei és szándékai
- Magán-írt/publikusan-olvasott dialektikus kötelezettség tár

DOS(A) (dialectical obligation store)

(A, θ, +/-)

Assert(Pi, φ)

Pre-conditions: Pi beszélő azt kívánja, hogy minden Pj (j=/=i) résztvevő elhigye, hogy Pi hiszi a φ-t.

$((Pi, \varphi, +) /\in DOS(Pi)) \wedge (\forall j =/=i)(Di\ Bj\ Bi\ \varphi)$.

Post-conditions: Minden Pk (k =/=i) résztvevő azt hiszi, hogy Pi azt kívánja, hogy minden Pj (j =/=i) résztvevő azt higye, hogy Pi hiszi a φ-t.

$(Pi, \varphi,+) \in DOS(Pi) \wedge (\forall k =/=i)\ (\forall\ j =/=i)\ (Bk\ Di\ Bj\ Bi\ \varphi)$.

Dialectical Obligations: (Pi, φ, +) –t hozzáadjuk a DOS(Pi)-hez (a Pi beszélő dialektikus kötelezettség tárához)

Fatio protokoll/4 Operatív szemantika

D1(φ): Claim or Not: Egy procedura minden φ állításhoz, ami lehetővé teszi Pi ágens számára, hogy eldöntse, kimondja-e az assert(Pi, φ) lokuciót, vagy sem. A mechanizmus kimenete: listen ill. utter-assert(φ).

D2: React or Not: Egy procedura minden φ állításhoz, ami lehetővé teszi Pi ágens számára, az assert(Pk, φ)-t követően eldöntse, kimondja-e a question(Pj, Pi, φ) vagy a challenge(Pj, Pi, φ) lokuciókat,
A mechanizmus kimenete: listen, utter-question(Pi, φ),
vagy utter-challenge(Pi, φ).

D3(φ): Defend or Not: Egy procedura, hogy egy állítása kapcsán indoklás dialektikus közelezettsége alatt lévő Pi ágens eldönthesse, hogy az indoklás lokuciót kimondja-e a közelezettség teljesítése érdekében.
A mechanizmus kimenete: listen ill. utter-justify(.)

D4(φ): Fold or Not: Egy procedura, hogy egy állítása kapcsán indoklás dialektikus közelezettsége alatt lévő Pi ágens eldönthesse, hogy a retract(.) lokuciót kimondja-e. A mechanizmus kimenete: listen ill. utter-retract(.).

D5: Listen or Do: Egy procedura, hogy az ágens a résztvevők üzeneteire vár és ha érkeznek, eldönthesse, hogy mely D1-D4 viselkedéshez folyamodjon. A kimenet: listen és do-mech(Di), i= 1,2,3,4.

Fatio protokoll/5 Állapotátmenetek rendszere

⟨Pi, **K**, s⟩ Pi ágens által alkalmazott **K** mechanizmus s kimenettel Pi részéről
⟨Pi, **K**,.⟩ nincs konkrét kimenet

Állapotátmenet ua. az ágens mechanizmusai között
⟨Pi, **D5**, listen⟩ → ⟨Pi, **D5**,.⟩

Állapotátmenet két különböző ágens mechanizmusai között egy lokució révén
⟨Pi, **D1**, utter-assert(φ)⟩ F1 → ⟨Pj, **D5**, do-mech(D2)⟩

F1→	*assert* kimondása
F2→	*question* kimondása
…	

(részlet)

TR1: ⟨Pi, **D1**(φ), listen⟩ → ⟨Pi, **D5**,.⟩
TR2: ⟨Pi, **D1**, utter-assert(φ)⟩ F1 → ⟨Pi, **D5**, listen⟩
TR3: ⟨Pi, **D1**, utter-assert(φ)⟩ F1 → ⟨Pj, **D5**, do-mech(D2)⟩
TR4: ⟨Pj, **D2**, listen⟩ → ⟨Pj, **D5**, .⟩
TR5: ⟨Pj, **D2**, utter-question(Pi, φ)⟩ F2 → ⟨Pj, **D5**, listen⟩
TR6: ⟨Pj, **D2**, utter-question(Pi, φ)⟩ F2 → ⟨Pi, **D5**, do-mech(D3(φ))⟩
TR7: ⟨Pj, **D2**, utter-question(Pi, φ)⟩ F2 → ⟨Pk, **D5**, listen⟩
…

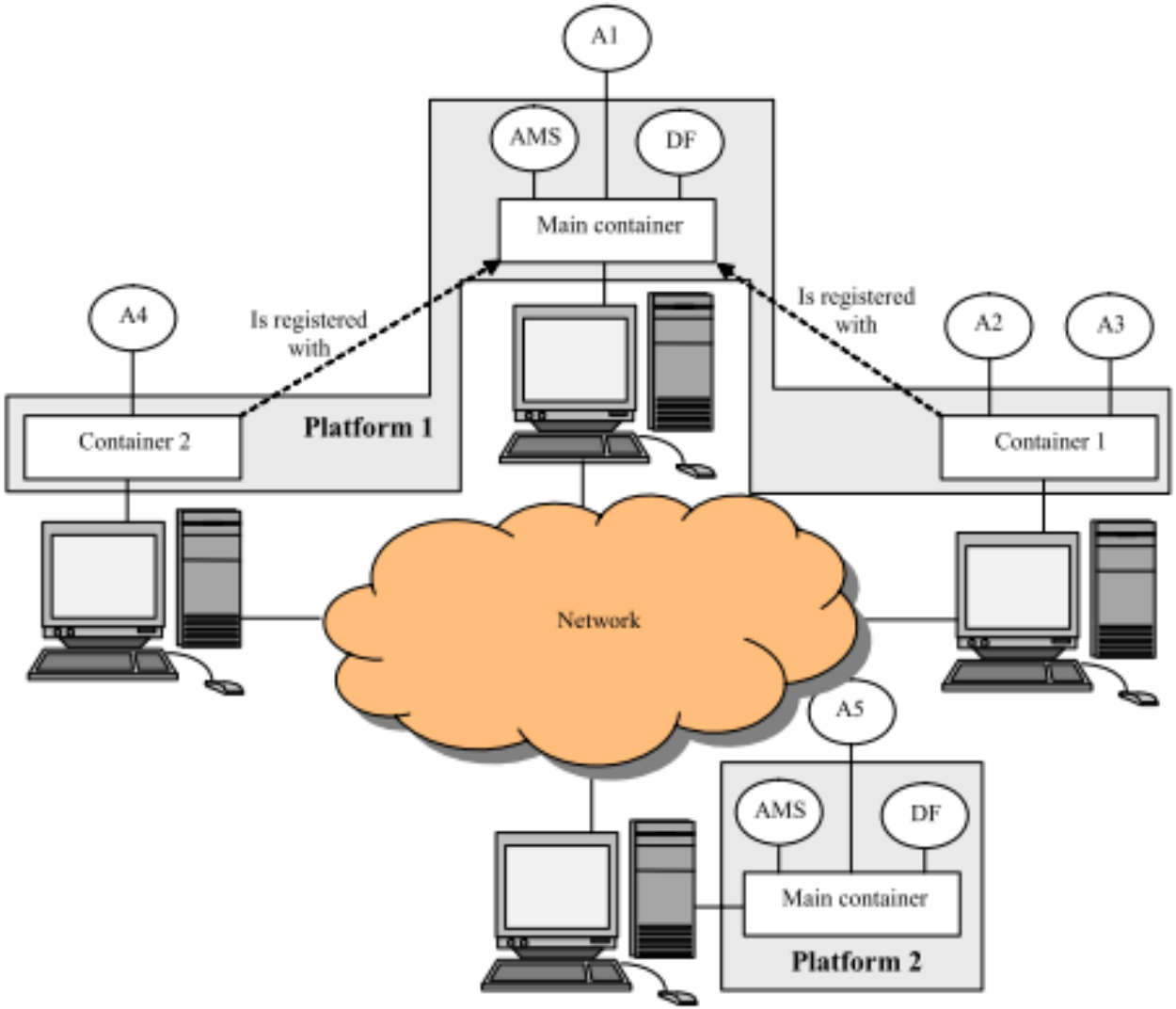
TR21: ⟨Pi, **D5**, do-mech(D4(φ))⟩ → ⟨Pi, **D4**(φ), .⟩

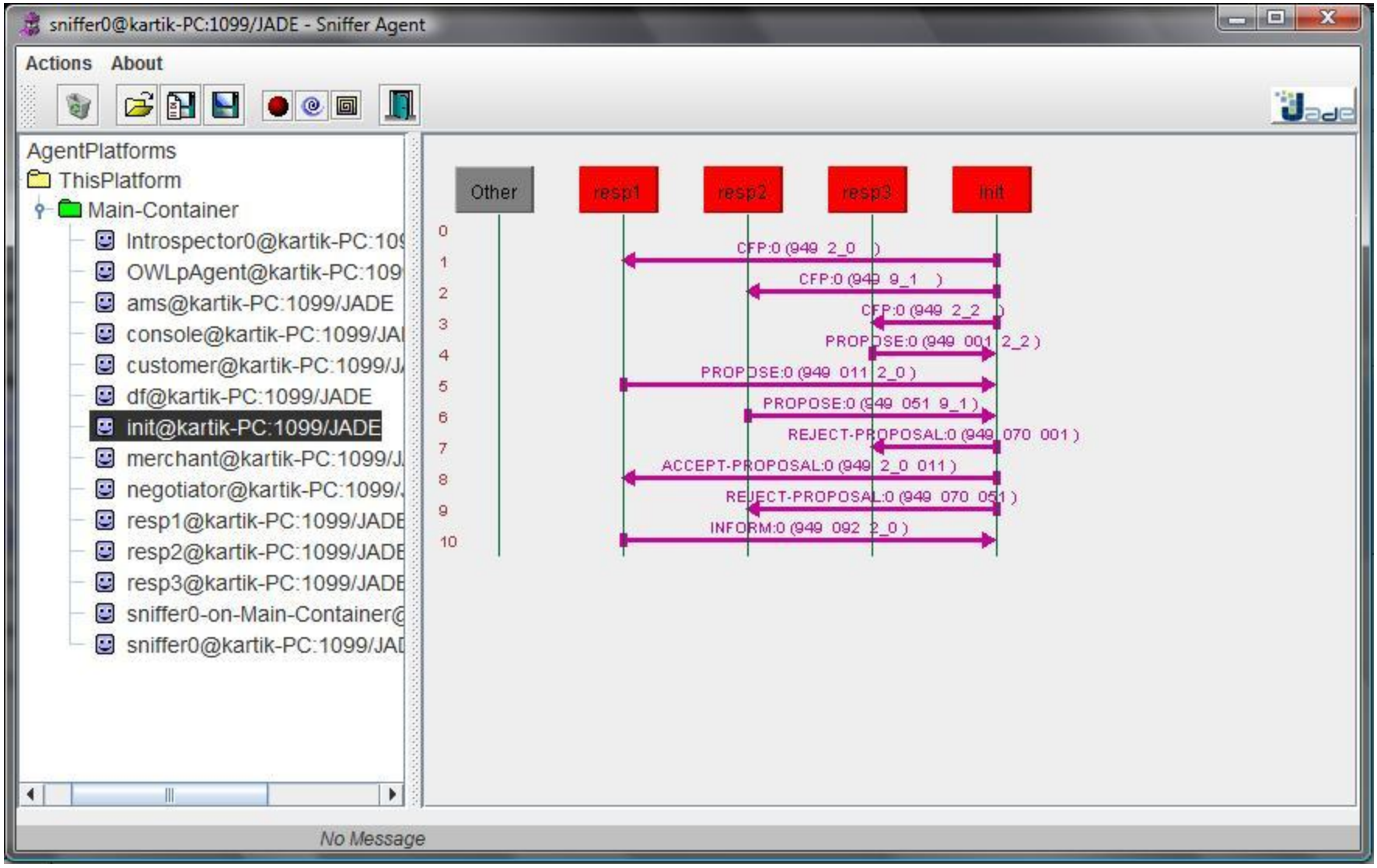
FIPA szabvány –
Jade platform – kisérleti ágensközösség

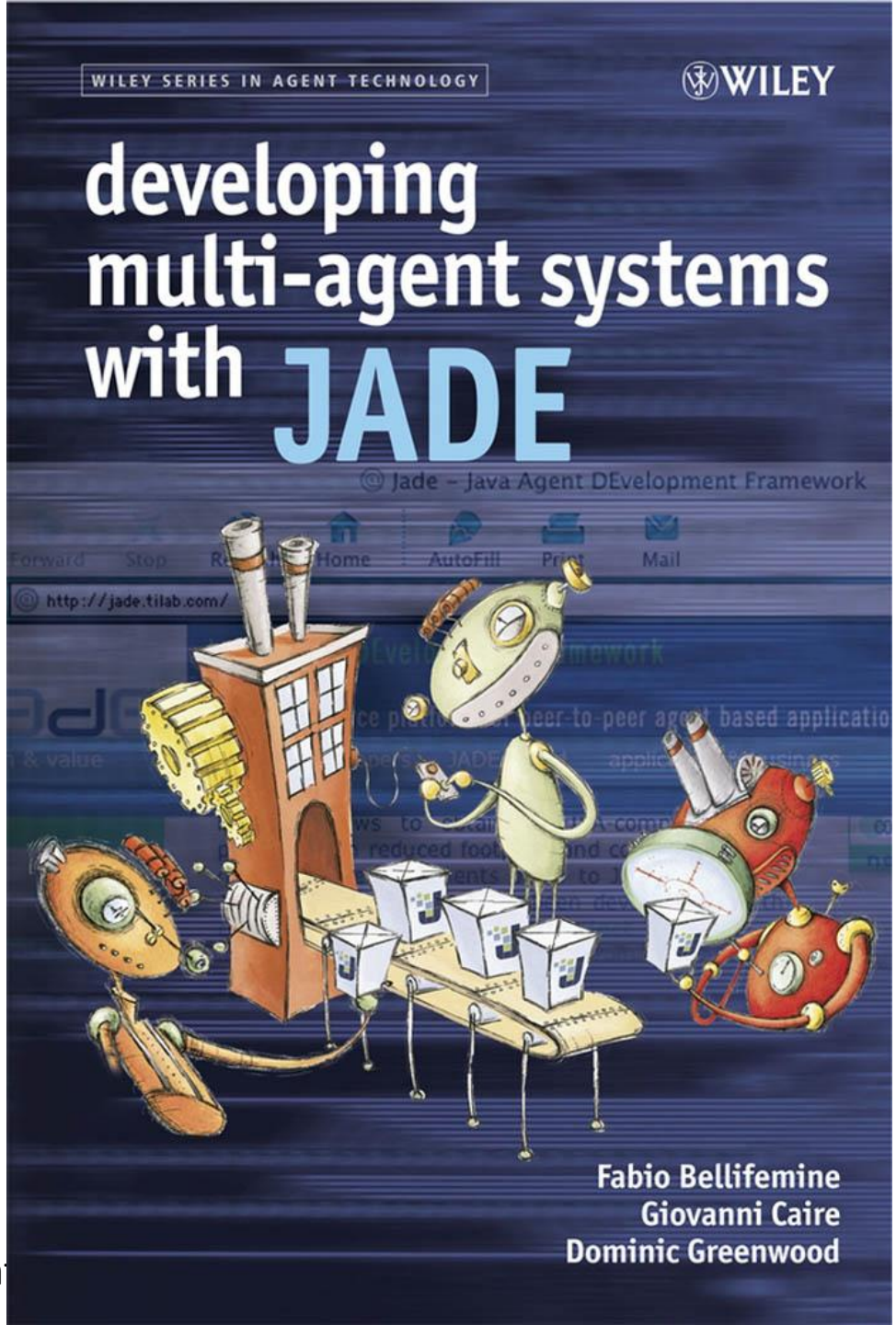
Jade a FIPA szabvány (www.fipa.org) implementációja Java-ban.
Jade rendszer indulásakor egy **platformot** létesít (több hoszton elosztott), ahol az un. **konténerekben** indíthatók az ágensközösségek.
A platform logikai infrastruktúra, ez oldja meg az ágensek kommunikációját.

A platformhoz tartoznak hivatalból (automatikusan jönnek is létre):
AMS (**Agent Management System**) - a platform (közösség) „kapuőre”,
ACC (Agent Communication Channel) – a kommunikáció ágense,
DF (**Directory Facilitator**) - a platform telefon könyve (Sárga Oldalak),

A platform felhasználói felületéhez tartoznak:
RMA (**Remote Monitoring Agent**) - beépített kémágens, amely mindenről és mindenkiről tart számon,
Sniffer - amely a kommunikáció forgalmát megjeleníti,
Introspector - amely az ágensek életútját kiséri végig,
kívánság szerint akár több **DA** (**Dummy Agent**) - ágensbőrbe bújtatott emberi felhasználó (embert csomagoló wrapper-agent).


A1
AMS
DF
Main container
A4
Is registered with
Is registered with
A2
A3
Container 2
Platform 1
Container 1
Network
A5
AMS
DF
Main container
Platform 2


sniffer0@kartik-PC:1099/JADE - Sniffer Agent
Actions
About
AgentPlatforms
ThisPlatform
Main-Container
Introspector0@kartik-PC:109
OWLpAgent@kartik-PC:109
ams@kartik-PC:1099/JADE
console@kartik-PC:1099/JA
customer@kartik-PC:1099/J
df@kartik-PC:1099/JADE
init@kartik-PC:1099/JADE
merchant@kartik-PC:1099/J
negotiator@kartik-PC:1099/
resp1@kartik-PC:1099/JADE
resp2@kartik-PC:1099/JADE
resp3@kartik-PC:1099/JADE
sniffer0-on-Main-Container@
sniffer0@kartik-PC:1099/JA
Other
resp1
resp2
resp3
Init
0
1
2
3
4
5
6
7
8
9
10
CFP:0 (949 2_0)
CFP:0 (949 9_1)
CFP:0 (949 2_2)
PROPOSE:0 (949 001 2_2)
PROPOSE:0 (949 011 2_0)
PROPOSE:0 (949 051 9_1)
REJECT-PROPOSAL:0 (949 070 001)
ACCEPT-PROPOSAL:0 (949 2_0 011)
REJECT-PROPOSAL:0 (949 070 051)
INFORM:0 (949 092 2_0)
No Message


WILEY SERIES IN AGENT TECHNOLOGY
WILEY
developing multi-agent systems with JADE
Jade – Java Agent DEvelopment Framework
http://jade.tilab.com/
Fabio Bellifemine
Giovanni Caire
Dominic Greenwood

In

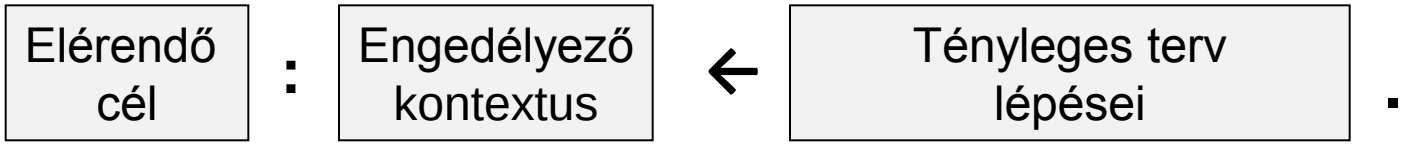
BDI ágensek programozási nyelvei – AgentSpeak(L)

Ágens AgentSpeak specifikációja:

hiedelmek halmaza (logikai jellegű tények)
tervek halmaza (kontextus-érzékeny, esemény-triggerelt receptek a hierarchikusan dekomponálható célok elérésére, elemei cél-orientált cselekvések).

Ágens viselkedése: receptkönyvtár (“tervek”) = reaktív tervvégrehajtó rendszer

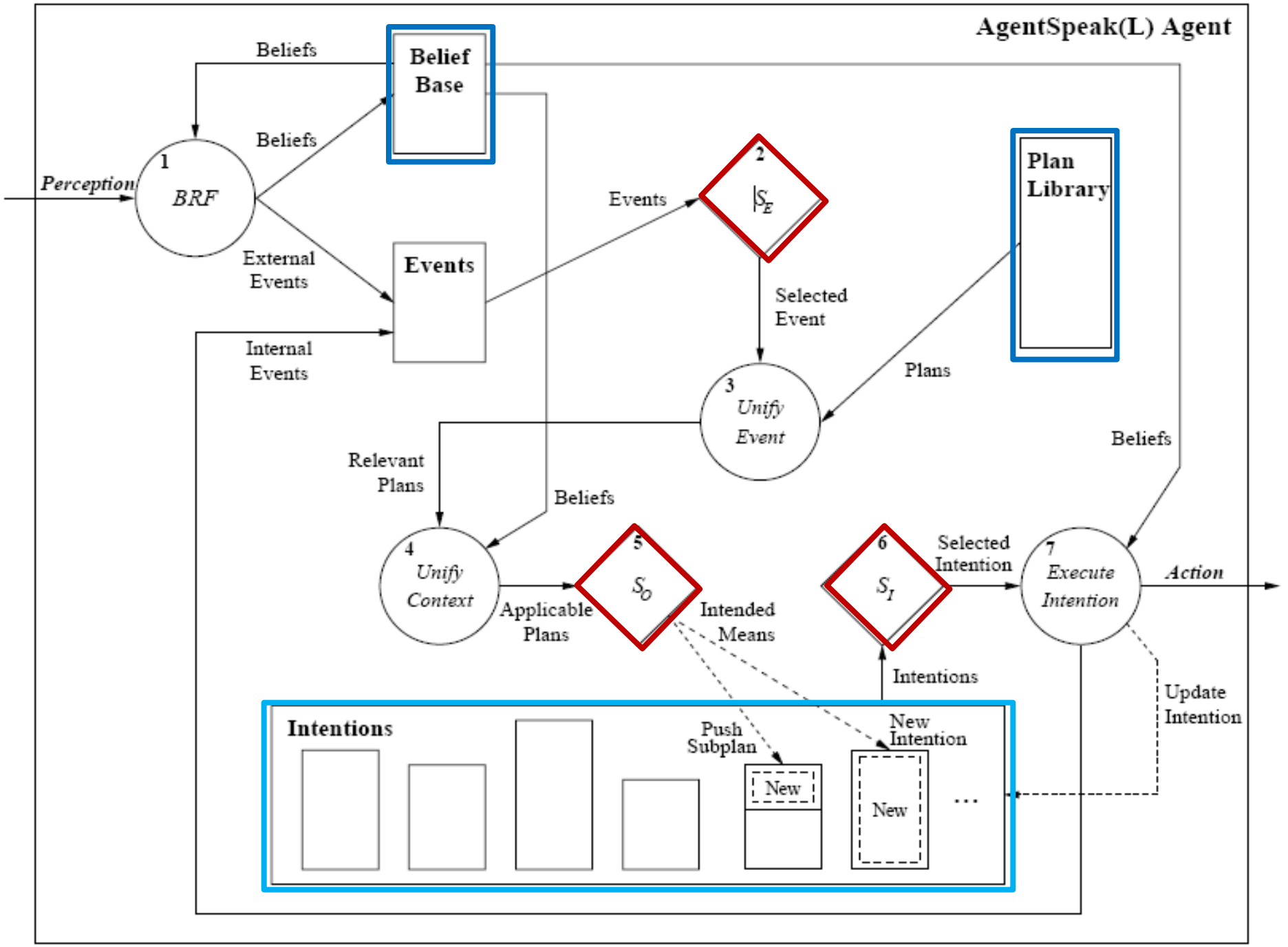
Terv:

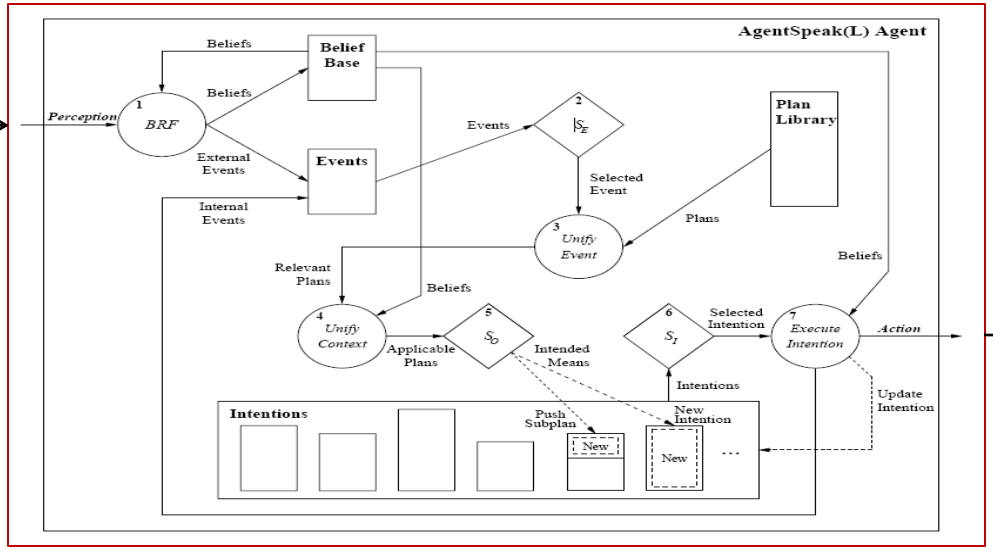


Cél: +!goal, +?goal, −!goal, −?goal, +belief, −belief **teljesítési célok**, **teszt célok**

Kontextus: belief | Context $\wedge$ Context | Context $\vee$ Context | $\neg$Context | $\exists$x.Context

Terv lépései: action | +belief | −belief | ?Context | !event | Plan; Plan


AgentSpeak(L) Agent
Beliefs
Belief Base
Beliefs
Perception
1
BRF
External Events
Events
Events
2
S_E
Selected Event
Plan Library
Internal Events
Plans
3
Unify Event
Beliefs
Relevant Plans
Beliefs
4
Unify Context
Applicable Plans
5
S_O
Intended Means
6
S_I
Selected Intention
7
Execute Intention
Action
Intentions
Update Intention
Intentions
Push Subplan
New Intention
New
New
...



Érzet

Cselekvés

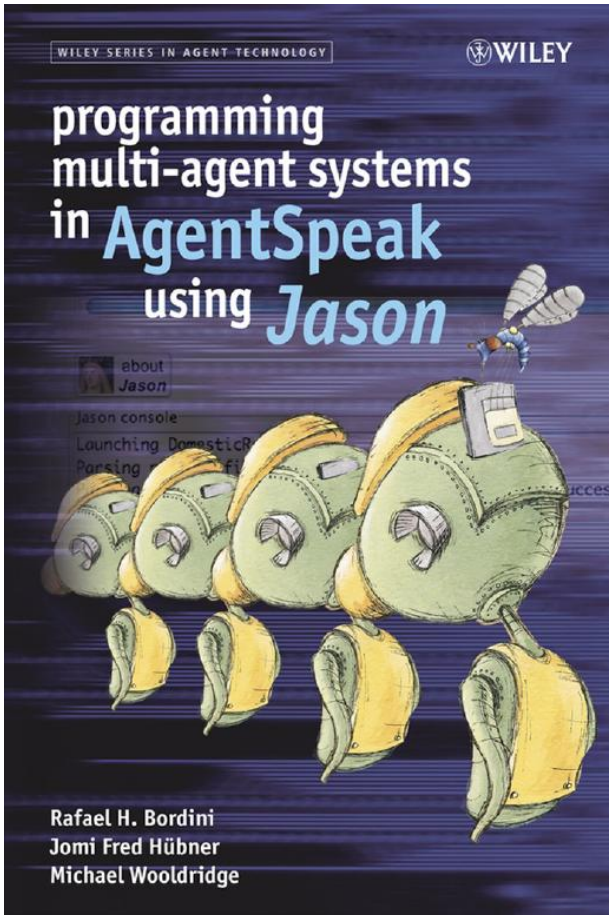
Környezet

beépített/programozható cselekvések és érzetek

Jason – AgentSpeak platform Javaban, kiterjesztésekkel

http://jason.sourceforge.net/

Szabadon kiterjeszthető,
felhasználói szinten módosítható (Java)



MAS – Együttműködés

Együttműködés = megosztás + kommunikáció

Mit lehet megosztani:

information, tudás, adat, eredmény, konklúzió, hipotézis, …

ilyen megosztásból általában „homogén” közösségek („tudásban egyenlők”),

feladat, cél, ...

ilyenből pedig „strukturált” közösségek (specializálódás) profitálnak .

Együttműködés lehetséges, ha a felek (I/N?) :

kognitív módon elismerik egymást

közös céljuk van (akár negatív jelenségek mérséklése)

etikailag elismerik egymást

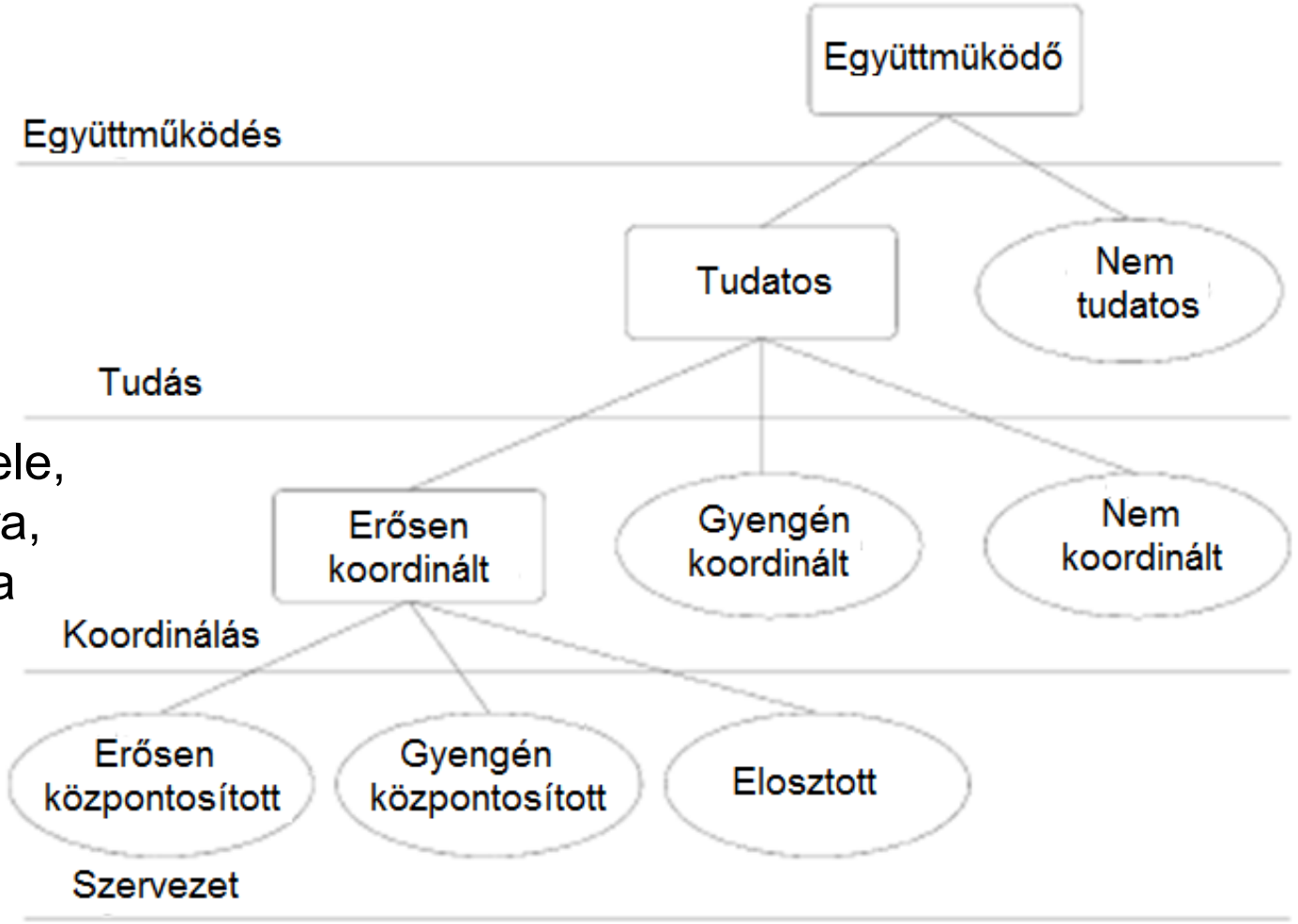
megbíznak egymásban

…

Együttműködés szintjei

Koordinálás dim:
együttműködés,
tudás,
koordinálás,
szervezet

Rendszer dim:
kommunikáció,
szervezet összetétele,
rendszerarchitektúra,
szervezet nagysága



Együttműködési protokollok

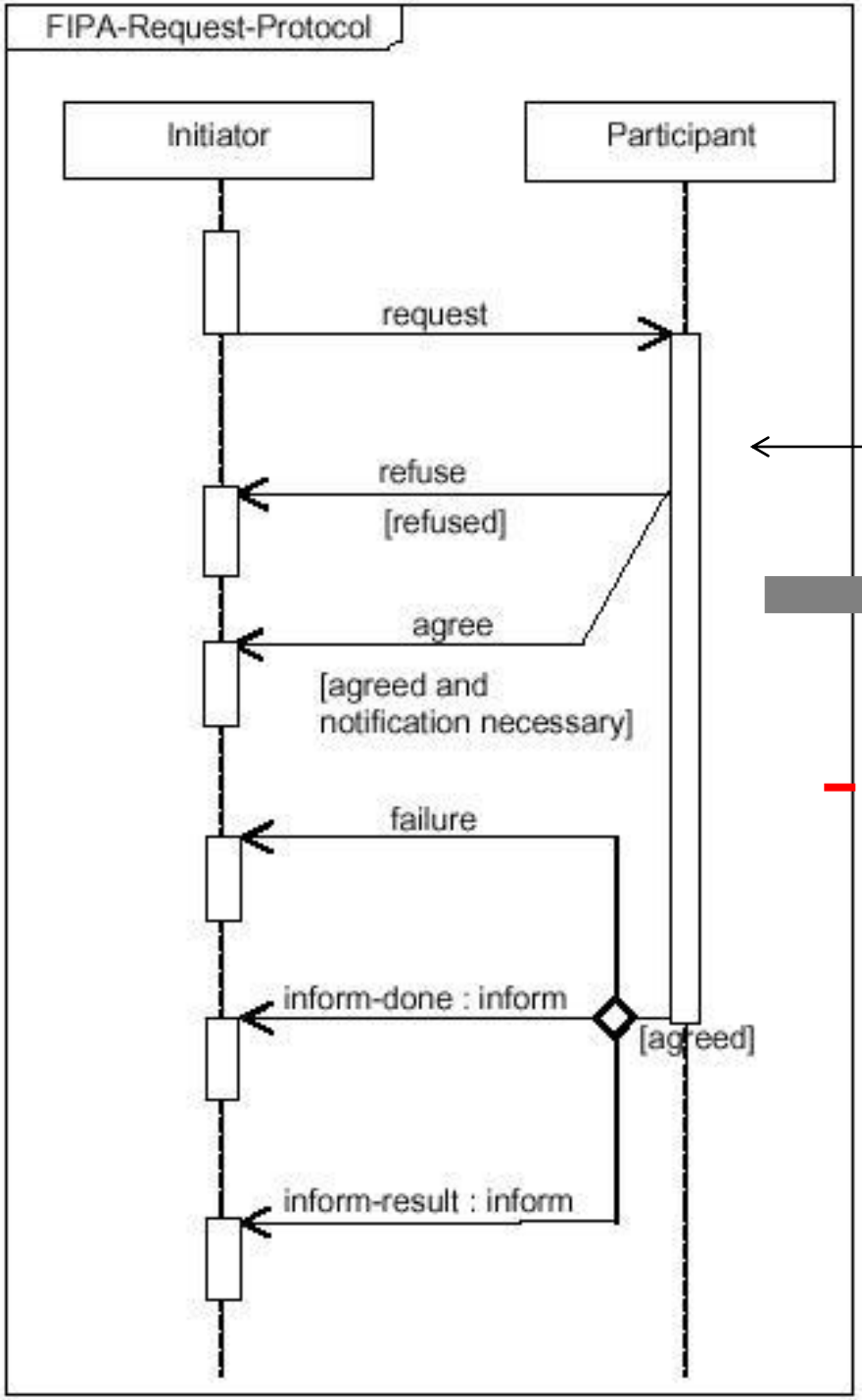
Globális koherencia elérése a működésben (közös kontextus),
de egyéni autonómia nem megsértése.

Megosztott célok, közös feladatok azonosítása,
de szükségtelen protokollok kerülése. Kritikus erőforrások terhelését kerülni .

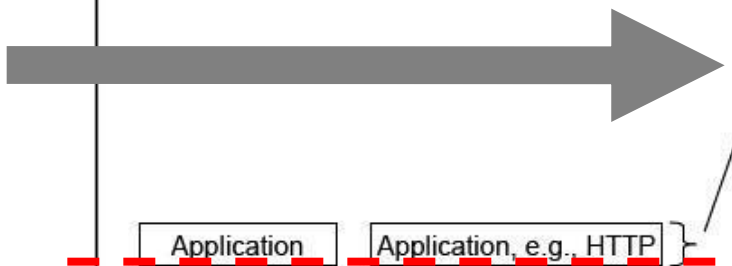
Tudás- és bizonyíték fuzionálása (ezt mind jó lenne tudni biztosítani),
feladatokat megfelelő képességekkel párosítani,
de figyelni a kommunikáció és szinkronizálás költségeinek csökkentésére.

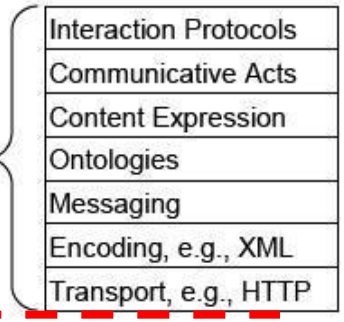
protokoll $\Rightarrow$ informális
$\Rightarrow$ formális
$\Rightarrow$ leírása A(gent extended)UML, XML, FSM, stb.
$\Rightarrow$ logikával (információs attitűdök, modalitások, idő)

protokoll verifikálás:
logikai
heurisztikus: szimulációk
(pl. komplex gyártási folyamat, elosztott ágensek, …)



AUML modell





OSI	TCP/IP
Application	Application, e.g., HTTP
Presentation	
Session	
Transport	Transport, e.g., TCP
Network	Network, e.g., IP
Data link	Host to Network. E.g., Ethernet
Physical	

ACL

Protokollok tipusai (szervezettől függ)

zárt(abb) szervezet (hierarchia, team, közösség, ...)

- üzenetváltás szekvencia
  - merev (mechanikus, eltérés nem lehetséges)
  - nem kiterjeszthető („bedrótozott”)
- közösség erősen strukturált, megjelenik benne főnöki szerep
- taszk megosztás folyik,
- közösség gyengébben strukturált, megjelenik benne
- aszimmetrikus menedzseri szerep, taszk megosztás folyik

nyílt(abb) szervezet (piac, ..., web, e-kereskedelem, ...)

- hiedelem-, cél alapú, rugalmas, egyenlősdi közösség,
- nagyjából szimmetrikus hatások:
- információ, eredménymegosztás folyik,
- érvelés alapú tárgyalási protokoll, aukció, szavazás

gyengén strukturált közösségek, megjelenik a döntnöki szerep

- információ-megosztás: konfliktushelyzetek kezelése speciális protokollokkal

Taszk megosztás Vállalkozási háló protokollal (*Contract Nets*)

1. Menedzser ágens átveszi a feladatot és lebontja kisebb „porciókra”, „taszkokra”. (eldönti, hogy nem képes egyedül megoldani)
2. Menedzser ágens vállalkozót keres taszkjához, vagy taszkjának egyes részeihez.
3. E célból szétküldi (broadcast) a taszk (probléma) leírását (esetleges Megkötésekkel, pl. határidő) és kedvező ajánlatokra vár.
4. Vállalkozó ágensek összemérik a meghirdetett taszk leírását a saját képességeivel (tudás modellel) és vagy nem reagálnak, vagy beküldik a jelentkezésüket (milyen feltételekkel vállalkoznak a feladatra, milyen minőségű megoldást képesek szállítani, stb.).
5. Menedzser ágens választja ki a legjobbnak tűnő ajánlatot és a feladatot véglegesen kiadja elvégzésre.
6. Vállalkozó ágens a feladatát megvalósítja és a megoldást a Menedzsernek beküldi.
7. A befutó megoldásokból a Menedzser ágens összerakja a teljes feladat megoldását és a felhasználónak elküldi.

Vállalkozási háló protokoll (*Contract Nets*)

Szükséges üzenetek (informálisan)

- Taszkok meghírdetése
- Licitbeküldés
- Taszk odaítélés
- Előrehaladás beszámolója
- Végleges beszámoló (eredménnyel együtt)
- Terminálló üzenet (hátha a Menedzser a szerződést fel szeretne bontani)

Vállalkozási háló protokoll (*Contract Nets*)

A feladatot végző konkrét ágens identitása előre **ismeretlen**,

A Menedzser ágens tudása lényegesen több, mint a Vállalkozó ágensekké, abban, hogy az ő feladata kapcsolattartás a felhasználóval és a feladat dekompozíció /megoldás szintézis.

Nem kell tudnia a feladatot megoldani, viszont tudnia kell mérlegelni a beküldött jelentkezéseket (mások modellje, feladat modellje, stb. alapján).

Minden vállalkozó ágensnek rendelkeznie kell:

- problématerületre vonatkozó problémamegoldó tudással,
- önmaga modelljével, hogy a vállalkozás sikerét mérlegelni tudja.

Szabad autonóm módon **elutasítania** is ...

Lehetőség: a **fokozatos romlás** biztosítása (szuboptimális viselkedés, …)