

# **Kooperatív és Tanuló Rendszerek**

## **4. ACL-KQML-szabványok**

**Dobrowiecki Tadeusz  
Horváth Gábor**

# Knowledge Sharing Effort (KSE)

ARPA, ca. 1990 tudásmegosztás, tudás újrafelhasználása

tudásmegosztás – kommunikáció, közös nyelv: szintaxis + szemantika + pragmatika

**KIF – Knowledge Interchange Format** (szintaxis)

**Ontolingua** – megosztható ontológiák nyelve (szemantika)

**KQML** – magas szintű kölcsönhatás (pragmatika)

**Knowledge Query and Manipulation Language**

ACL (Agent Communication Language) szótár **belső** nyelv (KIF)  
**külső** nyelv (KQML)

ACL üzenet – KQML kifejezés, argumentumai – KIF termék, vagy állítások

ACL üzenetek – „beszédaktusok után”

beszédaktus – ágens szándékszintű modellje

- reláció **ágens**

**jelentéssel rendelkező ítéletállítás**

**attütüdök véges halmaza (hisz, állít, félt, remél, ....)**

pl. < **ágens**, **fél**, **esik\_az\_eső(most)** >

Kommunikáció: **megmutatni** másoknak a BDI állapotunkat  
egy eszköz **kísérlet** mások BDI állapotának **befolyásolására**

# KQML (Knowledge Query and Manipulation Language

– tudás lekérdező és manipuláló nyelv)

## Ágens alapú rendszerek interfész kérdése

**üzenetszállítás** módja

**nyelv**, mi az üzenet jelentése

**stratégia/eljárás** = a párbeszéd struktúrája

**architektúra** = rendszerek összekapcsolása

KQML több, mint egy ACL – **ajánlás kommunikációra, üzenet tartalomra,**  
kontextus szinteken kívül **architekturális elemekre.**

## A kommunikáció szintjei

### Adatcsere 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):

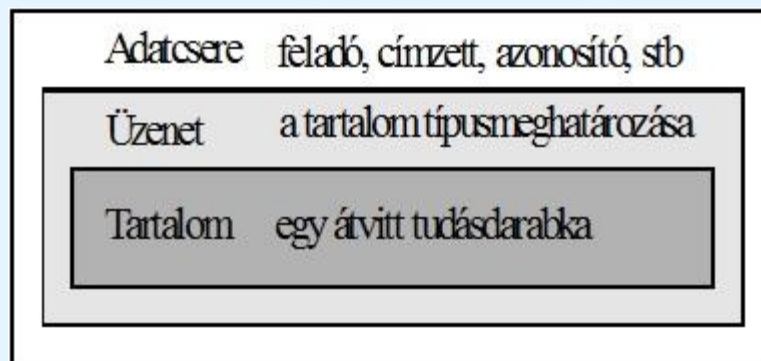
*üzenettartalom azonosítása,*

*típus-meghatározása*

### Tartalom szint (kommunikáció tartalma):

*Az átadott információ közvetítése.*

Bármit tartalmazhat, amiben a kommunikáló ágensek megegyeznek.



# Üzenettípusok a KQML nyelvben

(avagy milyen performativumból merítenek)

**Alapvető lekérdezés** - *evaluate, ask-if, ask-oke, ask-in, ask-all*

A kérdező egy kérdés kiértékelését kéri egy másik ágenstől, melyre egy választ vár

**Többválaszos lekérdezés** - *stream-in, stream-all*

Az ágens azt kéri, hogy a kérdésre minden esetben válaszoljon, amikor az ahhoz kapcsolódó tudása megváltozik. Ez az üzenet tehát a jövőben több választ is eredményezhet, attól függően, hogy a válaszadó milyen információkat szerez.

**Válaszok** - *reply, sorry*

Az egy vagy többválaszos kérdésekre érkező válasz, illetve annak jelzése, hogy a válaszadónak nincs adat a birtokában.

**Általános közlés** - *tell, **achieve**, cancel, untell, **unachieve***

A *tell* egy állítás, az *achieve* egy kívánság, mely állítás teljesülését a küldő szeretné elérni. Mindkettőnek létezik a visszavonása is, illetve van egy külön kulcsszó egy azonosítóval meghatározott korábbi üzenet törlésére.

**Generátor típusú üzenetek** - *standby, ready, next, rest, discard, generator*

Ezen üzenetek mindegyike további üzenetek küldését kezdeményezi.

**Képesség definíciók** - *advertise, subscribe, monitor*

Az ágensek felajánlhatják szolgáltatásaikat, illetve igényelhetik mások szolgáltatását.

**Hálózattal kapcsolatosak** - *register, unregister, forward, broadcast, route*

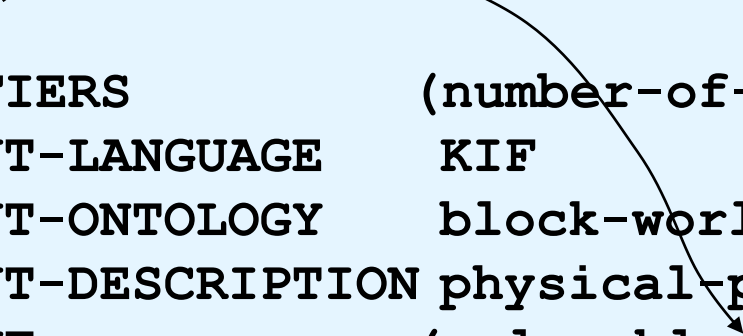
Ezen kulcsszavak írják le egyrészt új ágensek bejelentkezését, régiek kilépését a rendszerből, másrészt az ágensek közötti csomagok továbbküldését, szétküldését, illetve útjának meghatározását.

Bár a formára sincs megkötés, a KQML implementációk általában a LISP nyelvben megismert listás (lineáris) adatszerkezetet használják.

Az **üzenet szint** szerepe kétféle lehet: **adminisztratív** vagy **tartalom** jellegű.

**Adminisztratív üzenet** tartalma kötött, a rendszer működéséhez szükséges adatokat közvetíti. (pl. *az új ágensek bemutatkozása, ágens képességek közzététele, illetve egy ágens által igényelt adatok leírása.*)

**Tartalom típusú üzenet** - egy tudásdarabka átvitele, meghatározva annak ontológiáját, témáját, nyelvét és tartalmát.

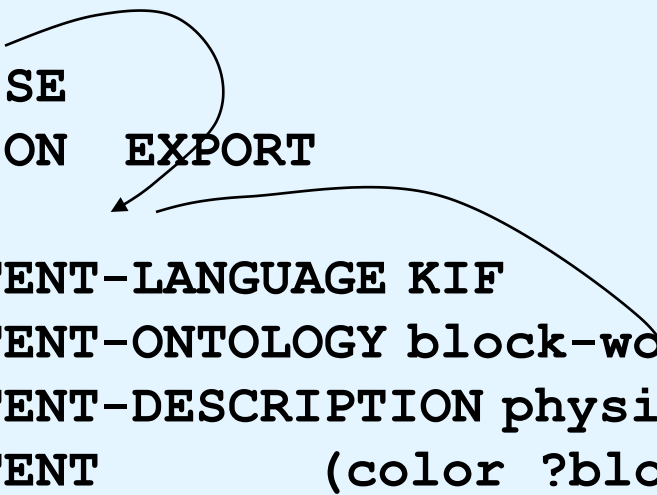


```
(QUERY
  QUALIFIERS      (number-of-answers 1)
  CONTENT-LANGUAGE KIF
  CONTENT-ONTOLOGY block-world
  CONTENT-DESCRIPTION physical-property
  CONTENT         (color block1 ?color))
```

*KQML tartalom jellegű üzenet:* Egy kérdés a kocka világban, KIF nyelven, egy fizikai tulajdonságra vonatkozó kérdés: a *block1* nevű kocka színe.

Egy ágens “hirdetése” - az ágens azt a képességét fogalmazza meg, hogy képes kockák színére utaló állításokat tenni (azaz ilyen kérdésekre válaszolni).

```
(ADVERTISE
 DIRECTION EXPORT
 (TELL
  CONTENT-LANGUAGE KIF
  CONTENT-ONTOLOGY block-world
  CONTENT-DESCRIPTION physical-property
  CONTENT          (color ?block ?color)))
```



*KQML adminisztratív üzenet:* **Ez esetben az üzenet tartalma egy újabb üzenet!**

**A performatívumok egymásba ágyazhatóak, így komplex üzenetek is kialakíthatóak.**

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))

**(tell** :language KIF :ontology structures :in-reply-to s1

:content (fastens frame12 motor1))

**(untell** :language KIF :ontology motors :in-reply-to s1

:content (= (val (torque motor1) (sim-time 5)) (scalar 12 kgf))

**(tell** :language KIF :ontology motors :in-reply-to s1

:content (= (val (torque motor1) (sim-time 5)) (scalar 13 kgf))

.....

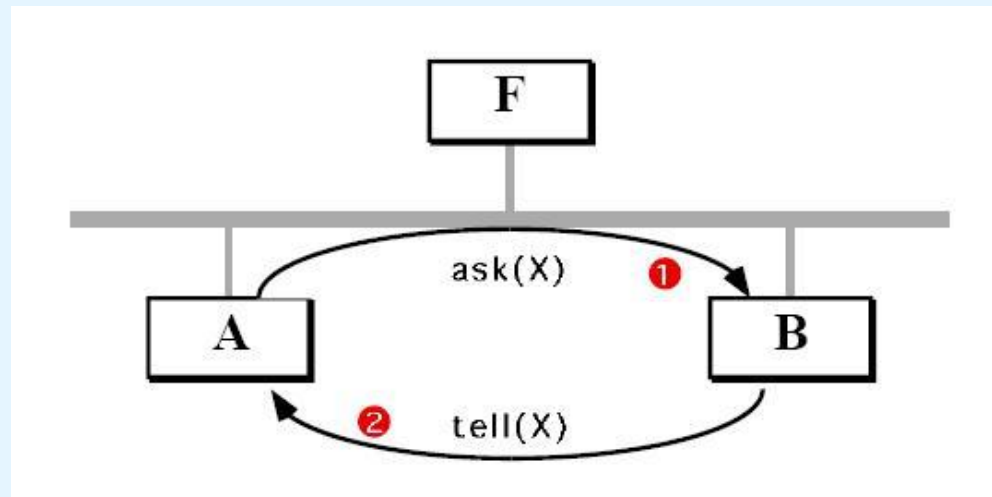
## A KQML architektúra

- az adatcserét közvetlenül megvalósító **KQML útvonal választó (KQML router)** minden egyes ágens mellett, az üzenetek tartalmuktól független továbbítása, illetve adminisztrálja az ágens rendszerbe lépését, illetve távozását.
- magasabb funkciókat ellátó **kommunikációt segítő ágens (facilitator)**. minden magasabb szintű feladatot lát el

### **A kommunikációt segítő ágens feladatai**

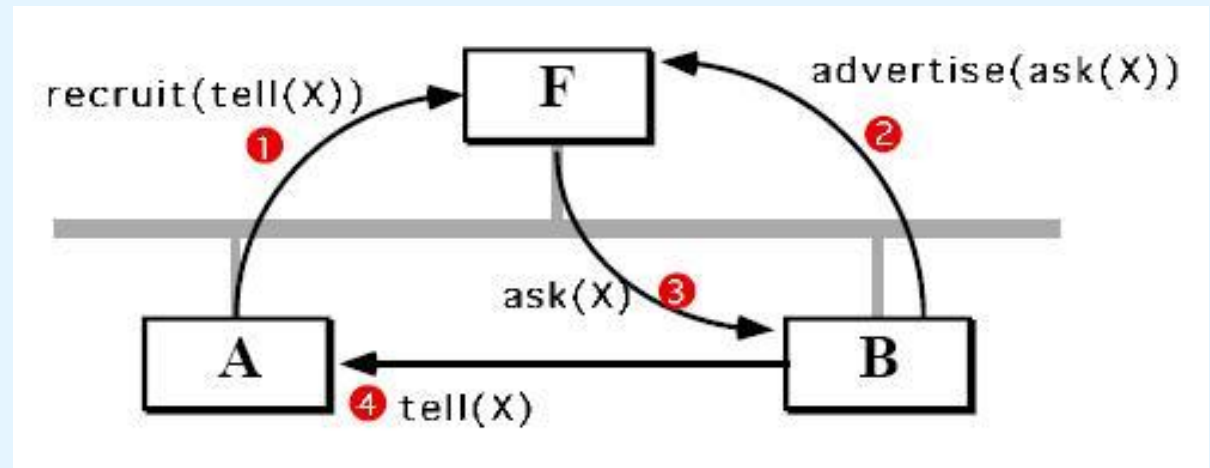
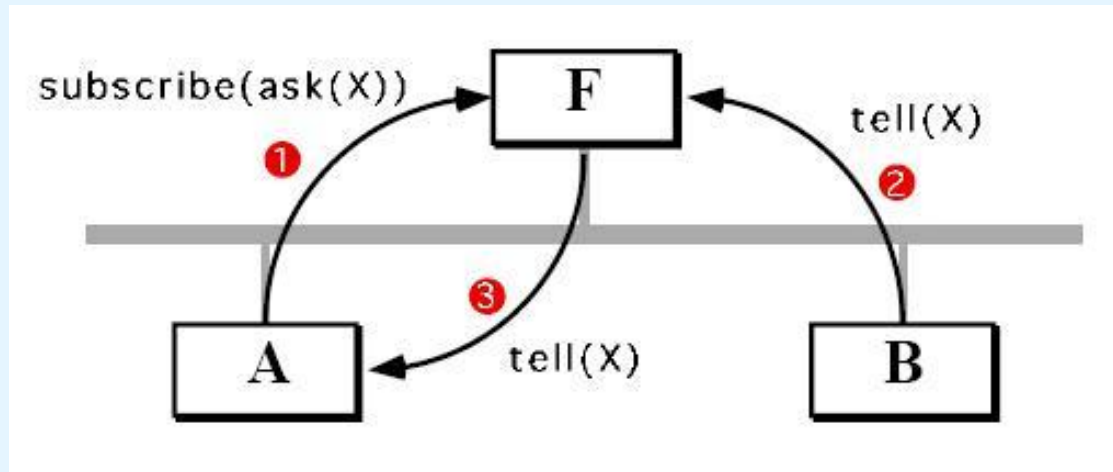
A routerek hozzá fordulnak a nem közvetíthető, pl. nem teljesen címzett csomagokkal. Az üzenetek tartalma alapján képes a pontos címzés előállítására. (*SzR*)  
Nyilvántartja az ágensek neveit és címét: elvégzi a név-cím transzformációt. Ágensek közötti tartalom fordítást végez. (*ontológia szerver*)  
Az ágensek hirdeteményeit tárolja, illetve felhasználja a kérdező-válaszadó párok kialakításában. (*Jade DF*)

### **Tipikus kommunikációs utak**

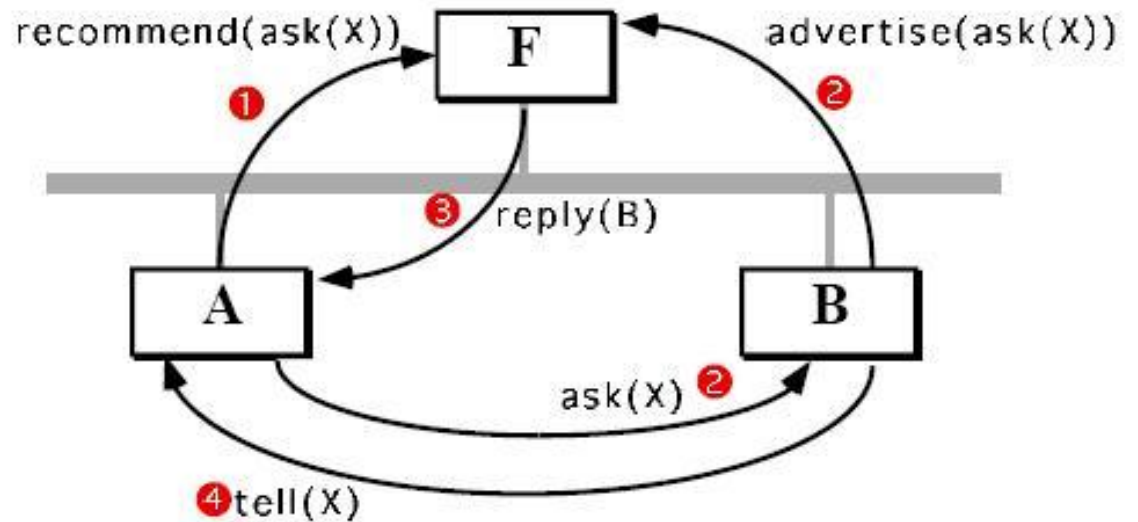
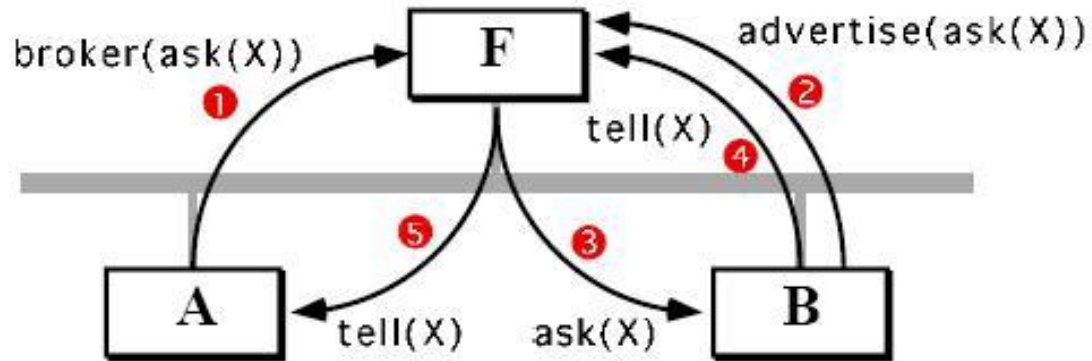




# Tipikus kommunikációs utak



# Tipikus kommunikációs utak

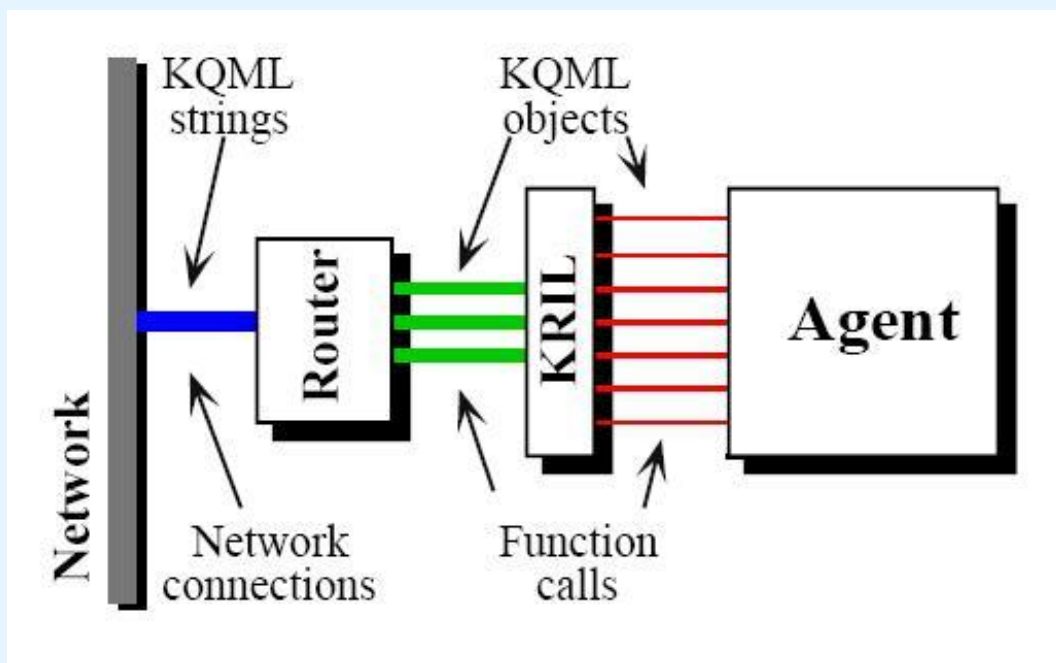


## KRIL: ágensek integrálása a KQML rendszerbe

KRIL interfész (**KQML Router Interface Library**): ágens programok a KRIL API-t használva egyszerű függvényhívásokkal érik el a rendszer szolgáltatásait.

*send-kqml-message(char \*msg, int msglen);*

*declare-kqml-message-handler(void (\*hndlr)(char \*msg, int len));*



## Későbbi problémák a KQML rendszerrel

A legfontosabb kritika a **szemantika aluldefiniáltsága**, azaz, hogy a **performatívumok definíciói kétértelműek, hiányosak, rosszak.**

### Kétértelműség.

A szintaxissal pontosan definiált struktúrák szemantikája nem egyértelmű, **implementációtól függhet**, hiszen az üzenettípus magyarázata csak természetes nyelven fogalmazódott meg (konzisztens tervezés távoli platformokon nehéz).

(DENY tartalom .....jelentése, hogy a tartalom a hallgatóra nézve nem igaz.

de

(DENY (TELL tartalom ..... jelentése nem világos. Korábban nem mondta (letagadja, hogy mondta, vagy nem hiszi, amit most mond?

**nem (TELL A), vagy (TELL (nem A)) ?**

### Félreértelmezett performatívumok.

Néhány performatívum kulcsszó valójában nem performatívum, mivel **nem eredményez közvetlen akciót**, csak felkérést jelent, amit azonban a **fogadó figyelmen kívül hagyhat.**

(ACHIEVE felkérés hallgató felé, hogy valamije igaz legyen, de a formalizmus mellőzi a felkérés mechanizmusát. Nincs elkötelezettség a cselekvésre. Hogyan lehet a hallgatót rávenni a cselekvésre?

**KQML üzenetei valójában csupán a direktívumok, verdikátívumok és asszertívumok.**

## Hiányzó performatívumok.

Az alap KQML nyelvből teljesen hiányoznak a **megbízás típusú** performatívumok, amelyekre a fogadónak egy rövid válasszal mindenképpen reagálnia kell (elfogadom, nem fogadom el a megbízást).

Az ilyen jellegű azonnali visszajelzések hiánya nagyban ronthatja egy kooperáló rendszer hatékonyságát.

Mi hiányzik ?

a **komisszívumok!** az **elkötelezettség kifejezése** (ez a legfontosabb).

IGÉR

JAVASLATOT ELFOGAD

BELEEGYEZIK

....

stb.

## Szemantikailag specifikált KQML (1997)

performatívum: **min. előfeltételhalmaz + min. utófeltételhalmaz +**  
sikeresen elvégzett performatívum által előidézett **végállapot**

Dolgozzunk az alábbi modális operátorokkal:

- Bel** (A, P) P igaz (vagy bizonyítható) az A részére (a képességei szerint).
- Know** (A, S) S mentális állapot ismert az A által.
- Want** (A, S) A kívánja az S mentális állapot bekövetkezését.
- Int** (A, S) A szeretné az S-t, el van kötelezve a hozzá vezető cselekvésre.
- Proc** (A, M) A cselekvésében az M üzenetet dolgozza fel.
- Sendmsg** (A, B, M) A cselekvésében B-nek M üzenetet küld.

pl.

**Know** (A, **Bel** (B, „.....”)), feltéve, hogy **Bel** (B, „.....”) el lett küldve

### Szemantikus leírás komponensei:

- **természetes nyelvű leírás:** performatívum intuitív jelentése
- az illokúciós aktus/erő **formális kifejezés**
- **előfeltételek:** Pre(A) hogy elküldhesse  
Pre(B) hogy (sikeresen) feldolgozhassa
- **utófeltételek:** Post(A) állapota sikeres feldolgozás után  
Post(B) - // -
- **teljesítési feltétel:** Completion (javasolt végállapot,  
-ha a szándékok érvényesülnek)

Példa üzenet szemantikus leírására:

### **Advertise (A, B, M)**

1. A 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)))

Komisszívum, ha B = Facilitator,

akkor B azonos minden olyan ágenssel, amit a B tud.

Indulás – 1995/6

[www.fipa.org](http://www.fipa.org)



több tíz telecom cég, egyetem, ...

2005-től része IEEE Computer Society, IEEE CS Standard Group on ...  
**IEEE FIPA Standard Committee**

Szabvány kérdése - “gyors szabvány”  
- semmi megkötés ágensek belsejére

- **megkötés 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)

FIPA dokumentáció - összefoglaló, formális leírás, alkalmazási példák

**Mi nem kerül be a szabványba (magas absztrakció szinten, legfeljebb javaslat)**

Ami **nehezen formalizálható** (pl. security, mobility)

Ami még **nincs készen szabványosításra**

Ami **már megfelelően szabványosított**



## **Szabvány által lefedett témák (múlt és jelen):**

### **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

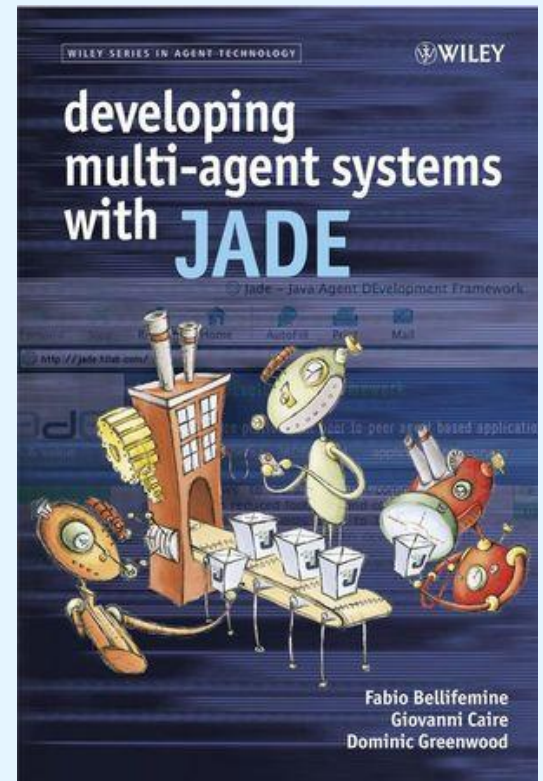
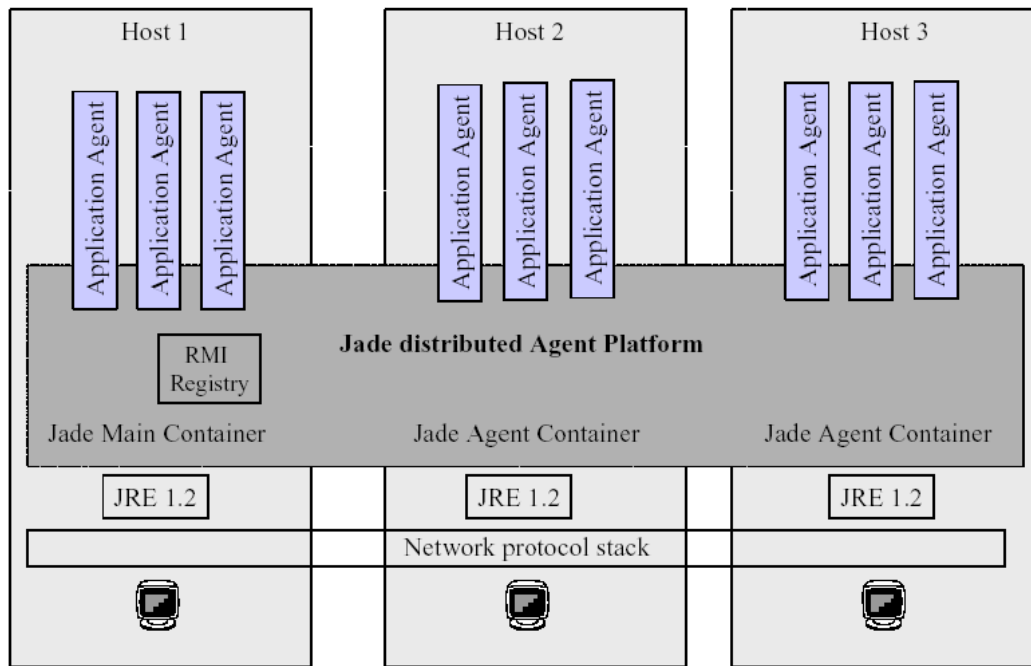
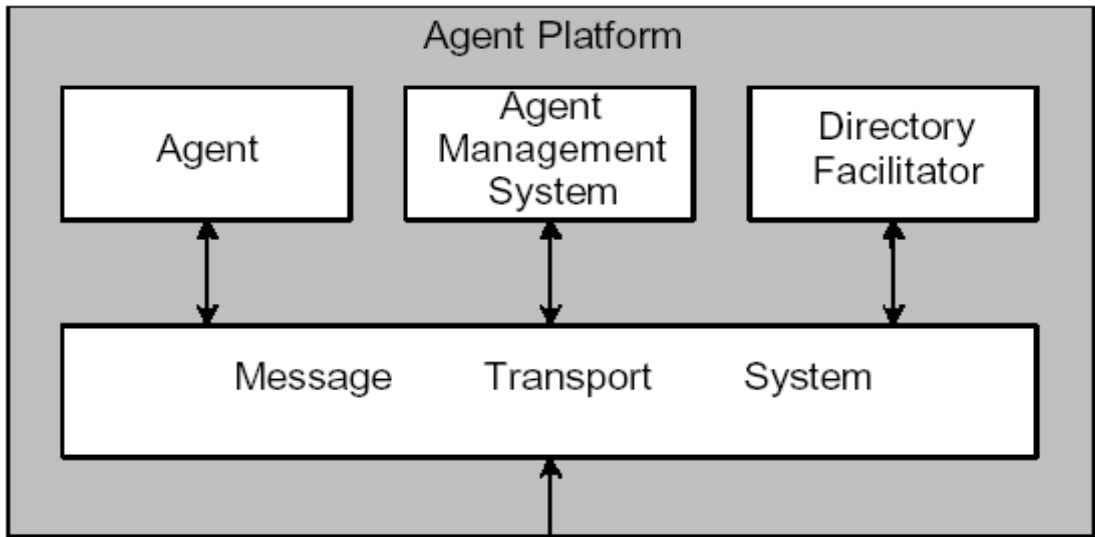
Message Buffering Service Specification

Messaging Interoperability Service Specification

Ontology Service Specification

### **Reference Applications**

Personel Travel Assistance, Personal Assistance, Nomadic Application Support, ...



# Abstract Architecture

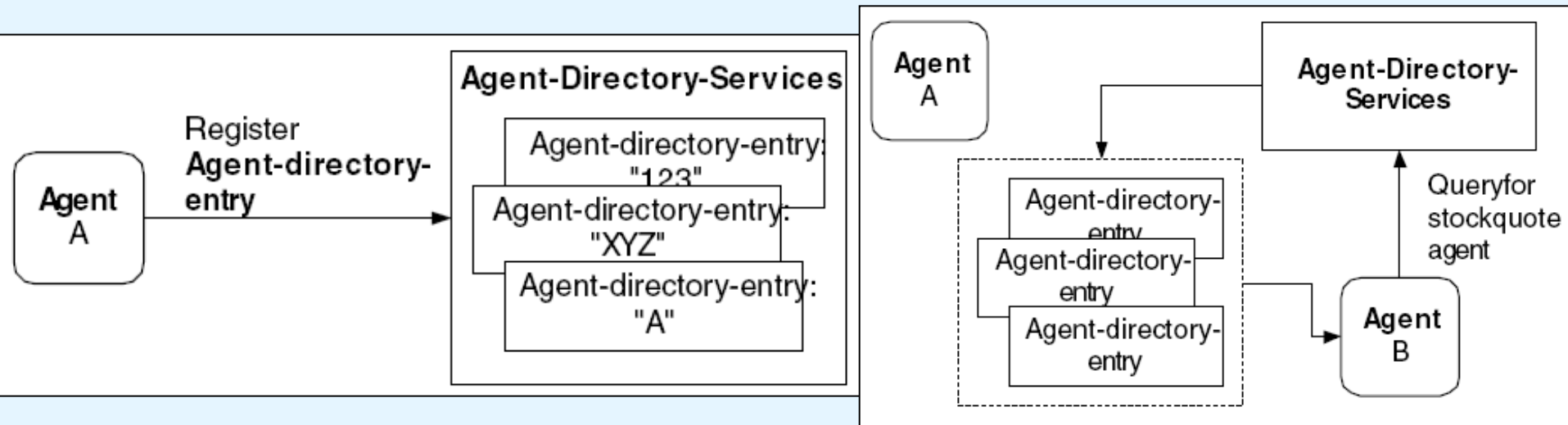
- Message Transport Interoperability
- Managing Multiple Message Transport Schemes
- Managing Message Encoding Schemes
- Locating Agents and Services via Directory Services

- Agent
  - ACL
  - Services

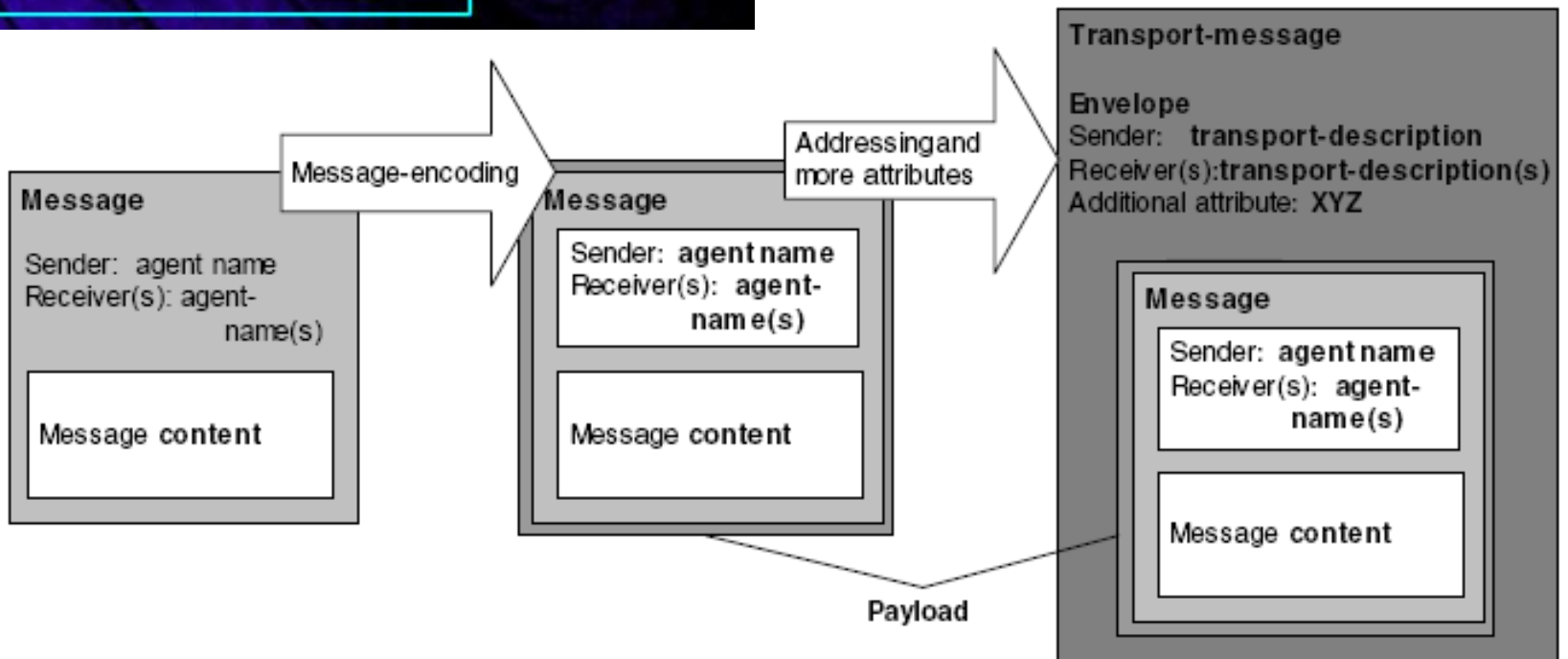
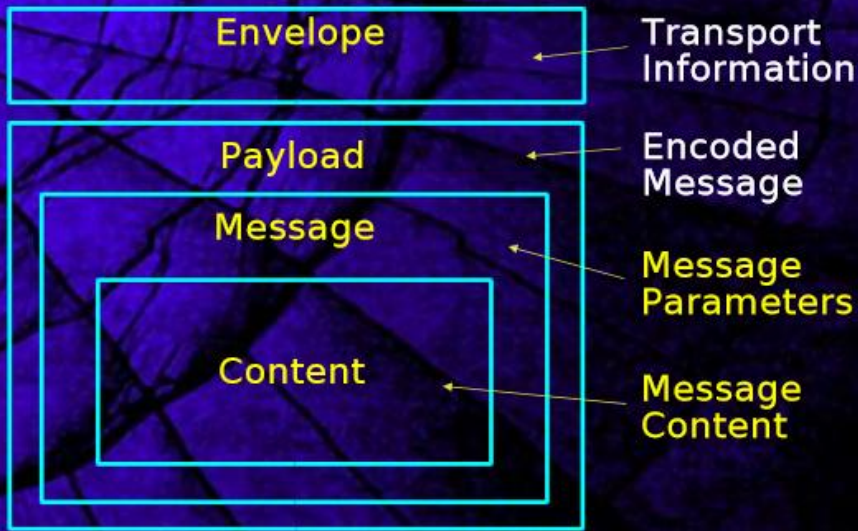
- Directory Services
  - Register, Modify, Delete, Query

- Encoding – Transport Services
  - Transform Encoding, Query Encoding, ....

- Message Transport Services
  - Bind Transport, Send/ Deliver Message



# - FIPA Message Structure -



# Agent Management

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

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

**White Pages** (ágens név – cím)

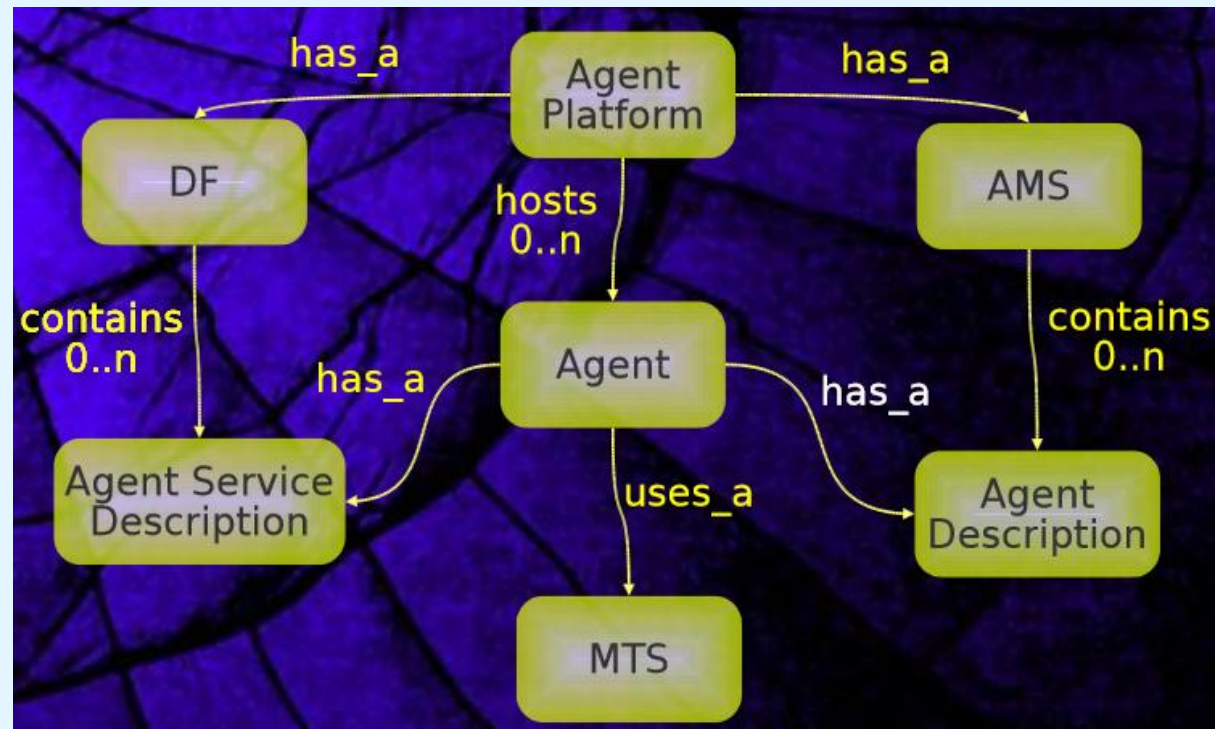
**AMS** – **Agent Management System** (agent name server), csak 1/ platform

(ágens köteles bejelentkezni)

**Yellow Pages** (ágens név – service server)

**DF** – **Directory Facilitator**, 1 vagy több (DF Federation)

(ágens nem köteles bejelentkezni)



## Agent Life Cycle

agent platform bounded  
application independent  
instance oriented  
unique

(pillanatnyilag abbamaradt)

### Active Agent

= MTS szállít feléje üzeneteket

### Initiated/Waiting/Suspended

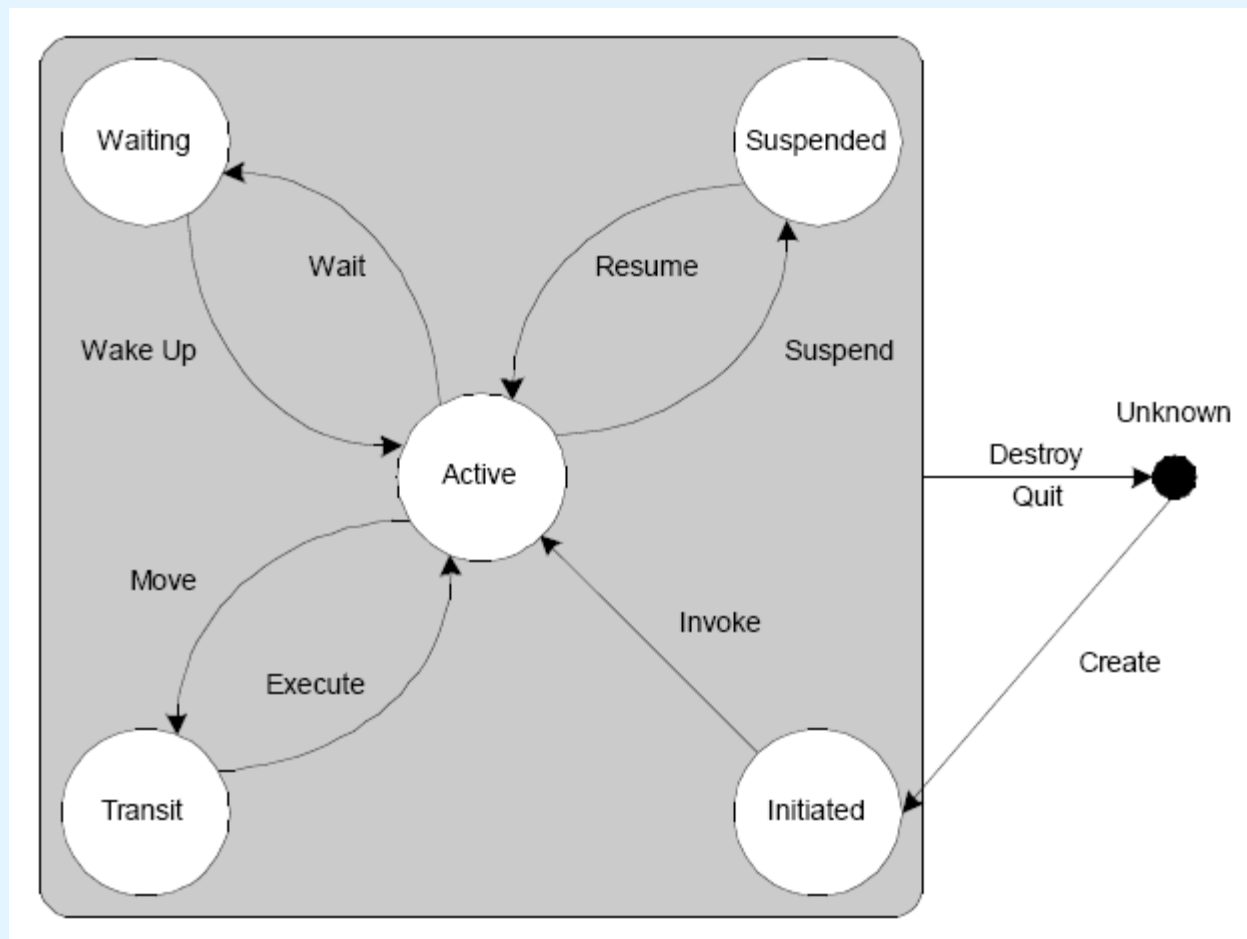
= üzenet puffereelés

### Transit

= üzenet puffereelés,  
v. forward

### AMS feladatai:

Suspend, Terminate,  
Create, Resume,  
Invoke, Execute  
resource management



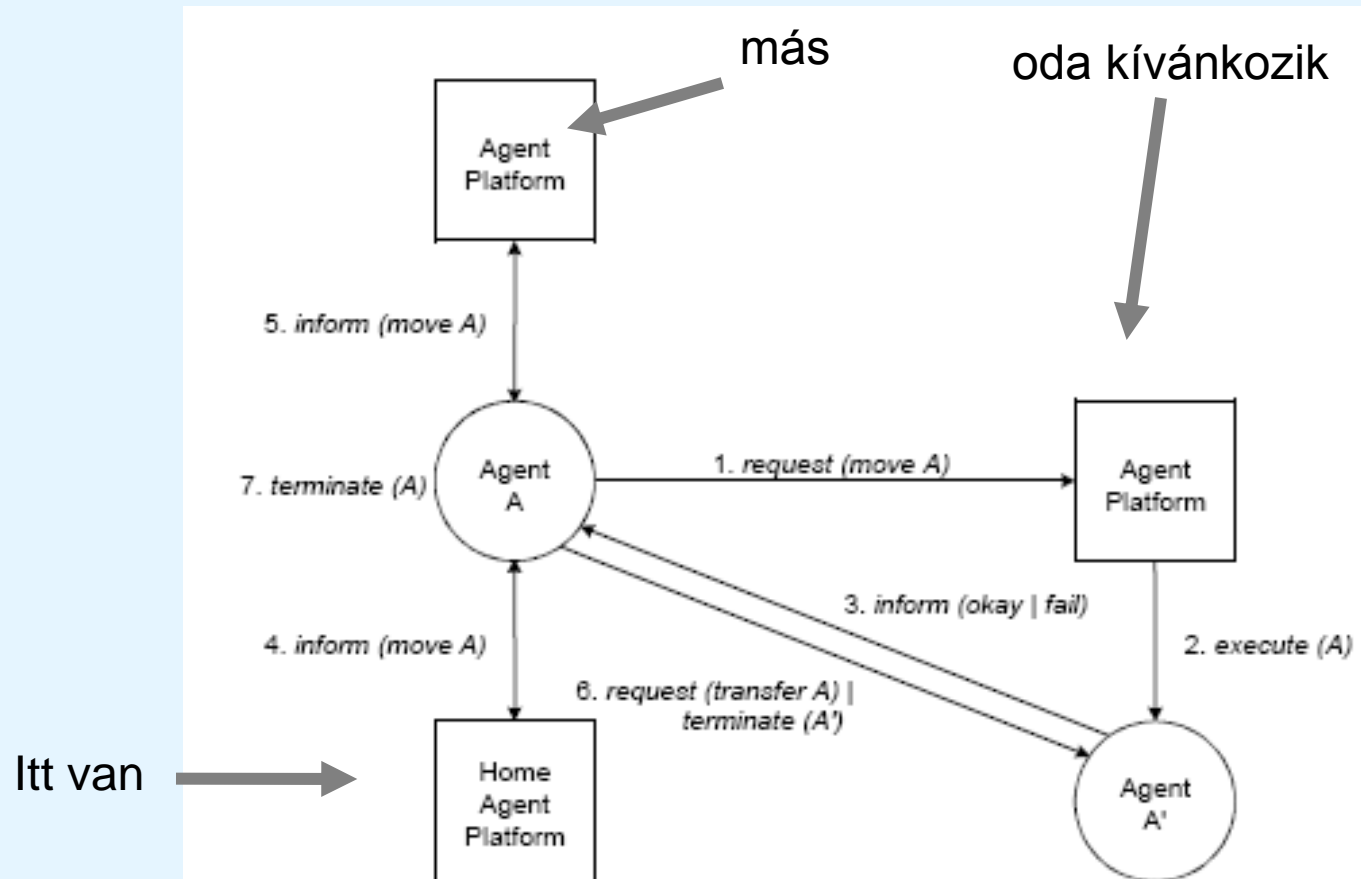
# Agent Management Support for Mobility Specification

Agent - moves code - new AP  
- transfers identity and authority

(pillanatnyilag abbamaradt)

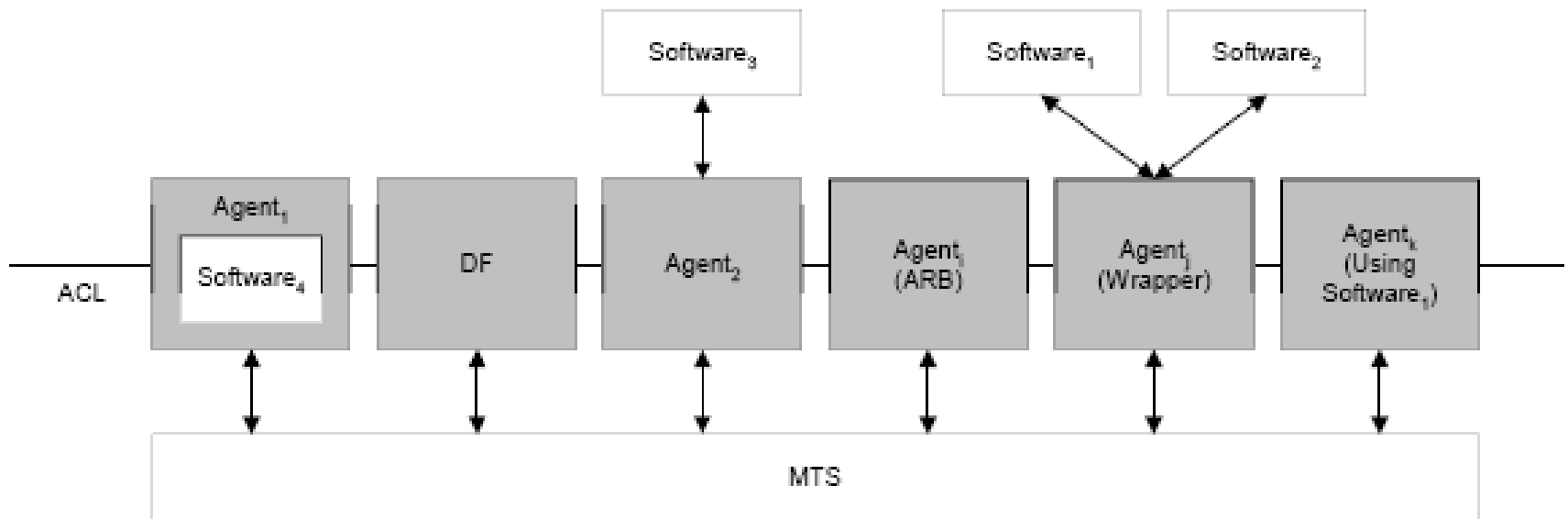
Mobility - Migration  $A \rightarrow A'$   
- Cloning  $A \rightarrow A'$   
- Invoking  $A \rightarrow B$

## Full migration









ARB regisztrálja magát DF-nél

WA regisztrálja magát DF-nél

Alkalmazói ágens DF-nél keresi az ARB-t

Alkalmazói ágens ARB-nél érdeklődik a keresett szoftverről

ARB megadja a szoftver paramétereit

Alkalmazói ágens DF-nél keresi a szoftverhez tartozó WA-ról

Alkalmazói ágens WA-nál szolgáltatást kér

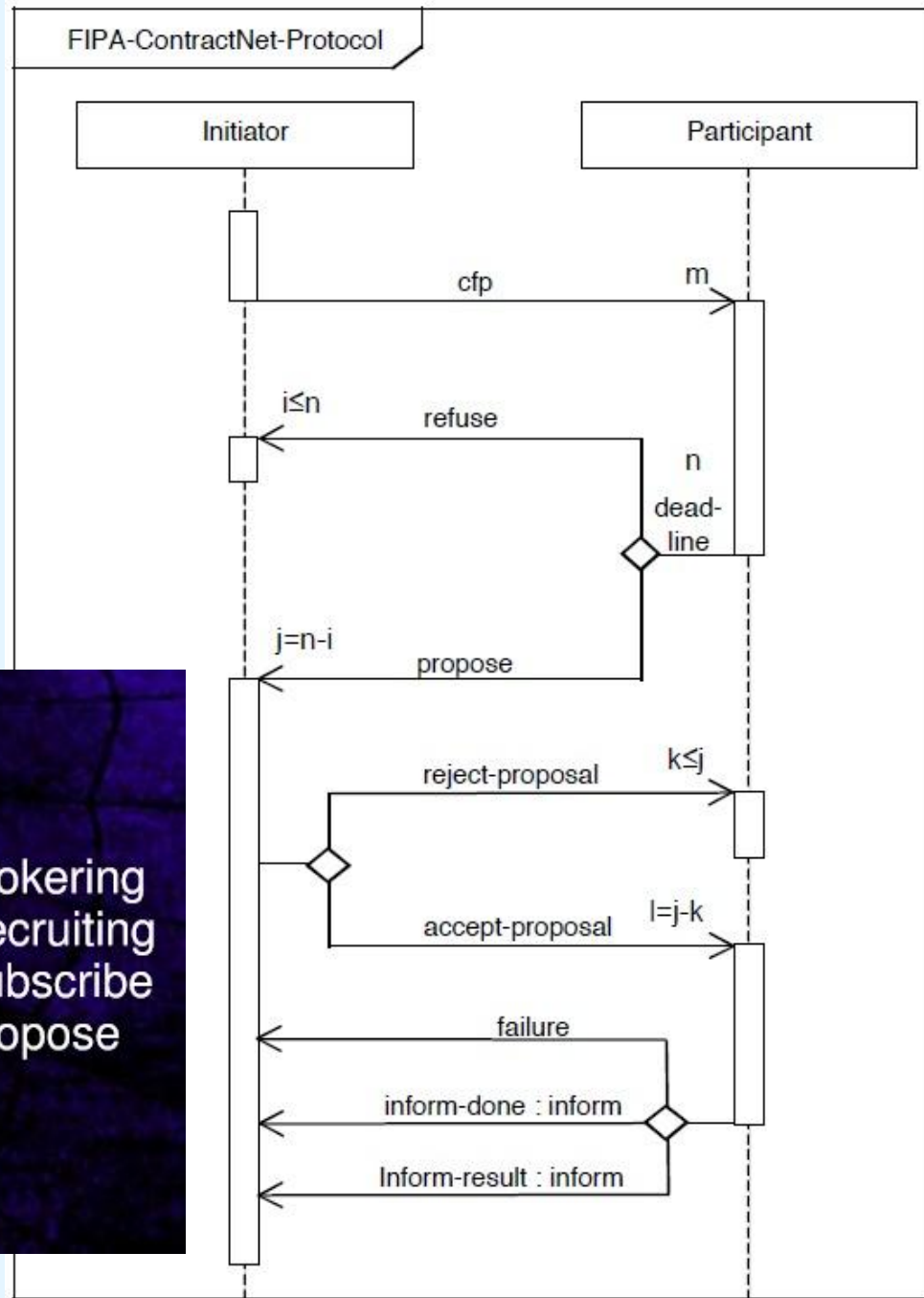
WA szoftver felé továbbítja a kérést és a paramétereiket

.....

**Agent Communication**  
**ACL Message Structure**  
**Message Transport**

**Interaction Protocol Library**  
**Specification**

**AUML – Agent-Based Extension**  
**to UML**



**- Interaction Protocols -**

*FIPA defined IPs are:*

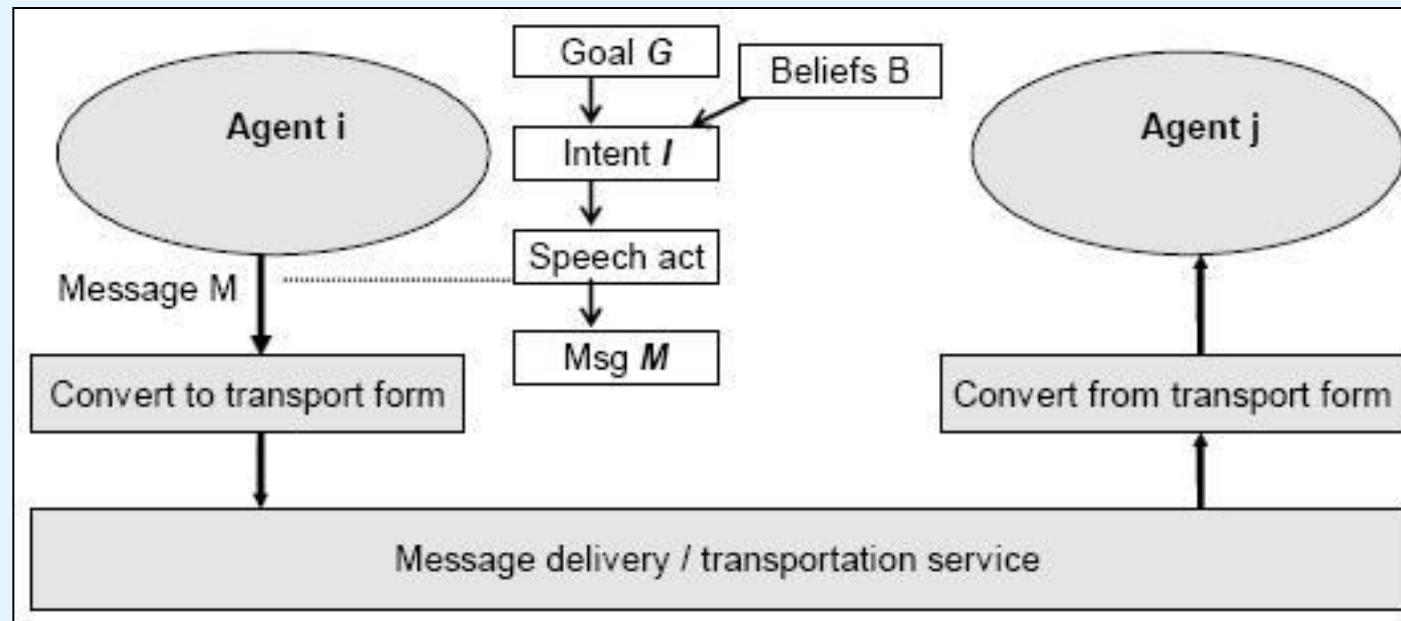
- FIPA-Request
- FIPA-Query
- FIPA-Request-When
- FIPA-Contract-Net
- FIPA-Iterated-Contract-Net
- FIPA-Auction-English
- FIPA-Auction-Dutch
- FIPA-Brokering
- FIPA-Recruiting
- FIPA-Subscribe
- FIPA-Propose

## Communicative Act Library Specification

**Accept Proposal, Agree, Cancel, Call for Proposals, Confirm, Disconfirm, Failure, Inform, Inform\_if, Inform\_Ref, Not Understood, Propagate, Propose, Proxy, Query If, Query Ref, Refuse, Reject Proposal, Request, Request When, Request Whenever, Subscribe**

### Inclusion Criteria

- Summary of Semantic Force and Content Type
- Detailed Natural Language Description
- **Formal Model in SL (Acts's Semantics, FP, RE)**
- Examples of Usage
- Substantial and Clear Documentation
- Utility Made Clear



# Semantic Language SL

(**not** <Wff>)

(**and** <Wff0> <Wff1>) (**or** <Wff0> <Wff1>)

(**implies** <Wff0> <Wff1>) (**equiv** <Wff0> <Wff1>)

(**forall** <var> <Wff>) (**exists** <var> <Wff>)

**B** <agent> <expr>

**U** <agent> <expr>

**I** <agent> <expr>

szándéka van és tervezt

**PG** <agent> <expr>

tartós célja van, de nem szükségk. tervezt

(**feasible** <actexpr> <Wff>) igaz, hogy cselekvés megtörténthet  
közvetlenül utána Wff igaz lesz

(feasible <actexpr>) = (feasible <actexpr> true)

(**done** <actexpr> <Wff>) igaz, hogy cselekvés épp megtörtént  
és előtte Wff igaz volt

(done <actexpr>) = (done <actexpr> true)

(**iota** x (P x)) pont olyan x, amire igaz P(x)

(**any** <term> <formula>) akármilyen objektum, ami a formulát teljesíti

(**all** <term> <formula>) minden objektum, ami ...

$$\mathbf{B}_i \phi = \mathbf{B}_i \phi \vee \mathbf{B}_i \neg \phi, \quad \mathbf{A}_{b_1 \dots b_n} \phi = \mathbf{B}_i \mathbf{B}_j \mathbf{B}_i \dots \phi$$

**Alap: Confirm  
Disconfirm  
Inform  
Request**

## **Accept-Proposal**

$\langle i, \text{accept-proposal } (j, \langle j, \text{act} \rangle, \phi) \rangle = \langle i, \text{inform } (j, \text{li Done } (\langle j, \text{act} \rangle, \phi)) \rangle$

FP:  $B_i \alpha \wedge \neg B_i (B_{ifj} \alpha \vee U_{ifj} \alpha)$

RE:  $B_j \alpha$

$\alpha = \text{li Done } (\langle j, \text{act} \rangle, \phi)$

## **Confirm**

$\langle i, \text{confirm } (j, \phi) \rangle$

FP:  $B_i \phi \wedge B_i U_j \phi$

RE:  $B_j \phi$

## **Disconfirm**

$\langle i, \text{disconfirm } (j, \phi) \rangle$

FP:  $\neg B_i \phi \wedge B_i (U_j \phi \vee B_j \phi)$

RE:  $\neg B_j \phi$

## **Cancel**

$\langle i, \text{cancel } (j, a) \rangle = \langle i, \text{disconfirm } (j, \text{li Done } (a)) \rangle$

FP:  $\neg \text{li Done } (a) \wedge B_i (B_j \text{li Done } (a) \vee U_j \text{li Done } (a))$

RE:  $B_j \neg \text{li Done } (a)$

.....