سیستم‌های چندعاملی

درس ۱۲

برقراری ارتباط

Communication

کاظم فولادی قلعه
دانشکده مهندسی، دانشکدگان فارابی
دانشگاه تهران

http://courses.fouladi.ir/mas

عامل‌های برقرارکننده‌ی ارتباط

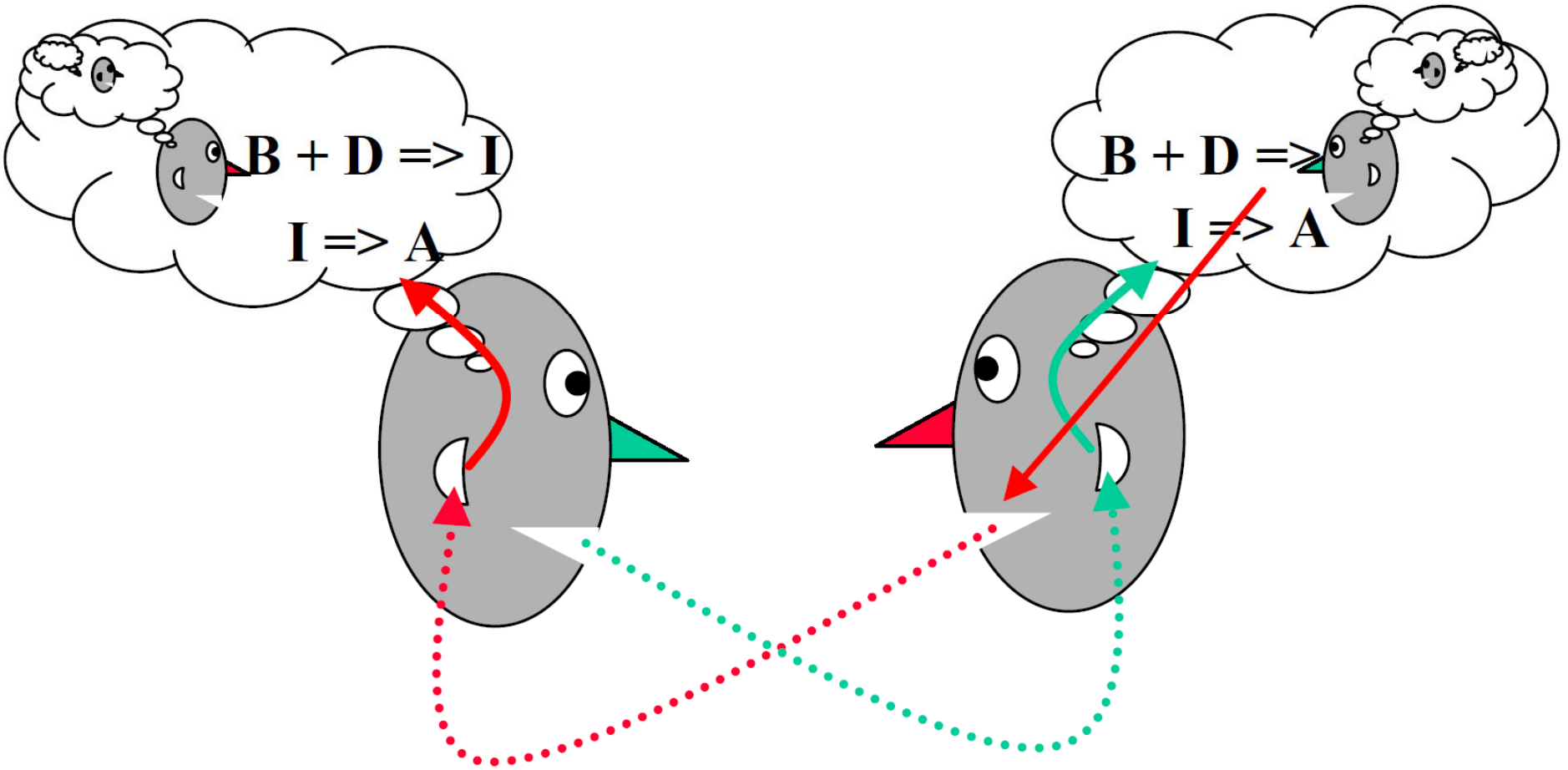
COMMUNICATING AGENTS

اندرکنش‌های چندعاملی معمولاً با نوعی از **برقراری ارتباط** وابستگی دارد.

عامل‌ها می‌توانند برای مقاصد مختلفی با هم **ارتباط** برقرار کنند:

- برای حل یک مسئله به صورت همکارانه
- برای مذاکره با هم
- برای تبادل دانایی با یکدیگر
- ...

BDI Model and Communication



سطوح برقراری ارتباط

سطوح انتزاعی برقراری ارتباط برای عامل‌های محاسباتی

COMMUNICATION LEVELS

سطح کاربرد
The Application Level

سطح زبان
The Language Level

سطح شبکه
The Network Level

کنش‌های برقراری ارتباط

COMMUNICATIVE ACTS

یک کنش ابتدایی ارتباطی،
می‌تواند به صورت یک **کنش** در نظر گرفته شود که
دانایی یک عامل را در مورد حالت فعلی را تغییر می‌دهد.

برای مثال، در معمای کلاه‌ها، اگر عامل ۱ بگوید
«کلاه عامل ۳ قرمز است»
مجموع اطلاعات (دانایی) عامل ۳ از $\{a,b\}$ به $\{a\}$ تغییر می‌کند.

از کنش‌های ابتدایی ارتباطی، معمولاً با عنوان کنش‌های برقراری ارتباط یا کنش‌های گفتاری *(Speech Acts)* یاد می‌شود.

کنش‌های برقراری ارتباط

انواع کنش‌های برقراری ارتباط

TYPES OF COMMUNICATIVE ACTS

اطلاع دادن ***Informing***	«**من نمی‌دانم رنگ کلاهم چیست**» (مثلاً در معمای کلاه‌ها)
پرس‌وجو ***Querying***	«**آیا تو رنگ کلاهت را می‌دانی؟**» (مثلاً در معمای کلاه‌ها)
تعهد دادن ***Committing***	برای یک کنش خاص «**من از چهارراه عبور خواهم کرد**» (مثلاً در هماهنگی با روش حذف متغیر)
نهی کردن ***Prohibiting***	از یک کنش «**تو نباید از چهارراه عبور کنی**» (مثلاً در انتساب نقش‌ها)
امر کردن ***Directing***	در مورد یک کنش «**عبور کن!**» (مثلاً در یک ساختار سازمانی)

کنش‌های گفتاری

SPEECH ACTS

نظریه‌های «کنش گفتاری» از نظریه‌های **پراگماتیک** زبان هستند:
تئوری‌های مربوط به چگونگی **استفاده** از زبان:

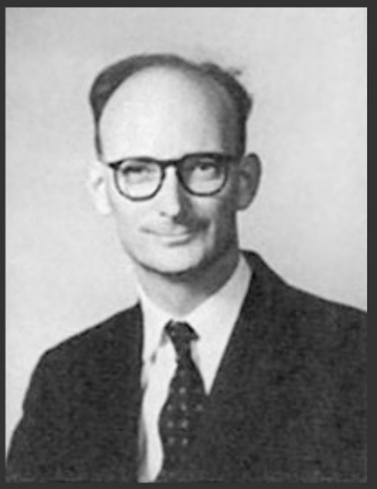
زبان چگونه به‌صورت روزمره توسط افراد برای رسیدن به اهداف و قصدها استفاده می‌شود؟

آستن (۱۹۶۲)	برخی از گفته‌ها *(utterance)* مشابه کنش‌های فیزیکی **حالت دنیا را تغییر می‌دهند.**
سرل (۱۹۶۹)	**هر چیزی** که گفته می‌شود، با قصد برآورده شدن یک هدف یا قصد بیان می‌شود.

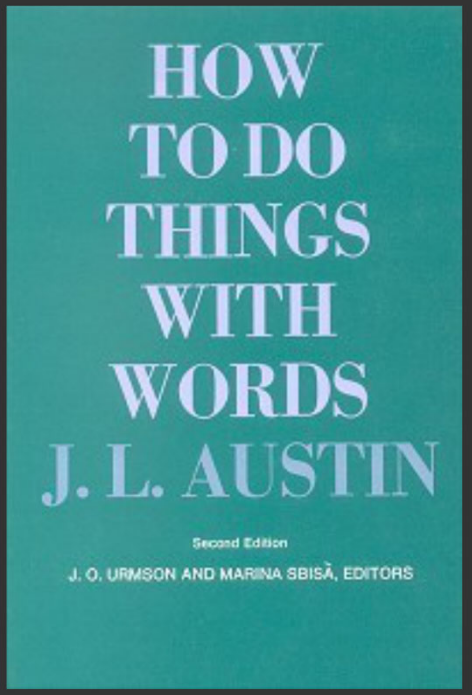
نظریه‌ی کنش‌های گفتاری

چگونه از گفته‌ها برای رسیدن به قصدها استفاده می‌شود؟

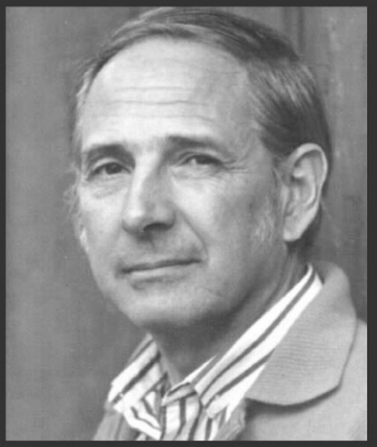
John Langshaw Austin



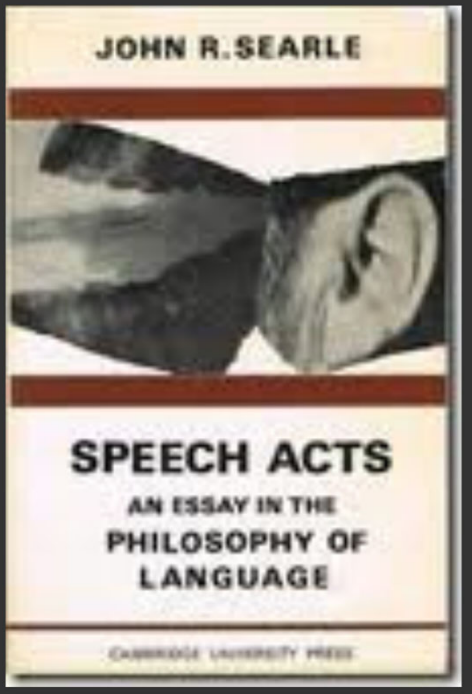
in 1962, published the book:



John R. Searle



in 1969, published the book:



کنش‌های گفتاری

انواع کنش‌های گفتاری از نگاه سرل

SPEECH ACTS

بازنماینده *Representatives*	اطلاع‌رسانی در مورد چیزی (مانند: بارن می‌بارد)
رهنمودی *Directives*	سعی دارد شنونده کاری را انجام دهد (مانند: لطفا غذا درست کن)
تعهدی *Commisives*	گوینده به چیزی تعهد می‌دهد (مانند: من قول می‌دهم به میهمانی بروم)
بیانی *Expressives*	گوینده یک حالت ذهنی را بیان می‌کند (مانند: خوشحال هستم، متشکرم)
اعلانی *Declarations*	نام‌گذاری یا اعلان یک واقعه (مانند: اعلان جنگ)

در مورد اینکه این تیپولوژی کنش‌های گفتاری مناسب است یا نه، تردید وجود دارد!

کنش‌های گفتاری

مؤلفه‌های عمومی

SPEECH ACTS

فعل اجرایی *Performative Verb*	مانند: درخواست، اطلاع دادن، ...
محتوای گزاره‌ای *Propositional Content*	مانند: بیان وضعیت (مثلاً: در بسته است.)

هر سه جمله، حاصل یک محتوای گزاره‌ای هستند، اما با فعل‌های اجرایی متفاوت	**“The door is closed”** **request** speech act = “please close the door” **inform** speech act = “the door is closed!” **inquire** speech act = “is the door closed?”

معناشناسی مبتنی بر پلن

PLAN-BASED SEMANTICS

معناشناسی کنش‌های گفتاری

چگونه می‌توان معنای یک کنش گفتاری را تعریف کرد؟

یکی از روش‌ها، استفاده از زبان استاندارد طرح‌ریزی است.

(فقط باید توجه کرد که گوینده همیشه نمی‌تواند
شنونده را مجبور کند که یک حالت ذهنی مطلوب را بپذیرد.)

معناشناسی مبتنی بر پلن

مثال

PLAN-BASED SEMANTICS

The semantics for "request"

request(s, h, φ)

precondition:

- *s believes h can do φ*

(you don't ask someone to do something unless you think they can do it)

- *s believe h believe h can do φ*

(you don't ask someone unless they believe they can do it)

- *s believe s want φ*

(you don't ask someone unless you want it!)

post-condition:

- *h believe s believe s want φ*

(the effect is to make them aware of your desire)

زبان‌های برقراری ارتباط عامل‌ها

AGENT COMMUNICATION LANGUAGES (ACL)

ACL

*ACL*ها فرمت‌های استاندارد برای تبادل پیام هستند.

زبان‌های برقراری ارتباط عامل‌ها

KQML

AGENT COMMUNICATION LANGUAGES (ACL)

ACL

*ACL*ها فرمت‌های استاندارد برای تبادل پیام هستند.

یک نمونه‌ی خاص از *ACL*:

KQML

از دو بخش تشکیل شده است:

زبان پرس‌وجو و دستکاری دانایی	فرمت تبادل دانایی
Knowledge Query and Manipulation Language (KQML)	***Knowledge Interchange Format (KIF)***

زبان‌های برقراری ارتباط عامل‌ها

KQML

AGENT COMMUNICATION LANGUAGES (ACL)

زبان پرس‌وجو و دستکاری دانایی

Knowledge Query and Manipulation Language (KQML)

یک زبان بیرونی *(outer)*

«فعل‌های ارتباطی» (اجرایی) قابل قبول متعددی را تعریف می‌کند.

کاری با **محتوا** و **مضمون** پیام ندارد.

Example **performatives**

- **ask-if** (‘is it true that. . . ’)
- **perform** (‘please perform the … ‘)
- **tell** (‘it is true that. . . ’)
- **reply** (‘the answer is . . . ’)

فرمت تبادل دانایی

Knowledge Interchange Format (KIF)

زبانی برای بیان **محتوای** پیام یا **دانایی دامنه**

قابل استفاده برای نوشتن **آنتولوژی‌ها**

(بر مبنای منطق مرتبه اول)

زبان‌های برقراری ارتباط عامل‌ها

KQML: مثال

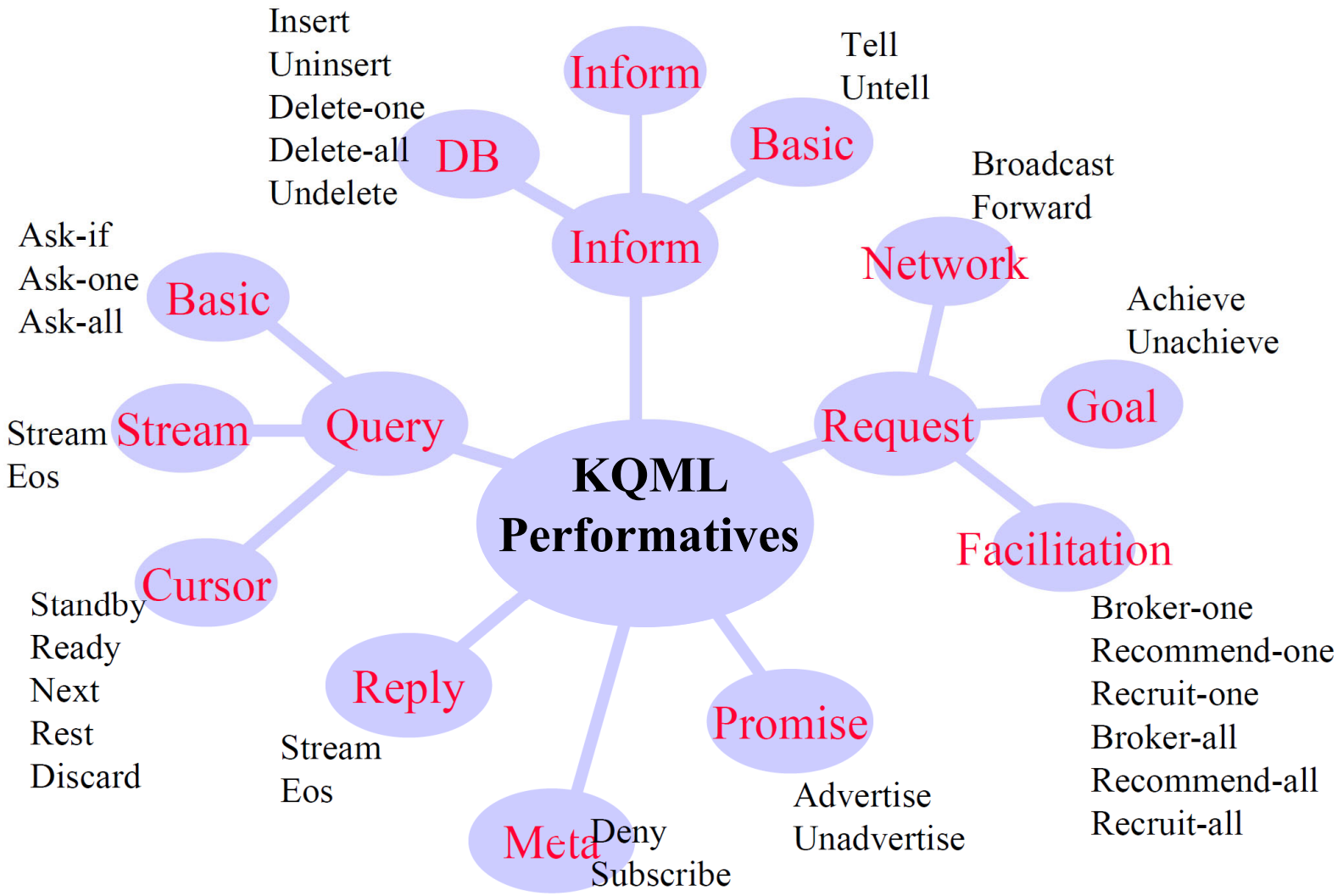
برای مثال،
یک روبات فوتبالیست که می‌خواهد از سایر روبات‌ها بپرسد که «توپ کجاست»،
می‌تواند از پیام *KQML* زیر استفاده کند:

```
(ask-all
   :sender  robot-A
   :content  (BALL ?position)
   :language  PROLOG
   :ontology  ROBOT-SOCCER
   :reply-with  identifier-X
)
```

زبان‌های برقراری ارتباط عامل‌ها

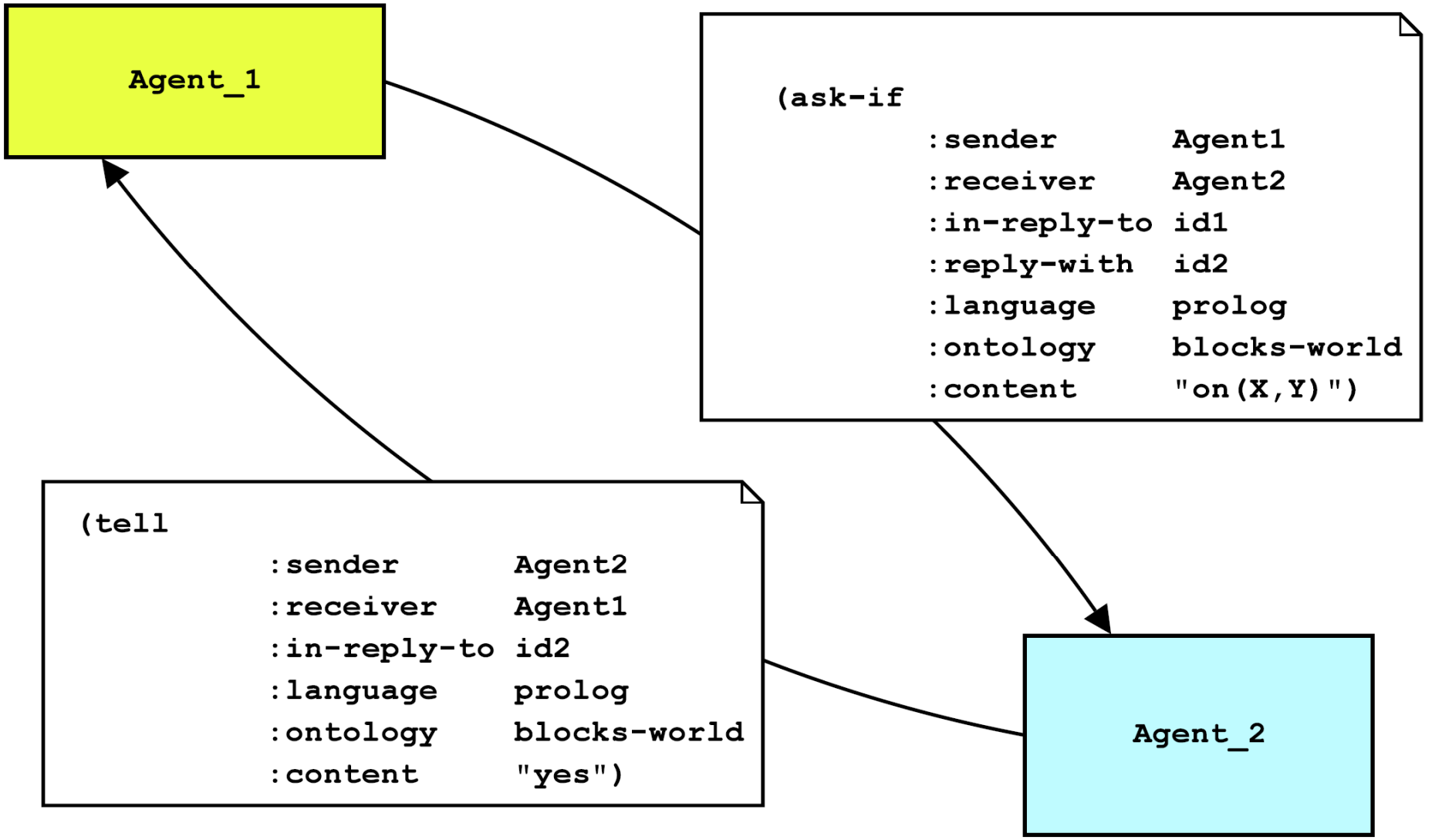
KQML: فعل‌های اجرایی

KQML PERFORMATIVES



زبان‌های برقراری ارتباط عامل‌ها

مثال: KQML



زبان‌های برقراری ارتباط عامل‌ها

مثال: KQML و KIF

```
(stream-about
   :sender       A
   :receiver     B
   :language     KIF
   :ontology     motors
   :reply-with   q1
   :content m1
)
```

A asks B for information about motor 1, using the ontology (represented in KIF) about motors.

```
(tell
   :sender       B
   :receiver     A
   :in-reply-to q1
   :content
    (= (torque m1) (scalar 12 kgf))
)
(tell
   :sender       B
   :receiver     A
   :in-reply-to q1
   :content
     (= (status m1) normal)
)
(eos
   :sender       B
   :receiver     A
   :in-reply-to q1
)
```

B responds to A's query q1. Two facts are sent:
1) that the torque of motor 1 is 12kgf; and
2) that the status of the motor is normal.

The ask stream is terminated using the eos performative.

آنتولوژی‌ها

ONTOLOGIES

برای اینکه عامل‌ها بتوانند با هم ارتباط برقرار کنند،
باید بر روی **کلمات** (و اصطلاحاتی) که برای توصیف یک دامنه استفاده می‌کنند،
توافق داشته باشند.

⇓

نیاز به **آنتولوژی**

آنتولوژی (هست‌شناسی)
Ontology

مشخص‌سازی صوری مجموعه‌ای از اصطلاحات

اغلب مشکل وجود زبان‌های متعدد، یک دغدغه است.

در *KSE* تلاش‌های بسیاری برای تعریف آنتولوژی‌های مشترک و متداول صورت گرفته است.

زبان‌های برقراری ارتباط عامل‌ها

KQML: مشکلات

مشکلات *KQML*

مجموعه‌ی **فعل‌های اجرایی پایه‌ی** *KQML* شناور است.
(پیاده‌سازی‌های مختلف نمی‌توانند با هم کار کنند.)

مکانیسم‌های انتقال پیام دقیقاً تعریف نشده است.
(مجدد مشکل کار کردن پیاده‌سازی‌های مختف با هم وجود دارد.)

معناشناسی *KQMK* به‌صورت قاطع تعریف نشده است.
(ابهام منجر به مشکل در کار کردن پیاده‌سازی‌های ناهمتا می‌شود.)

هیچ نوع فعل **تعهدی** در این زبان وجود ندارد.
(در صورت نبود توانایی برای تعهد به یک وظیفه، عامل‌ها نمی‌توانند رفتارشان را هماهنگ کنند.)

مجموعه‌ی فعل‌های اجرایی عمدتاً تک‌کاره (*ad-hoc*) و بسیار بزرگ هستند.
(مشکل در مدیریت و استفاده از آنها)

زبان‌های برقراری ارتباط عامل‌ها

FIPA ACL

AGENT COMMUNICATION LANGUAGES (ACL)

ACL

*ACL*ها فرمت‌های استاندارد برای تبادل پیام هستند.

یک نمونه‌ی جدیدتر از *ACL*:

FIPA ACL

Foundation for Intelligent Physical Agents (FIPA)

ساختار پایه: مشابه *KQML*

* تعداد فعل‌های اجرایی به ۲۰ مورد محدود شده است.
* با استفاده از زبان *SL* یک معناشناسی صوری برای این زبان تعریف شده است

(زبان *SL* می‌تواند باورها، مطلوب‌ها و باورهای نامطمئن را همانند کنش‌ها بازنمایی کند.)

زبان‌های برقراری ارتباط عامل‌ها

مثال: FIPA ACL

FIPA ACL example

```
(inform
     :sender     agent1
     :receiver   agent5
     :content    (price good200 150)
     :language   sl
     :ontology   hpl-auction
)
```

زبان‌های برقراری ارتباط عامل‌ها

FIPA ACL: فعل‌های اجرایی

FIPA ACL Performatives

performative	passing info	requesting info	negotiation	performing actions	error handling
accept-proposal			x		
agree				x	
cancel		x		x	
cfp			x		
confirm	x				
disconfirm	x				
failure					x
inform	x				
inform-if	x				
inform-ref	x				
not-understood					x
propose			x		
query-if		x			
query-ref		x			
refuse				x	
reject-proposal			x		
request				x	
request-when				x	
request-whenever				x	
subscribe		x			

زبان‌های برقراری ارتباط عامل‌ها

FIPA ACL: فعل‌های اجرایی پایه: «اطلاع‌دادن» و «درخواست»

فعل‌های اجرایی پایه
اطلاع‌دادن *Inform*
درخواست *Request*

سایر فعل‌های اجرایی، بر حسب این دو مورد تعریف می‌شوند.

معنای این دو فعل اجرایی، در دو بخش تعریف می‌شود:

پیش‌شرط *Pre-condition*	چه چیزی باید درست باشد تا کنش گفتاری موفق شود؟
اثر رسیونال *Rational Effect*	فرستنده‌ی پیام امیدوار است چه اتفاقی بیفتد؟

زبان‌های برقراری ارتباط عامل‌ها

FIPA ACL: فعل‌های اجرایی پایه: «اطلاع‌دادن» و «درخواست»: مشخصات

فعل‌های اجرایی پایه

اطلاع‌دادن
Inform

FIPA "Inform" Performative

The **content** is a statement.
The **pre-condition** is that the sender:
- holds that the content is true;
- intends that the recipient believe the content;
- does not already believe that the recipient is aware of whether content is true or not

Note that the speaker only has to believe that what he says is true.

درخواست
Request

FIPA "Request" Performative

The **content** is an action.
The **pre-condition** is that the sender:
- intends action content to be performed;
- believes recipient is capable of performing this action;
- does not believe that sender already intends to perform action

The last captures the fact that you don't speak if you don't need to.

ارزش برقراری ارتباط

THE VALUE OF COMMUNICATION

هر عامل در یک حالت خاص با این دو پرسش مواجه می‌شود:

پرسش:
از **چه کنشی** باید استفاده کند؟
با **چه کسی** باید ارتباط برقرار کند؟

یک **مقدار** را به عنوان شاخصی برای میزان خوبی کنش،
به یک کنش برقراری ارتباط نسبت دهیم (مانند مقادیر Q).

$$\Downarrow$$

در هر موقعیت، هر عامل باید کنش ارتباطی با **بالاترین ارزش** را انتخاب کند.

اما، این **مقادیر ارزش** باید از کجا بیایند؟

بازی‌های بیزی

BAYESIAN GAMES

بازی بیزی
Bayesian Game

یک **بازی بیزی** یک بازی استراتژیک تحت مشاهده‌پذیری جزئی است.

۱. $n > 1$ عامل در دنیا وجود دارد.

۲. مجموعه‌ی S از حالت‌های دنیا وجود دارد.

۳. برای هر عامل i یک افراز $\mathcal{P}_i$ از حالت‌های دنیا وجود دارد.

۴. هر حالت $s \in S$ یک بازی استراتژیک G_s با میزان سود *(payoff)* متناظر تعریف می‌کند.

۵. هر حالت با یک احتمال (پیشین) اتفاق می‌افتد که برای همه‌ی عامل‌ها مساوی است.

پرسش:

عامل‌ها در یک بازی بیزی باید چه کنش‌هایی را انتخاب کنند؟

بازی‌های بیزی

مثال

BAYESIAN GAMES

	L	R
U	$2, 2$	$1, 0$
D	$0, 1$	$0, 0$

G_a (probability 0.7)

	L	R
U	$0, 0$	$0, 1$
D	$-1, 0$	$2, 2$

G_b (probability 0.3)

$$n = 2, \quad S = \{a, b\}, \quad \mathcal{P}_1 = \{\{a\}, \{b\}\}, \text{ and } \mathcal{P}_2 = \{\{a, b\}\}$$

$$P(s = a) = 0.7$$
$$P(s = b) = 0.3$$

کنش‌های عامل‌ها باید وابسته به حالتی باشد که واقعاً تحقق یافته است.

بازی‌های بیزی و برقراری ارتباط

BAYESIAN GAMES AND COMMUNICATION

	L	R
U	$2,2$	$1,0$
D	$0,1$	$0,0$

G_a (probability 0.7)

	L	R
U	$0,0$	$0,1$
D	$-1,0$	$2,2$

G_b (probability 0.3)

این بازی بیزی دارای تعادل نش (U,L) است.

این انتخاب از کنش توأم، به هریک از عامل‌ها پی‌آف مورد انتظار زیر را می‌دهد:

$$0.7 * 2 + 0.3 * 0 = 1.4.$$

حال، اگر در هر $s \in S$ عامل ۱ به عامل ۲ در مورد s اطلاع بدهد،
افراز عامل ۲ به صورت مقابل تغییر می‌کند:

$$\mathcal{P}_2 = \{\{a\}, \{b\}\}$$

حال عامل‌ها می‌توانند مقدار صحیح تعادل را برای هر s محاسبه کند:
در $s = a$ آنها (U,L) را انتخاب می‌کنند و در $s = b$ آنها (D,R) را انتخاب می‌کنند.

پی‌آف مورد انتظار جدید می‌شود:

$$0.7 * 2 + 0.3 * 2 = 2.$$

ارزش این کنش برقراری ارتباط = $2 - 1.4 = 0.6$

هماهنگ‌سازی از طریق برقراری ارتباط

COORDINATION VIA COMMUNICATION

تاکنون الگوریتم‌های هماهنگی متعددی را برای عامل‌های همکار دیده‌ایم.

این الگوریتم‌ها بر برخی فرض‌های **دانایی مشترک** تکیه می‌کردند.

این الگوریتم‌ها می‌توانند تغییر پیدا کنند
تا به صورت کاملاً صریح، **ارتباطات** را به حساب آورند.

در صورت برقراری ارتباط، نیاز به **مقدار کمتری** دانایی مشترک وجود دارد.

برقراری ارتباط، به یک عامل اجازه می‌دهد که
تنها کنش بهینه‌ی **خود** را در نقطه‌ی تعادل محاسبه کند.

قراردادهای اجتماعی از طریق برقراری ارتباط

SOCIAL CONVENTIONS VIA COMMUNICATION

For each agent i in parallel

If $i \neq 1$

Wait until all actions $(a_1^*, \ldots, a_{i-1}^*)$ are received.

End

Compute an equilibrium that contains $(a_1^*, \ldots, a_{i-1}^*, \ldots)$.

Choose component a_i^* from this equilibrium.

Broadcast action a_i^* to all agents $i+1, \ldots, n$.

End

انتساب نقش‌ها از طریق برقراری ارتباط

الگوریتم حریصانه‌ی توزیع‌شده

ROLE ASSIGNMENT VIA COMMUNICATION

For each agent i in parallel
  $I = \{\}$.
  For each role $j = 1, \ldots, n$
    Compute the potential r_{ij} of agent i for role j.
    Broadcast r_{ij} to all agents.
  End
  Wait until all $r_{i'j}$, for $i' \notin I$ and $j = 1, \ldots, n$, are received.
  For each role $j = 1, \ldots, n$
    Assign role j to agent $i^* = \arg\max_{i'} \{r_{i'j}\}$.
    Add i^* to I.
  End
End

حذف متغیر بدون برقراری ارتباط

با هدف یافتن تعادل نش بهینه‌ی پارتو است که u (سودمندی سراسری) را ماکزیمم می‌کند

VARIABLE ELIMINATION WITHOUT COMMUNICATION

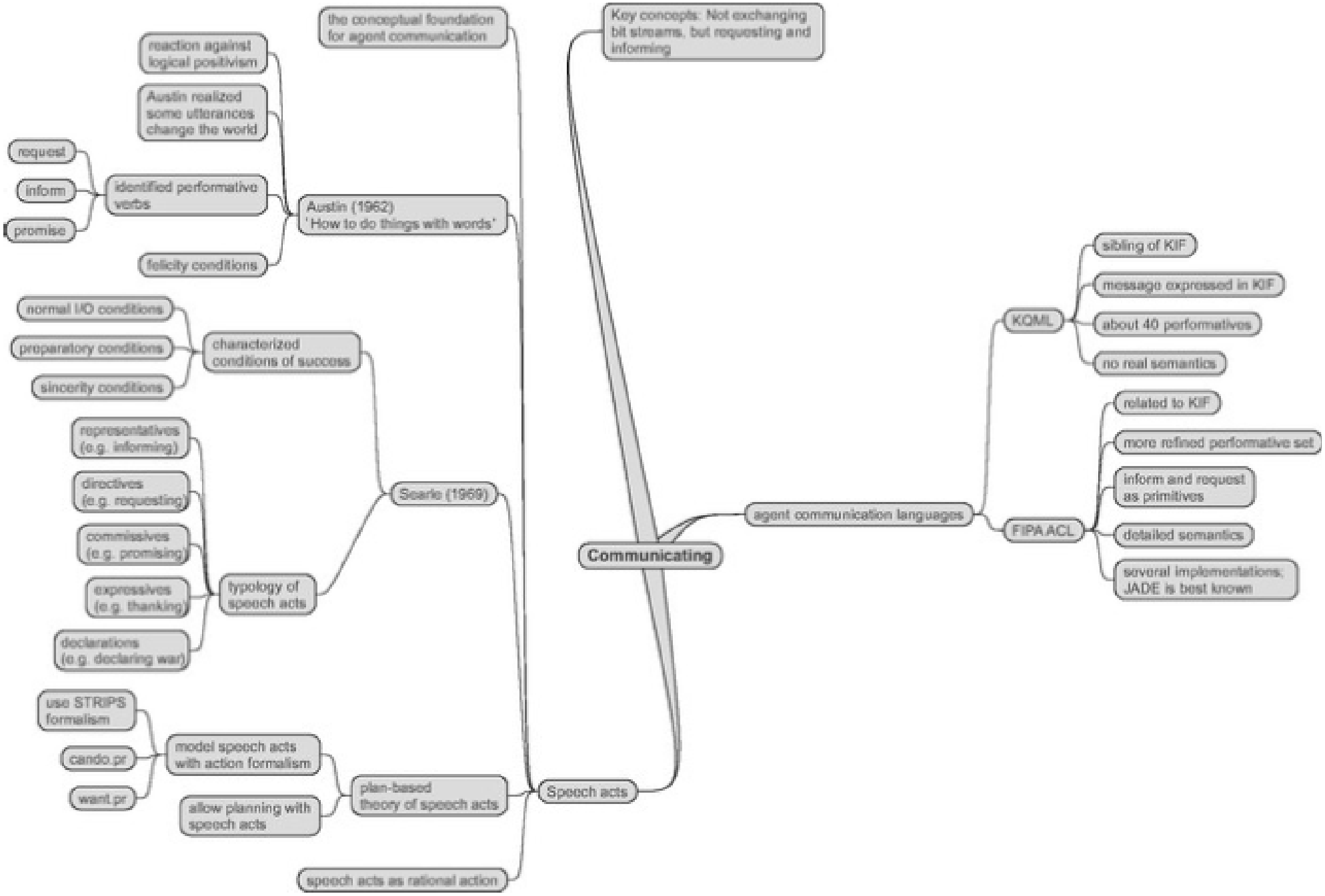
For each agent in parallel
  $F = \{f_1, \ldots, f_k\}$.
  For each agent $i = 1, 2, \ldots, n$
    Find all $f_j(a_i, a_{-i}) \in F$ that involve a_i.
    Compute $B_i(a_{-i}) = \arg\max_{a_i} \sum_j f_j(a_i, a_{-i})$.
    Compute $f_{k+i}(a_{-i}) = \max_{a_i} \sum_j f_j(a_i, a_{-i})$.
    Remove all $f_j(a_i, a_{-i})$ from F and add $f_{k+i}(a_{-i})$ in F.
  End
  For each agent $i = n, n-1,...,1$
    Choose $a_i^* \in B_i(a_{-i}^*)$ based on a fixed ordering of actions.
  End
End

حذف متغیر از طریق برقراری ارتباط

با هدف یافتن تعادل نش بهینه‌ی پارتو است که u (سودمندی سراسری) را ماکزیمم می‌کند

VARIABLE ELIMINATION VIA COMMUNICATION

For each agent i in parallel
  If $i \neq 1$
    Wait until agent $i-1$ sends OK.
  End
  Let $f_j(a_i, a_{-i})$ be all local payoff functions (initial and communicated) that involve agent i.
  Compute $B_i(a_{-i}) = \arg\max_{a_i} \sum_j f_j(a_i, a_{-i})$.
  Compute $f^*(a_{-i}) = \max_{a_i} \sum_j f_j(a_i, a_{-i})$.
  Send $f^*(a_{-i})$ to agent $j = \min\{i+1, \ldots, n\}$, $j \in -i$.
  If $i \neq n$
    Send OK to agent $i+1$.
    Wait until all a^*_{-i} are received.
  End
  Choose any $a_i^* \in B_i(a^*_{-i})$.
  Broadcast a_i^* to all agents j such that $a_i \in \text{domain}(B_j)$.
End

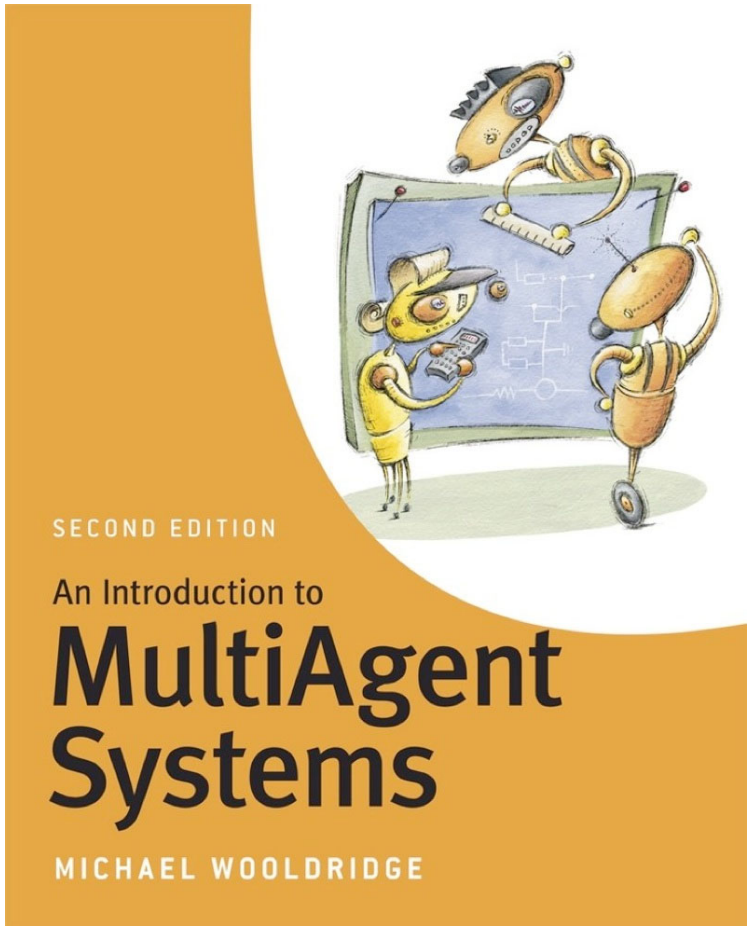

Communicating
Key concepts: Not exchanging bit streams, but requesting and informing
Speech acts
the conceptual foundation for agent communication
Austin (1962) 'How to do things with words'
reaction against logical positivism
Austin realized some utterances change the world
identified performative verbs
request
inform
promise
felicity conditions
Searle (1969)
characterized conditions of success
normal I/O conditions
preparatory conditions
sincerity conditions
typology of speech acts
representatives (e.g. informing)
directives (e.g. requesting)
commissives (e.g. promising)
expressives (e.g. thanking)
declarations (e.g. declaring war)
plan-based theory of speech acts
model speech acts with action formalism
use STRIPS formalism
cando.pr
want.pr
allow planning with speech acts
speech acts as rational action
agent communication languages
KQML
sibling of KIF
message expressed in KIF
about 40 performatives
no real semantics
FIPA ACL
related to KIF
more refined performative set
inform and request as primitives
detailed semantics
several implementations; JADE is best known

سیستم‌های چندعاملی

برقراری ارتباط

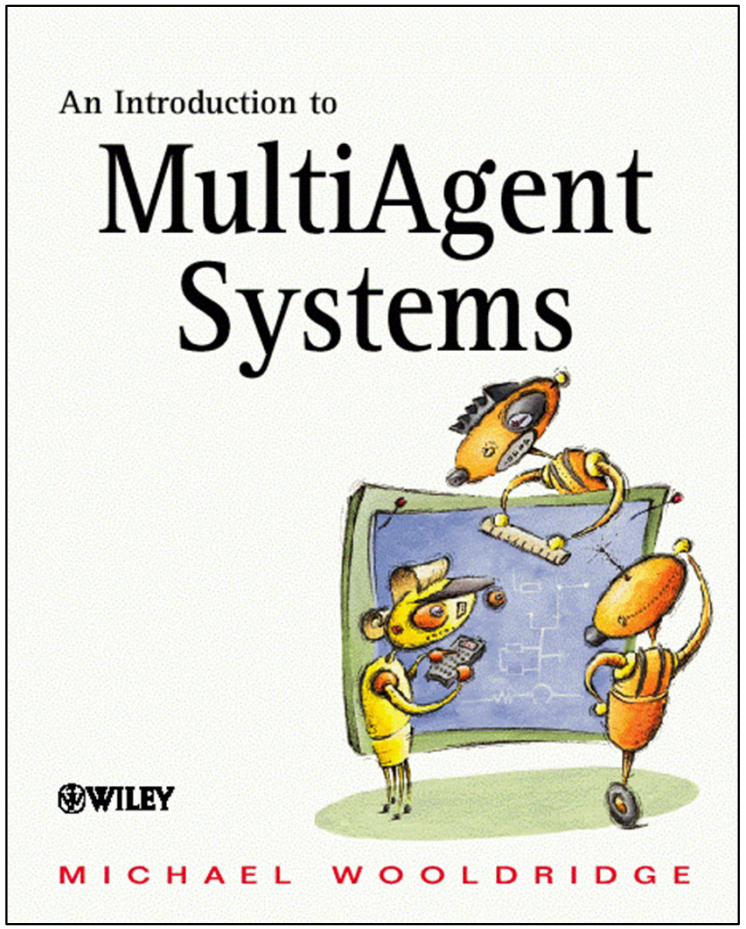
منابع

منبع اصلی

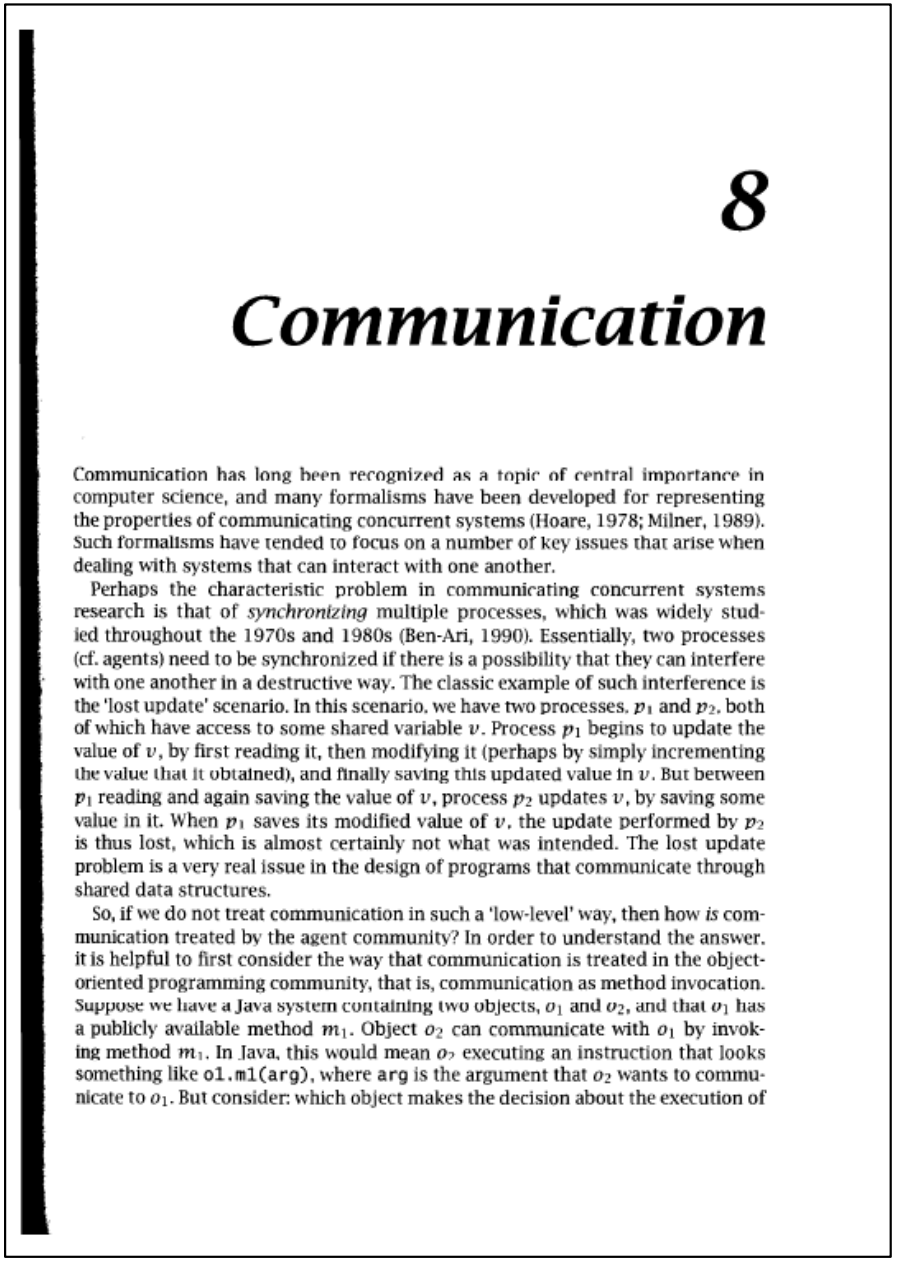


Michael Wooldridge,
An Introduction to Multiagent Systems,
Second Edition,
John Wiley & Sons, 2009.
Chapter 7

منبع کمکی



Michael Wooldridge,
An Introduction to Multiagent Systems,
John Wiley & Sons, 2002.
Chapter 8



8

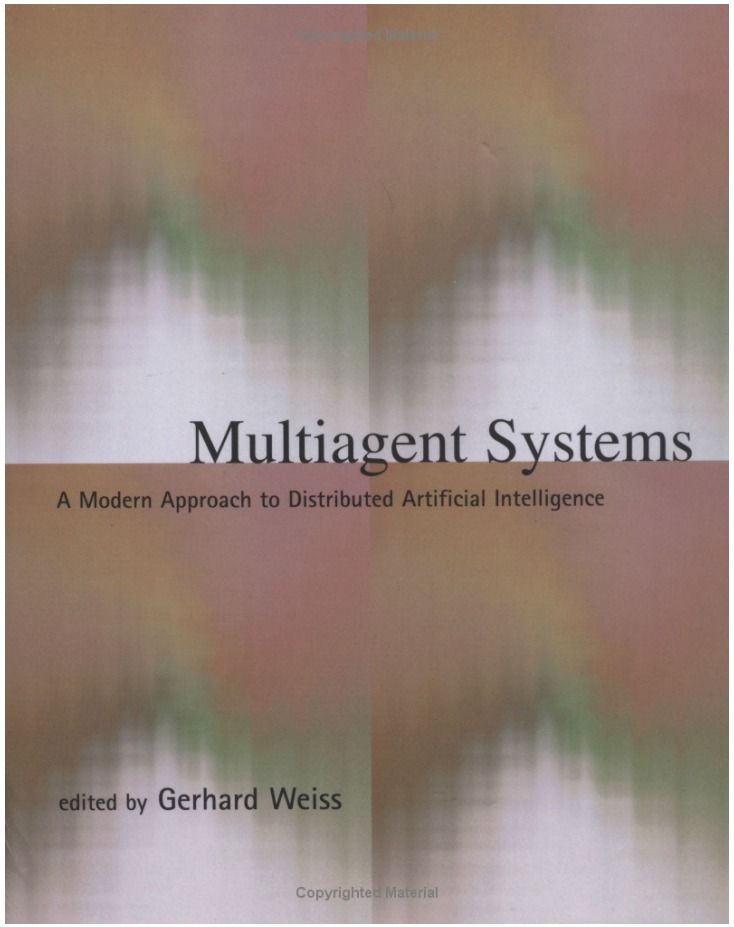
Communication

Communication has long been recognized as a topic of central importance in computer science, and many formalisms have been developed for representing the properties of communicating concurrent systems (Hoare, 1978; Milner, 1989). Such formalisms have tended to focus on a number of key issues that arise when dealing with systems that can interact with one another.

Perhaps the characteristic problem in communicating concurrent systems research is that of *synchronizing* multiple processes, which was widely studied throughout the 1970s and 1980s (Ben-Ari, 1990). Essentially, two processes (cf. agents) need to be synchronized if there is a possibility that they can interfere with one another in a destructive way. The classic example of such interference is the 'lost update' scenario. In this scenario, we have two processes, p_1 and p_2, both of which have access to some shared variable v. Process p_1 begins to update the value of v, by first reading it, then modifying it (perhaps by simply incrementing the value that it obtained), and finally saving this updated value in v. But between p_1 reading and again saving the value of v, process p_2 updates v, by saving some value in it. When p_1 saves its modified value of v, the update performed by p_2 is thus lost, which is almost certainly not what was intended. The lost update problem is a very real issue in the design of programs that communicate through shared data structures.

So, if we do not treat communication in such a 'low-level' way, then how *is* communication treated by the agent community? In order to understand the answer, it is helpful to first consider the way that communication is treated in the object-oriented programming community, that is, communication as method invocation. Suppose we have a Java system containing two objects, o_1 and o_2, and that o_1 has a publicly available method m_1. Object o_2 can communicate with o_1 by invoking method m_1. In Java, this would mean o_2 executing an instruction that looks something like `o1.m1(arg)`, where `arg` is the argument that o_2 wants to communicate to o_1. But consider: which object makes the decision about the execution of

منبع کمکی



Gerhard Weiss (ed.),
Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence,
MIT Press, 1999.
Chapter 2 (2.1-2.2)

2 Multiagent Systems and Societies of Agents

Michael N. Huhns and Larry M. Stephens

2.1 Introduction

Agents operate and exist in some environment, which typically is both computational and physical. The environment might be open or closed, and it might or might not contain other agents. Although there are situations where an agent can operate usefully by itself, the increasing interconnection and networking of computers is making such situations rare, and in the usual state of affairs the agent interacts with other agents. Whereas the previous chapter defined the structure and characteristics of an individual agent, the focus of this chapter is on systems with multiple agents. At times, the number of agents may be too numerous to deal with them individually, and it is then more convenient to deal with them collectively, as a society of agents.

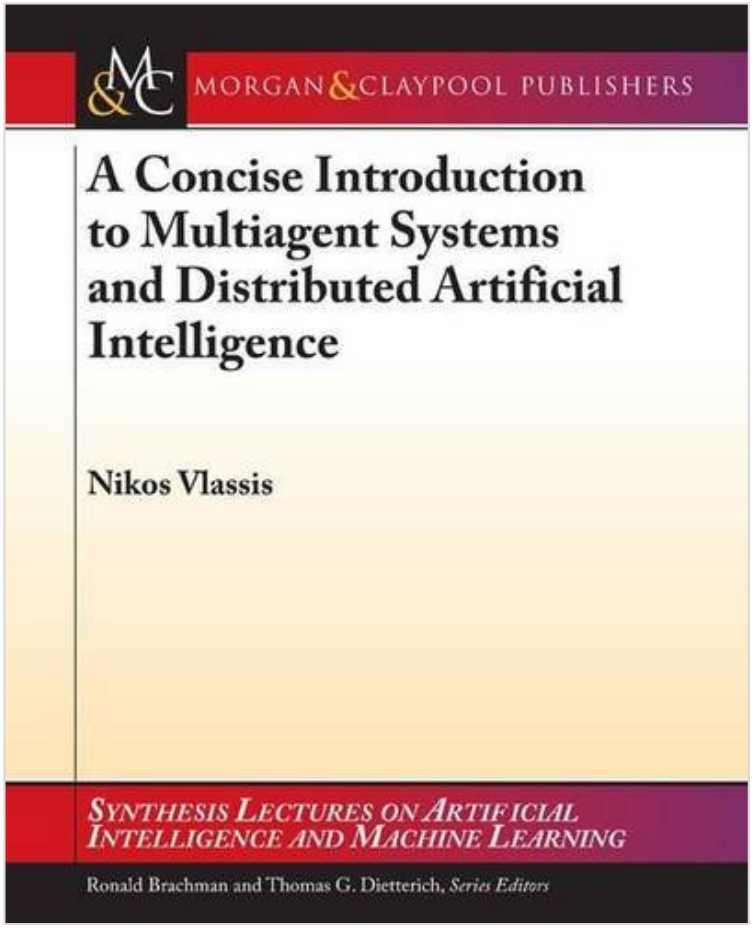
In this chapter, we will learn how to analyze, describe, and design environments in which agents can operate effectively and interact with each other productively. The environments will provide a computational infrastructure for such interactions to take place. The infrastructure will include protocols for agents to communicate and protocols for agents to interact.

Communication protocols enable agents to exchange and understand messages. Interaction protocols enable agents to have conversations, which for our purposes are structured exchanges of messages. As a concrete example of these, a communication protocol might specify that the following types of messages can be exchanged between two agents:

- Propose a course of action
- Accept a course of action
- Reject a course of action
- Retract a course of action
- Disagree with a proposed course of action
- Counterpropose a course of action

Based on these message types, the following conversation—an instance of an

منبع کمکی



Nikos Vlassis,
A Concise Introduction to Multiagent Systems and Distributed Artificial Intelligence,
Morgan & Claypool, 2007.
Chapter 4

CHAPTER 4

Coordination

In this chapter we address the problem of multiagent **coordination**. We analyze the problem using the framework of strategic games that we studied in Chapter 3, and we describe several practical techniques like social conventions, roles, and coordination graphs.

4.1 COORDINATION GAMES

As we argued in Chapter 1, decision making in a multiagent system should preferably be carried out in a decentralized manner for reasons of efficiency and robustness. This additionally requires developing a coordination mechanism. In the case of **collaborative** agents, coordination ensures that the agents do not obstruct each other when taking actions, and that these actions serve the common goal of the team (for example, two teammate soccer robots must coordinate their actions when deciding who should go for the ball). Informally, coordination can be regarded as the process by which the individual decisions of the agents result in good joint decisions for the group.

Formally, we can model a coordination problem as a **coordination game** using the tools of game theory, and solve it according to some solution concept, for instance Nash equilibrium. We have already seen an example in Fig. 3.2(b) of Chapter 3 of a strategic game where two cars meet at a crossroad and one driver should cross and the other one should stop. That game has two Nash equilibria, (*Cross*, *Stop*) and (*Stop*, *Cross*). In the case of n collaborative agents, all agents in the team share the same payoff function $u_1(a) = \ldots = u_n(a) \equiv u(a)$ in the corresponding coordination game. Figure 4.1 shows an example of a coordination game (played between two agents who want to go to the movies together) that also has two Nash equilibria. Generalizing from these two examples, we can formally define coordination as *the process in which a group of agents choose a single Pareto optimal Nash equilibrium in a game.*

	Thriller	*Comedy*
Thriller	1, 1	0, 0
Comedy	0, 0	1, 1

FIGURE 4.1: A coordination game