

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



سیستم‌های چند عاملی

درس ۶

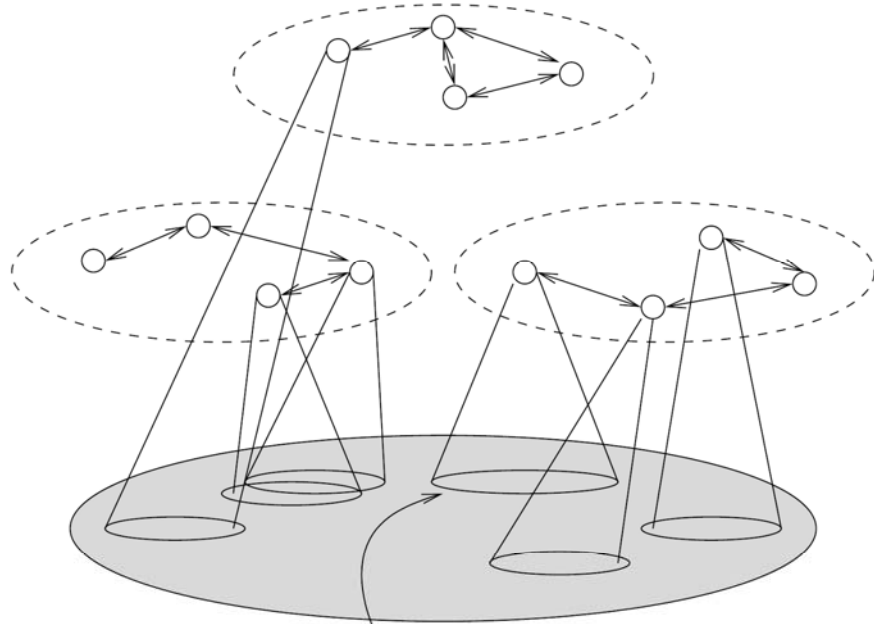
جامعه‌ی عامل‌ها و سیستم‌های چند عاملی

Society of Agents and Multiagent Systems

کاظم فولادی قلعه
دانشکده مهندسی، پردیس فارابی
دانشگاه تهران

<http://courses.fouladi.ir/mas>

سیستم چندعاملی



KEY

رابطه‌ی سازمانی

اندرکنش (تعامل)

عامل



organisational relationship

interaction

agent

Environment محیط

sphere of influence حوزهی تأثیر

برقراری رابطه میان عامل‌ها

انگیزه

MOTIVATION

دلایل برقراری رابطه میان عامل‌های محاسباتی

<p>پیمانه‌ای‌سازی <i>Modularization</i></p>	<p>موازی‌سازی <i>Parallelization</i></p>	<p>به اشتراک گذاری <i>Sharing</i></p>	<p>همکاری <i>Cooperation</i></p>
<p>توسعه و پیاده‌سازی سیستم به صورت پیمانه‌ای (<i>Modular</i>)</p>	<p>انجام کار موازی بر روی مسائل مشترک</p>	<p>به اشتراک گذاری تجربه‌ها</p>	<p>همکاری در حل مسئله</p>
<p>استفاده‌ی مجدد <i>Reusing</i></p>	<p>چند خبرگی <i>Multi-Expertise</i></p>	<p>تحمل‌پذیری نقص <i>Fault-Tolerance</i></p>	
<p>استفاده‌پذیری مجدد</p>	<p>بازنمایی دیدگاه‌های چندگانه و دانایی چند خبره</p>	<p>تحمل‌پذیری نقص از طریق افزونگی</p>	

روابط وابستگی در سیستم‌های چندعاملی

DEPENDENCE RELATIONS IN MULTIAGENT SYSTEMS

بین دو عامل رابطه‌ی وابستگی وجود دارد،
اگر یکی از عامل‌ها برای رسیدن به هدفش به عامل دیگر نیاز داشته باشد.

انواع رابطه‌های وابستگی

وابستگی متقابل <i>Reciprocal Dependence</i>	دوجانبه <i>Mutual</i>	یک‌جانبه <i>Unilateral</i>	استقلال <i>Independence</i>
عامل اول برای هدفی مشخص به عامل دوم وابسته است، در حالی که عامل دوم برای هدف دیگری به عامل اول وابسته است.	هر دو عامل برای رسیدن به یک هدف مشترک به هم وابسته هستند.	یک عامل به عامل دیگر وابسته است، اما عامل دوم به عامل اول وابسته نیست.	هیچ وابستگی بین دو عامل وجود ندارد.

عامل اول به وجود وابستگی باور دارد اما مطمئن نیست که عامل دیگر نیز به این وابستگی باور داشته باشد.	باور محلی به رابطه <i>Locally Believed</i>	انواع رابطه بر اساس باور عامل‌ها
عامل اول به وجود وابستگی باور دارد و مطمئن است که عامل دیگر نیز به این وابستگی باور داشته باشد.	باور دوجانبه به رابطه <i>Mutually Believed</i>	

سیستم‌های چند عاملی و جامعه‌ی عامل‌ها

سیستم‌های چندعاملی بهترین راه برای مشخص‌سازی یا طراحی سیستم‌های محاسباتی توزیع‌شده هستند.

- در محیط اطراف ما، پردازنده‌های کامپیوتری تعبیه شده‌ی زیادی وجود دارد:
- وقتی یک پردازنده‌ها، مثلاً در آشپزخانه به اندازه‌ی کافی هوشمند باشند که بتوانند «عامل» در نظر گرفته شوند، معمول است که آنها را مانند یک «کس» ببینیم تا «چیز»، مثلاً: «اجاق گاز، می‌داند چه زمانی گوشت پخته می‌شود».
 - وقتی این سیستم‌ها طوری به هم متصل شوند که بتوانند با هم تعامل کنند، آنها نیز بایستی بدانند، مثلاً: اینکه «گوشت و برنج باید حدوداً در یک زمان آماده شوند».
 - آشپزخانه‌ی شما با این توصیف، چیزی بیش از یک سیستم محاسباتی توزیع‌شده است. در واقع این آشپزخانه یک سیستم چندعاملی است.
 - و این انگیزه ایجاد می‌شود که این سیستم‌ها با عبارت‌های اجتماعی تشریح شوند.

محیط‌های چندعاملی

مشخصه‌ها

MULTIAGENT ENVIRONMENT: CHARACTERISTICS

مشخصه‌های محیط چندعاملی

دربردارنده‌ی عامل‌ها

Agents Included

شامل عامل‌های
خودمختار و توزیع‌شده،
خودمحور یا همکار

طراحی باز

Open Design

معمولاً باز هستند
و طراح متمرکز ندارند.

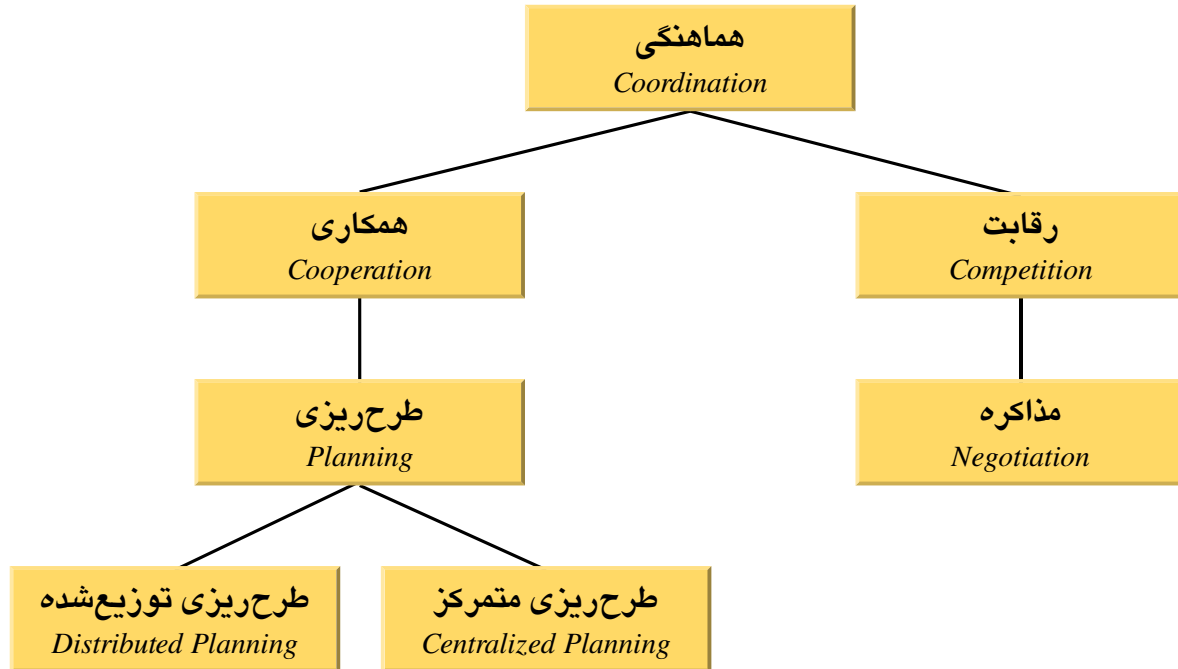
تدارک زیرساخت

Providing Substructure

فراهم کردن زیرساخت:
تعیین‌کننده‌ی پروتکل‌های ارتباطی
و تعاملی

هماهنگی

برخی از روش‌هایی که عامل‌ها رفتارشان را مطابق آن هماهنگ می‌کنند

COORDINATION

انسجام

COHERENCE

یک سیستم به چه میزان می‌تواند به عنوان **یک واحد** تلقی شود؟

انسجام
Coherence

انسجام

ملزومات حفظ انسجام در شرایط نبود کنترل سراسری صریح

COHERENCE

انسجام
Coherence

یک سیستم به چه میزان می‌تواند به عنوان **یک واحد** تلقی شود؟

ملزومات حفظ انسجام در شرایط نبود کنترل سراسری صریح

دانایی و شواهد

Knowledge & Evidences

عامل‌ها باید
دانایی و شواهد
را نگهداری کنند.

اجتناب از تضاد

Conflict Avoidance

عامل‌ها باید
از تضادها و برخوردهای
غیرضروری اجتناب کنند.

وظایف مشترک

Common Tasks

عامل‌ها باید
وظایف متداول و مشترک
را مشخص کنند.

اهداف مشترک

Shared Goals

عامل‌ها باید
اهداف مشترک با سایر
عامل‌ها را مشخص کنند.

انسجام

برخی از شکل‌های مفید سازمان‌دهی عامل‌ها

COHERENCE

انسجام
Coherence

یک سیستم به چه میزان می‌تواند به عنوان **یک واحد** تلقی شود؟

برخی شکل‌های سازمان‌دهی میان عامل‌ها، به انسجام کمک می‌کند:

اصول اقتصادی بازارها
Economic Principles of Markets

تعهدات اجتماعی
Social Commitments

سایمون معتقد است که:

اگرچه بازار برای شفاف‌سازی همه‌ی کالاها (مثلاً یافتن قیمت فروش هر چیزی) عالی است، اما در محاسبه‌ی تخصیص بهینه‌ی منابع کمتر مؤثر است: برای این، ساختارهای سازمانی ضروری است.

پروتکل‌های هماهنگی

COORDINATION PROTOCOLS

پروتکل‌های هماهنگی

Coordination Protocols

قراردادهای حاکم بر فرآیند هماهنگ‌سازی

وقتی منابع محدود است، عامل‌ها باید فعالیت‌هایشان را با هم هماهنگ کنند تا
(۱) به **منافع خود** برسند و (۲) **اهداف گروه** را برآورده سازند.

چرایی ضرورت هماهنگ‌سازی کنش‌های عامل‌ها

محدودیت عامل‌ها

Agents Limitation

هیچ عاملی، شایستگی، منابع یا اطلاعات کافی برای دستیابی به اهداف کل سیستم را ندارد.

قیدهای سراسری

Global Constraints

قیدهای سراسری باید برآورده شود

وابستگی کنش‌ها

Actions Dependencies

بین کنش‌های عامل‌ها وابستگی‌هایی وجود دارد

هماهنگی

اهداف هماهنگ‌سازی

COORDINATION

اهداف هماهنگ‌سازی

منع دوباره‌کاری

Redundancy Prevention

جلوگیری از حل مسئله به صورت
افزونه (دوباره‌کاری)

کنش‌های همگام

Synchronous Actions

اطمینان از اینکه
کنش‌های عامل‌ها همگام شده است

اطلاعات به موقع

Timely Information

فراهم کردن
اطلاعات به موقع برای سایر عامل‌ها

هماهنگی

سنگ‌بنای هماهنگی: تعهدات و قراردادهای

CORNERSTONES OF COORDINATION: COMMITMENTS AND CONVENTIONS

سنگ‌بنای هماهنگی

قرارداد <i>Convention</i>	تعهد <i>Commitment</i>
وسيله‌ای برای مدیریت تعهدات در موقعیت‌های متغیر	تضمینی برای وقوع یک دسته‌ی مشخص از کنش‌ها
<p>قراردادها:</p> <p>* شرایطی که تحت آنها تعهدات باید مجدداً ارزیابی شوند، را محدود می‌کند.</p> <p>* کنشی که باید واقع شود را مشخص می‌کنند:</p> <p>ابقا، تصفیه، واگذاری</p> <p>وقتی یک هدف وابسته تغییر می‌کند، عامل‌های مربوط باید مطلع شوند.</p>	<p>تعهدات، به عامل‌ها اجازه می‌دهند که فعالیت‌های آینده‌ی سایر عامل‌ها را پیش‌بینی کنند (بخصوص وقتی که به تضادهای وابستگی‌های میان‌عاملی، قیدهای سراسری و بهره‌برداری از منابع برخورد می‌کنیم).</p> <p>وقتی موقعیت تغییر می‌کند، عامل‌ها باید اعتبار تعهدات را مجدداً ارزیابی کنند.</p> <p>تعهدات میان عامل‌ها هم باید از درون سازگار باشند و هم باید با باورهای عامل‌ها سازگار باشد.</p>

هماهنگی

مثالی از یک قرارداد اجتماعی (۱): قرارداد اجتماعی پهنای باند محدود

LIMITED-BANDWIDTH SOCIAL CONVENTION

INVOKE WHEN

- Local commitment dropped
- Local commitment satisfied

ACTIONS

- Rule1: IF Local commitment satisfied
THEN inform all related commitments
- Rule2: IF local commitment dropped because unattainable or
motivation not present
THEN inform all strongly related commitments
- Rule3: IF local commitment dropped because unattainable or
motivation not present
AND communication resources not overburdened
THEN inform all weakly related commitments

مناسب برای محیطی با
منابع ارتباطی محدود

هماهنگی

مثالی از یک قرارداد اجتماعی (۲): قرارداد اجتماعی اقدام مشترک پایه

BASIC JOINT-ACTION SOCIAL CONVENTION

INVOKE WHEN

- Status of commitment to joint action changes
- Status of commitment to attaining joint action in present team context changes
- Status of joint commitment of a team member changes

ACTIONS

- Rule1: IF Status of commitment to joint action changes
OR Status of commitment to present team context changes
THEN inform all other team members of these changes
- Rule2: IF Status of joint commitment of a team member changes
THEN Determine whether joint commitment still viable

وقتی عامل‌ها تصمیم می‌گیرند یک اقدام مشترک را دنبال کنند، خودشان را به یک هدف مشترک توأمآ متعهد می‌کنند.

حداقل موارد اشتراکی برای یک تیم

وضعیت تعهد

به چارچوب تیمی مشترک

وضعیت تعهد

به هدف مشترک

وقتی باور یکی از عامل‌ها نسبت به هر یک از اینها تغییر کند، همه‌ی اعضای تیم باید مطلع شوند.

پروتکل‌های همکاری

COOPERATION PROTOCOLS

قراردادهای حاکم بر فرآیند همکاری

پروتکل‌های همکاری
Cooperation Protocols

فرآیند یک پروتکل همکاری

توزیع
وظایف



تجزیه‌ی
وظایف

با هدف امکان‌بخشی کاهش پیچیدگی آن وظیفه

موارد توجه در یک پروتکل همکاری

تعامل زیروظیفه‌ها/تداخل عامل‌ها

ممکن است بین زیروظیفه‌ها تعامل
و بین عامل‌ها تداخل وجود داشته باشد.

منابع و ظرفیت عامل‌ها

تجزیه باید منابع و ظرفیت عامل‌ها
را در نظر بگیرد.

تجزیه‌های ممکن وظیفه‌ها

چنین سیستمی اغلب باید بین تجزیه‌های
ممکن وظیفه‌ها، تصمیم‌گیری کند.

پروتکل‌های همکاری

منبع تجزیه‌ی وظیفه

COOPERATION PROTOCOLS

منبع تجزیه‌ی وظیفه‌ها

مسئله	عامل	طراح
در بازنمایی مسئله ذاتاً وجود داشته باشد (مانند گراف <i>AND-OR</i>)	توسط عامل‌ها با استفاده از طرح‌ریزی سلسله‌مراتبی انجام شود.	توسط طراح انجام شود و در خلال پیاده‌سازی برنامه‌نویسی شود.

پروتکل‌های همکاری

انواع تجزیه‌ی وظایف

COOPERATION PROTOCOLS

انواع تجزیه‌ی وظایف

کارکردی

Functionally

با توجه به خبرگی عامل‌های موجود

فضائی

Spatially

بر اساس جانمایی منابع اطلاعاتی یا نقاط تصمیم‌گیری

پروتکل‌های همکاری

ضوابط توزیع وظایف

COOPERATION PROTOCOLS

وقتی وظایف تجزیه شدند، بر اساس ضوابط زیر، توزیع می‌شوند:

اجتناب از ایجاد سربار روی منابع بحرانی

انتساب وظایف به عامل‌ها با مطابقت دادن توانایی‌ها

و اداری کردن یک عامل دارای دید گسترده به «انتساب وظایف به سایر عامل‌ها»

انتساب مسئولیت‌های همپوشان به عامل‌ها
برای دستیابی به انسجام بیشتر

انتساب وظایف دارای وابستگی بالا به عامل‌های مجاور فضائی و معنایی
به منظور می‌نیم کردن هزینه‌های ارتباطات و همگام‌سازی

انتساب مجدد وظایف در صورت لزوم
برای کامل کردن وظایف اورژانسی

پروتکل‌های همکاری

مکانیزم‌های توزیع وظایف

COOPERATION PROTOCOLS

مکانیزم‌های توزیع وظایف

ساختار سازمانی <i>Organizational Structure</i>	طرح‌ریزی چندعاملی <i>Multiagent Planning</i>	شبکه‌ی پیمانی <i>Contract Net</i>	مکانیزم‌های بازار <i>Market Mechanisms</i>
عامل‌ها دارای مسئولیت‌های ثابت برای وظایف خاص هستند	عامل‌های طرح‌ریزی وظایف را نسبت می‌دهند.	آگهی، قرعه‌کشی، پاداش	وظایف به عامل‌ها تخصیص داده می‌شود، بر اساس: توافق عمومی یا انتخاب دوجانبه (مانند قیمت‌گذاری روی کالا)

شبکه‌ی پیمانی

CONTRACT NET

پروتکل شبکه‌ی پیمانی

Contract Net Protocol

یکی از متداول‌ترین مکانیزم‌ها برای توزیع وظیفه‌ها

این پروتکل تعامل برای «حل مسئله‌ی همکاری»،
بر اساس مکانیزم پیمانکاری مورد استفاده در فعالیت‌های کسب و کار مدل شده است.

این پروتکل، راه‌حلی برای **مسئله‌ی اتصال‌دهی** فراهم می‌کند.

یافتن یک عامل مناسب برای کار بر روی یک وظیفه‌ی داده شده

مسئله‌ی اتصال‌دهی
Connection Problem

شبکه‌ی پیمانی

CONTRACT NET

شبکه‌ی پیمانی *Contract Net*

پیمانکار
Contractor

عامل‌هایی که ممکن
است بتوانند آن وظیفه
را حل کنند، پیمانکاران
بالقوه هستند.

مدیر
Manager

یک عامل با وظیفه‌ای که
باید آن را حل کند،
مدیر نام دارد.

شبکه‌ی پیمانی

پروتکل: از نگاه مدیر و پیمانکار

CONTRACT NET

شبکه‌ی پیمانی

Contract Net

پیمانکار

Contractor

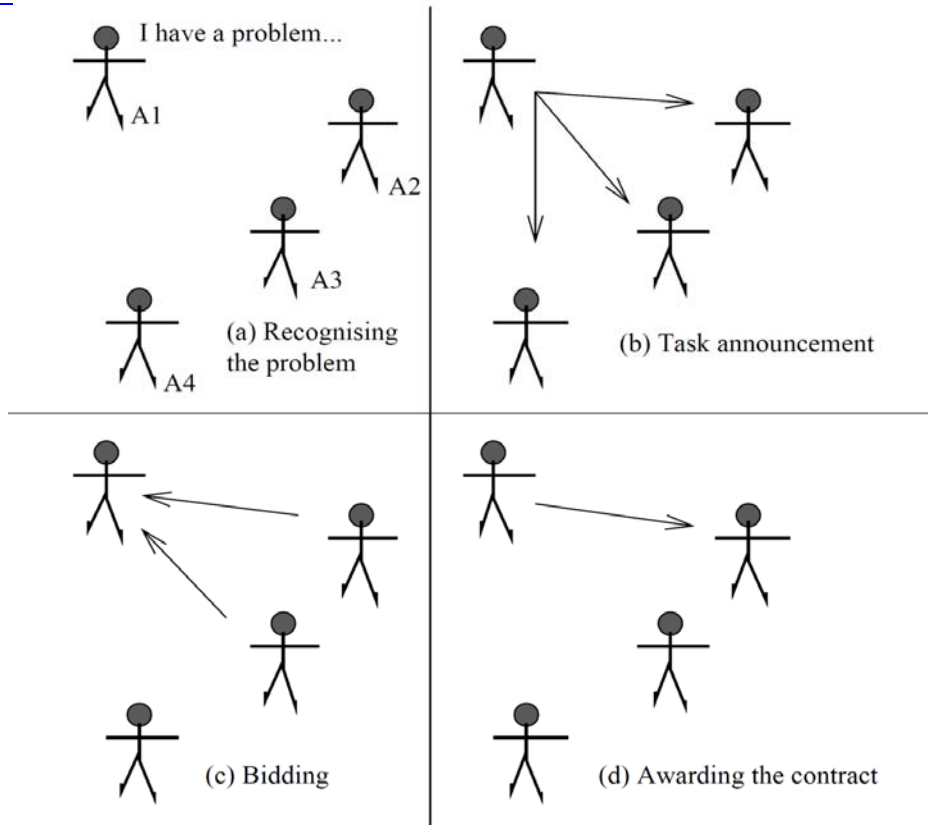
- دریافت آگهی وظیفه‌ها
- ارزیابی قابلیت خود
برای پاسخ‌دهی به آن آگهی‌ها
- پاسخ دادن
(انصراف دادن یا پیشنهاد دادن)
- انجام وظیفه
در صورتی که پیشنهاد وی پذیرفته شد
- گزارش دادن نتایج خود

مدیر

Manager

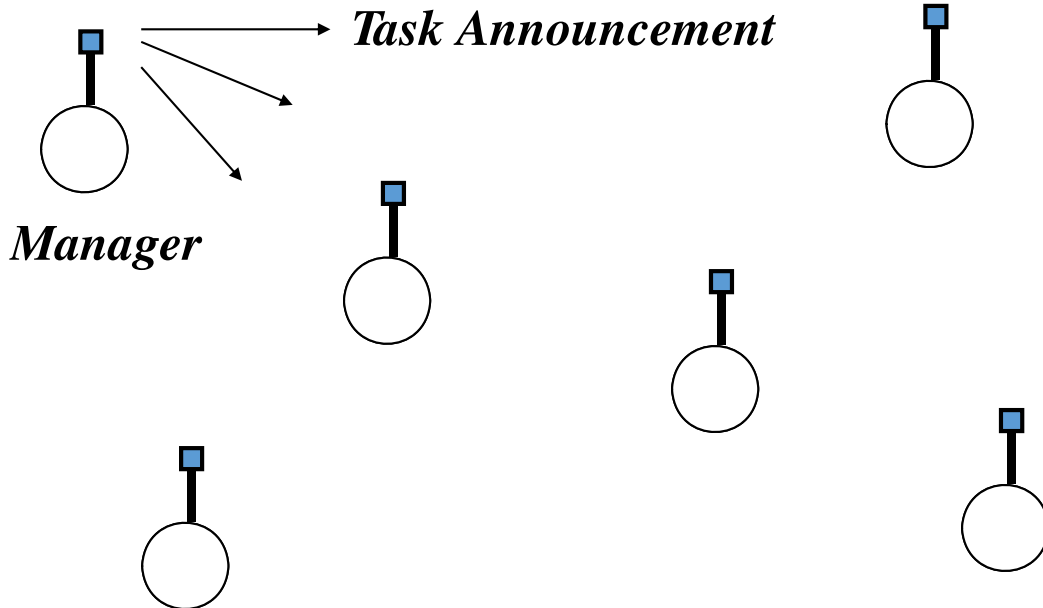
- آگهی کردن وظیفه‌ای که
نیاز به انجام آن وجود دارد.
- دریافت و ارزیابی پیشنهادهای
از پیمانکاران بالقوه
- اعطای یک پیمان (قرارداد)
به یک پیمانکار مناسب
- دریافت و ترکیب نتایج

شبکه‌ی پیمانی

CONTRACT NET

شبکه‌ی پیمانی

(۱) صدور آگهی یک وظیفه توسط یک عامل (مدیر)

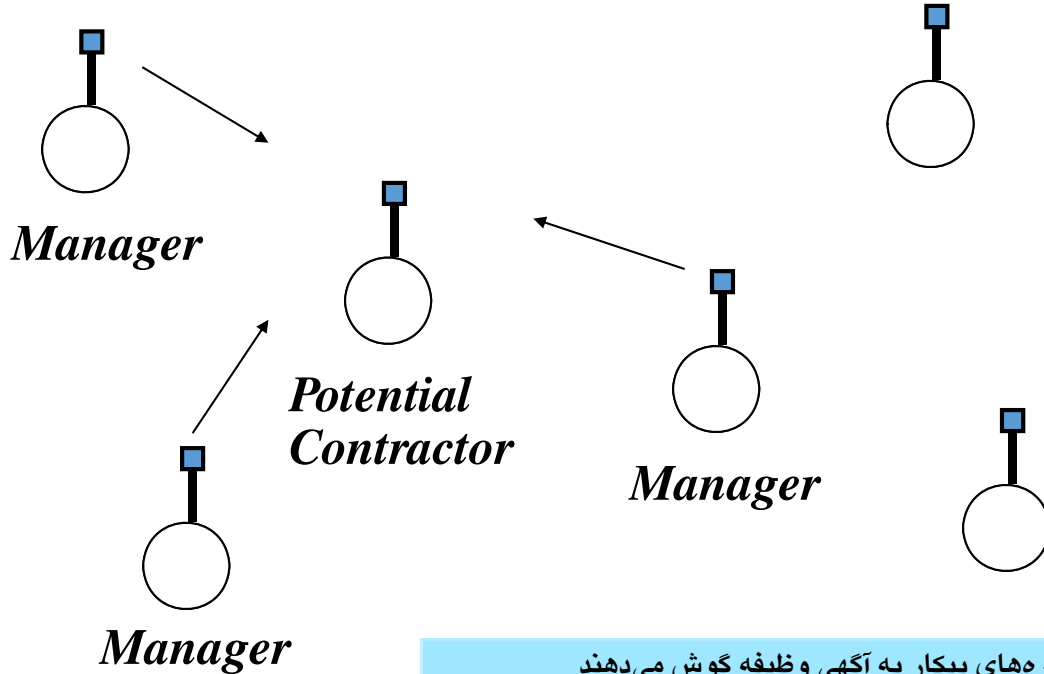


یک گره آگهی وظیفه را صادر می‌کند

Node Issues Task Announcement

شبکه‌ی پیمانی

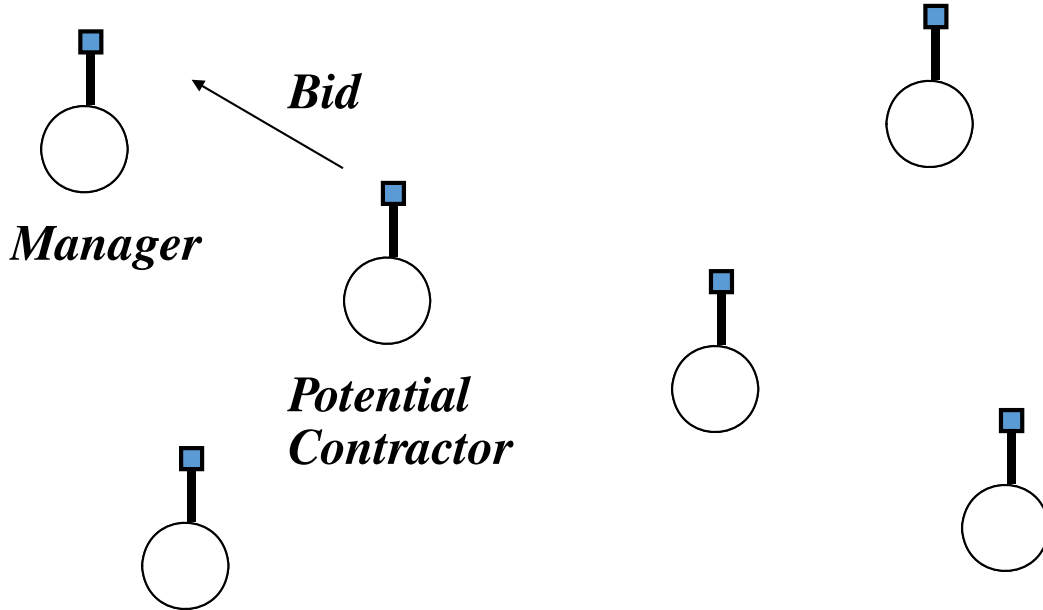
(۲) عامل‌های بیکار به آگهی وظیفه گوش می‌دهند



گره‌های بیکار به آگهی وظیفه گوش می‌دهند
Idle Node Listening to Task Announcements

شبکه‌ی پیمانی

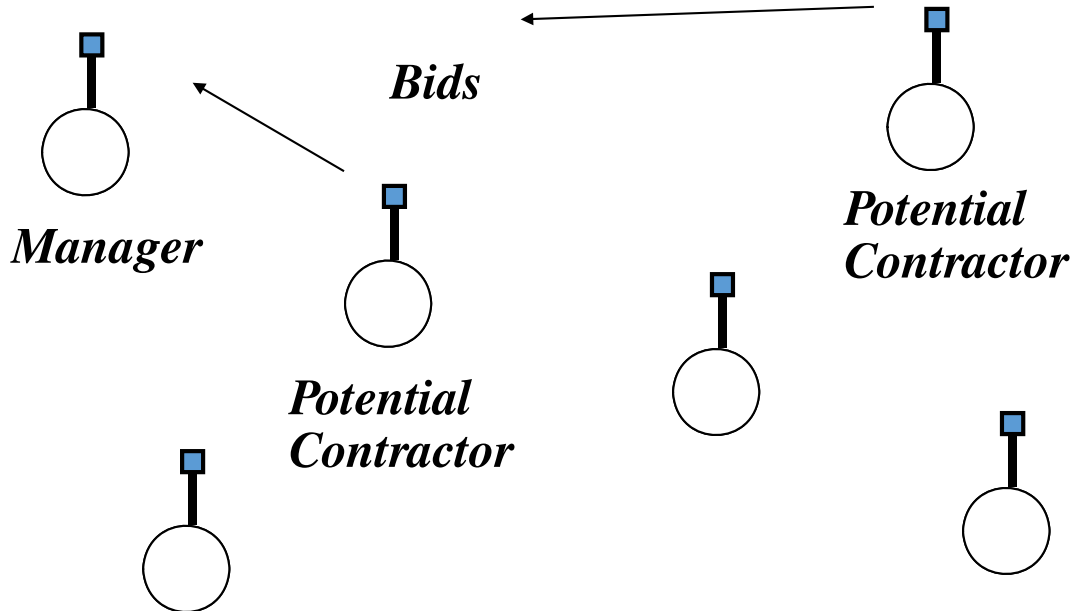
۳) یک عامل پیشنهاد خود را ارائه می‌کند



یک گره پیشنهاد خود را ارائه می‌کند
Node Submitting a Bid

شبکه‌ی پیمانی

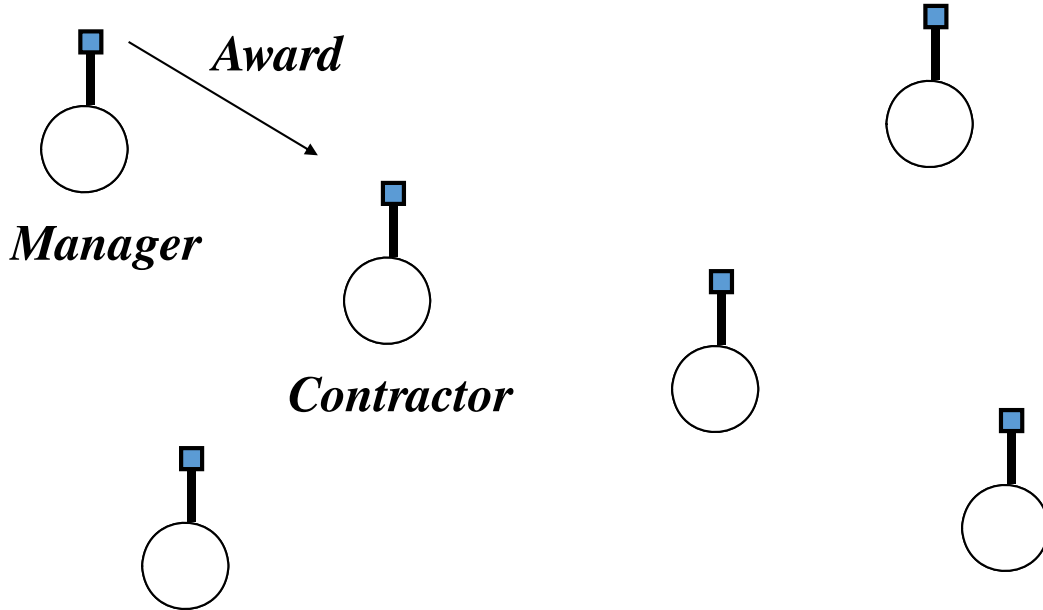
(۴) مدیر به پیشنهادهای گوش می‌دهد



مدیر به پیشنهادهای گوش می‌دهد
Manager listening to bids

شبکه‌ی پیمانی

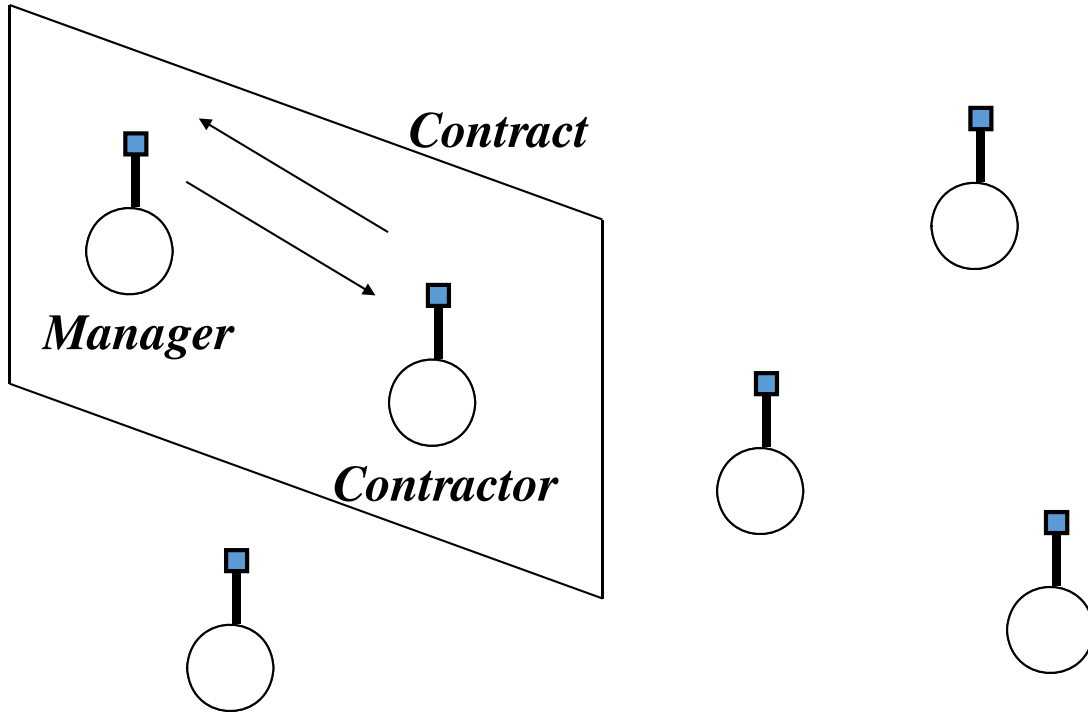
(۵) مدیر اعطای وظیفه را انجام می‌دهد



مدیر اعطای وظیفه را انجام می‌دهد
Manager Making an Award

شبکه‌ی پیمانی

۶) پیمان برقرار می‌شود



پیمان بسته می‌شود
Contract Established

شبکه‌ی پیمانی

ویژگی‌ها

<p>هر عامل می‌تواند به عنوان یک مدیر عمل کند؛ همچنین می‌تواند به عنوان پیمانکار عمل نماید.</p>	<p>مدیر <i>Manager</i></p> <p>پیمانکار <i>Contractor</i></p>
<p>سلسله مراتب کنترل، تجزیه‌ی بیشتر وظایف را ممکن می‌سازد.</p>	<p>سلسله مراتب کنترل <i>Control Hierarchy</i></p>
<p>اگر یک پیمانکار نتواند راه حل راضی‌کننده‌ای را فراهم کند، عامل دیگری در نظر گرفته می‌شود.</p>	<p>تنزل کارآیی مطبوع <i>Graceful Performance Degradation</i></p>

شبکه‌ی پیمانی

فرایند آگهی کردن وظیفه

فیلدهای آگهی وظیفه	
یک یا چند پیمانکار بالقوه	گیرنده <i>Adressee</i>
باید توسط پیشنهاد دهندگان رعایت شود	مشخص‌سازی شایستگی <i>Eligibility Specification</i>
به پیمانکاران بالقوه اجازه می‌دهد آگهی‌ها را رتبه‌بندی کنند	تجرید وظیفه <i>Task Abstraction</i>
اطلاعاتی که باید در یک پیشنهاد قرار داده شود تا مدیر بتواند پیشنهادها را مقایسه کند	مشخص‌سازی پیشنهاد <i>Bid Specification</i>
مهلت زمانی برای دریافت پیشنهادها	زمان انقضا <i>Expiration Time</i>

شبکه‌ی پیمانی



شبکه‌ی پیمانی



شبکه‌ی پیمانی

محدودیت‌ها

محدودیت‌های شبکه‌ی پیمانی

یک وظیفه می‌تواند به یک پیمانکار با قابلیت محدود داده شود، اگر یک پیمانکار شایسته‌تر در زمان اعطا، مشغول باشد.

یک مدیر مجبور نیست که به پیمانکاران بالقوه اطلاع بدهد که اعطای وظیفه به یکی از پیمانکاران صورت گرفته است.

امکان عدم دریافت پیشنهاد توسط مدیر

شبکه‌ی پیمانی

مشکل عدم دریافت پیشنهاد توسط مدیر

دلایل امکان عدم دریافت پیشنهاد توسط مدیر

همه‌ی پیمانکاران بالقوه ممکن است بر روی وظایف دیگری مشغول باشند.

یک پیمانکار بالقوه، وظیفه‌ی پیشنهاد شده را زیر سایر موارد مورد بررسی رتبه‌بندی کند.

هیچ پیمانکاری برای آن وظیفه شایسته نباشد.

راه حل:

مدیر می‌تواند از پیمانکاران پاسخ‌های فوری بخواهد،
با پیام‌هایی چون: «شایسته ولی مشغول»، «ناشایسته»، «غیرعلاقه‌مند»، ...

برای حل چنین مواردی، برای ساده کردن این پروتکل،
قراردادهای مستقیم، بدون مذاکره صادر می‌شوند؛
پیمانکار انتخاب شده، با یک پیغام «پذیرش» یا «رد» پاسخ می‌دهد.

سیستم‌های تخته‌سیاه

BLACKBOARD SYSTEMS

سیستم تخته‌سیاه

Blackboard System (BB)

نخستین طرح برای «حل مسئله‌ی همکارانه»: سیستم تخته سیاه
استعاره از چندین متخصص (منبع دانایی) که دور یک تخته سیاه ایستاده‌اند.

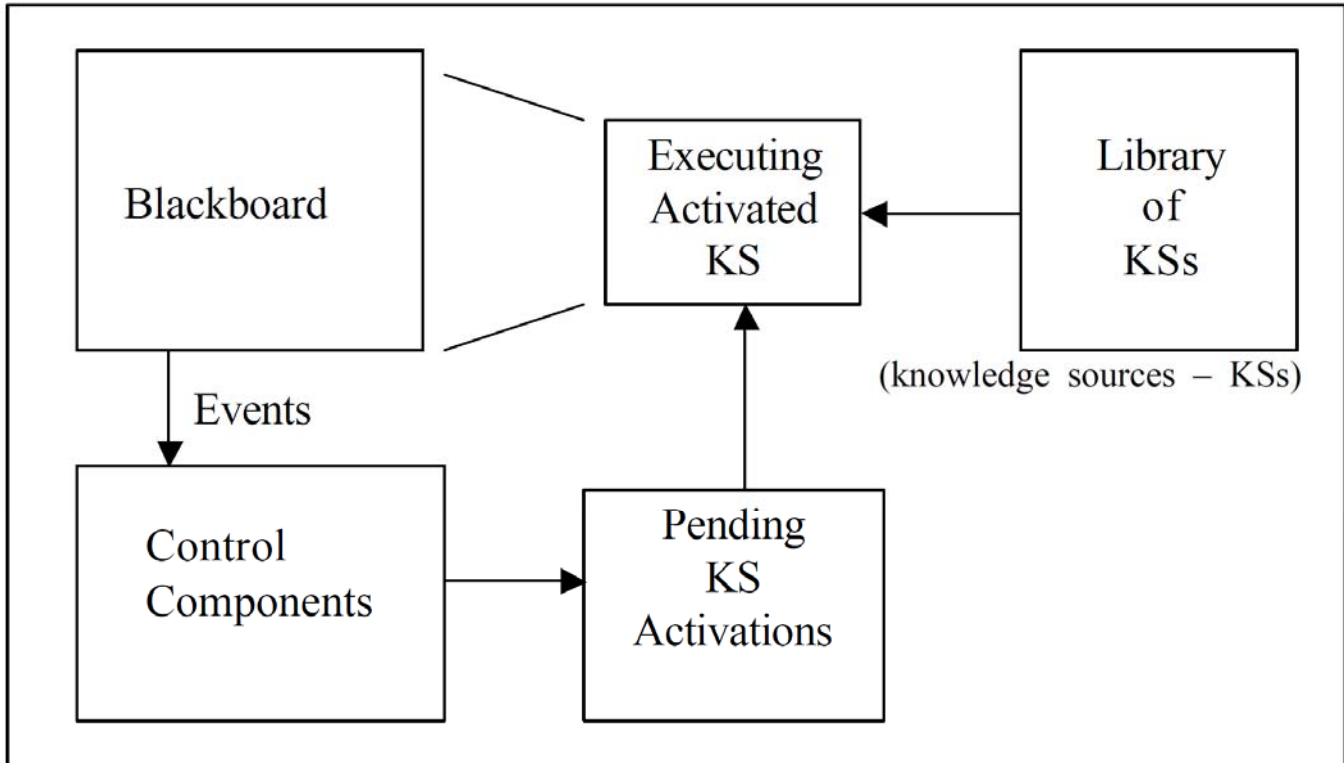
- * هر عامل هرگاه که بتواند، دستاورد خود را روی تخته‌سیاه می‌نویسد.
- * نتایج از طریق یک ساختمان داده‌ی مشترک (تخته سیاه، *BB*)، به اشتراک گذاشته می‌شوند.
- * چند عامل می‌توانند *BB* را بخوانند و در آن بنویسند.
- * عامل‌ها راه‌حل‌های جزئی خود را بر روی *BB* می‌نویسند.
- * *BB* می‌تواند در یک سلسله مراتب سازمان‌دهی شود.

گلوگاه سیستم تخته سیاه:

تخته‌سیاه به انحصار متقابل نیاز دارد.
← هیچ فعالیت همروندی نمی‌تواند وجود داشته باشد.

سیستم‌های تخته‌سیاه

معماری یک سیستم تخته‌سیاه ساده



سیستم‌های تخته‌سیاه

خصوصیات سیستم‌های تخته‌سیاه

عامل‌ها برای حل مسئله به‌طور مستقل کار می‌کنند. هر عامل با تکیه بر خبرگی خود و بدون توجه به سایرین نظر می‌دهد.	استقلال در خبرگی
بازنمایی داخلی و استدلال هر عامل خبره از دیگر خبره‌ها پوشیده است.	تنوع تکنیک‌های حل مسئله
هر نوع اطلاعاتی می‌تواند بر روی تخته سیاه نگاشته شود.	بازنمایی انعطاف‌پذیر
منابع دانایی (عامل‌های خبره) باید قادر باشند اطلاعات نوشته شده توسط دیگر منابع دانایی را به‌درستی تفسیر کنند.	زبان اندرکنش مشترک
منابع دانایی در پاسخ به رویدادهای خارجی و تخته‌سیاه فعال می‌شوند. نمونه‌ی رویدادها: اضافه کردن، تغییر یا حذف اطلاعات تخته سیاه	فعالیت مبتنی بر رویداد
یک بخش کنترل جدا از منابع دانایی عهده‌دار مدیریت حل مسئله می‌شود: انتخاب مناسب‌ترین اجرا برای فعالیت منبع دانایی بعدی	نیاز به کنترل
منابع دانایی هر جا که مناسب باشد برای حل مسئله کمک می‌کنند: مانند تصحیح راه‌حل، شروع خط جدیدی از استدلال	حل افزایشی

سازمان‌دهی سیستم‌های چندعاملی

انواع سازمان‌دهی در سیستم‌های چندعاملی

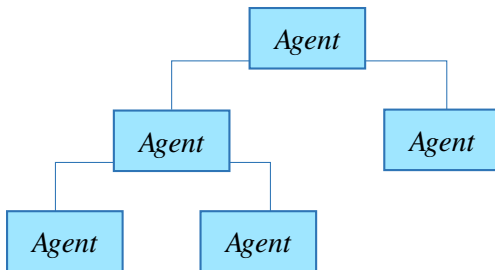
<p>پیمانه‌ای <i>Modular</i></p>	<p>جزئی از کل <i>Part of the Whole</i></p>	<p>مسطح <i>Flat</i></p>	<p>سلسله‌مراتبی <i>Hierarchical</i></p>
<p>هر پیمانه خود می‌تواند یک سیستم چندعاملی باشد.</p>	<p>هر عامل جزئی از یک عامل بزرگ‌تر است.</p>	<p>همه‌ی عامل‌ها در یک سطح؛ امکان ارتباط مستقیم هر عامل با سایر عامل‌ها</p>	<p>ارتباط عامل‌ها در یک ساختار سلسله‌مراتبی</p>

سازمان‌دهی سیستم‌های چندعاملی

ساختار سلسله‌مراتبی

انواع سازمان‌دهی در سیستم‌های چندعاملی

پیمانه‌ای <i>Modular</i>	جزئی از کل <i>Part of the Whole</i>	مسطح <i>Flat</i>	سلسله‌مراتبی <i>Hierarchical</i>
هر پیمانه خود می‌تواند یک سیستم چندعاملی باشد.	هر عامل جزئی از یک عامل بزرگ‌تر است.	همه‌ی عامل‌ها در یک سطح؛ امکان ارتباط مستقیم هر عامل با سایر عامل‌ها	ارتباط عامل‌ها در یک ساختار سلسله‌مراتبی



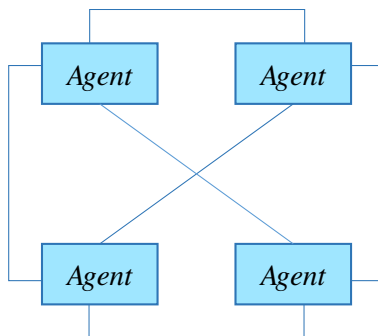
- مزیت: عدم نیاز به سرویس مکان‌یابی
- مزیت: کاهش قابل ملاحظه‌ی حجم ارتباطات
- عیب: عدم امکان سازمان‌دهی پویا برای بهبود عملکرد
- عیب: وابستگی لایه‌های پایین به لایه‌های بالاتر
- عیب: وجود نوعی کنترل مرکزی
- عیب: کاهش خودمختاری عامل‌ها

سازمان‌دهی سیستم‌های چندعاملی

ساختار سطح

انواع سازمان‌دهی در سیستم‌های چندعاملی

<p>پیمانه‌ای <i>Modular</i></p>	<p>جزئی از کل <i>Part of the Whole</i></p>	<p>سطح <i>Flat</i></p>	<p>سلسله‌مراتبی <i>Hierarchical</i></p>
<p>هر پیمانه خود می‌تواند یک سیستم چندعاملی باشد.</p>	<p>هر عامل جزئی از یک عامل بزرگ‌تر است.</p>	<p>همه‌ی عامل‌ها در یک سطح؛ امکان ارتباط مستقیم هر عامل با سایر عامل‌ها</p>	<p>ارتباط عامل‌ها در یک ساختار سلسله‌مراتبی</p>



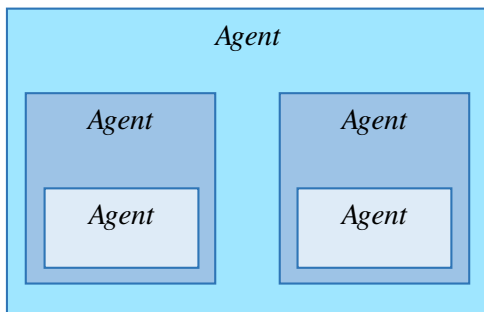
- مزیت: عدم تحمل هرگونه ساختار ایستا بر سیستم
- مزیت: امکان سازمان‌دهی پویا برای یک وظیفه‌ی خاص
- مزیت: عدم اعمال کنترل هیچ عاملی بر عامل دیگر
- مزیت: امکان خودمختاری کامل عامل‌ها و اهداف محلی آنها
- مزیت: پشتیبانی از باز بودن و توزیع‌شدگی سیستم
 - در صورت باز بودن سیستم: نیاز به سرویس مکان‌یابی
 - در صورت بسته بودن سیستم: عدم نیاز به سرویس مکان‌یابی (هر عامل در هر لحظه از موقعیت سایرین مطلع است.)
- عیب: سربار ارتباطی بالا

سازمان‌دهی سیستم‌های چندعاملی

جزئی از کل

انواع سازمان‌دهی در سیستم‌های چندعاملی

پیمانه‌ای <i>Modular</i>	جزئی از کل <i>Part of the Whole</i>	مسطح <i>Flat</i>	سلسله‌مراتبی <i>Hierarchical</i>
هر پیمانه خود می‌تواند یک سیستم چندعاملی باشد.	هر عامل جزئی از یک عامل بزرگ‌تر است.	همه‌ی عامل‌ها در یک سطح؛ امکان ارتباط مستقیم هر عامل با سایر عامل‌ها	ارتباط عامل‌ها در یک ساختار سلسله‌مراتبی



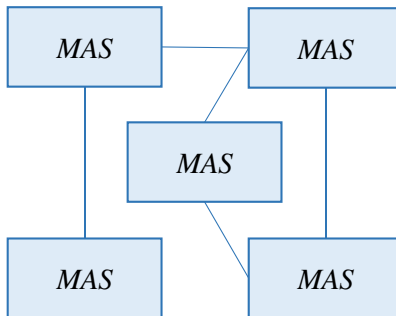
- عامل‌های جزء تحت کنترل عامل‌های کل هستند.
- مزیت: بالا رفتن کارایی در اثر ارتباط کنترلی محکم
- مزیت: کاهش سرباز ارتباطی
- عیب: محدود شدن سیستم به وظایف خوش‌تعریف نامنعطف

سازمان‌دهی سیستم‌های چندعاملی

پیمانه‌ای

انواع سازمان‌دهی در سیستم‌های چندعاملی

پیمانه‌ای <i>Modular</i>	جزئی از کل <i>Part of the Whole</i>	مسطح <i>Flat</i>	سلسله‌مراتبی <i>Hierarchical</i>
هر پیمانه خود می‌تواند یک سیستم چندعاملی باشد.	هر عامل جزئی از یک عامل بزرگ‌تر است.	همه‌ی عامل‌ها در یک سطح؛ امکان ارتباط مستقیم هر عامل با سایر عامل‌ها	ارتباط عامل‌ها در یک ساختار سلسله‌مراتبی



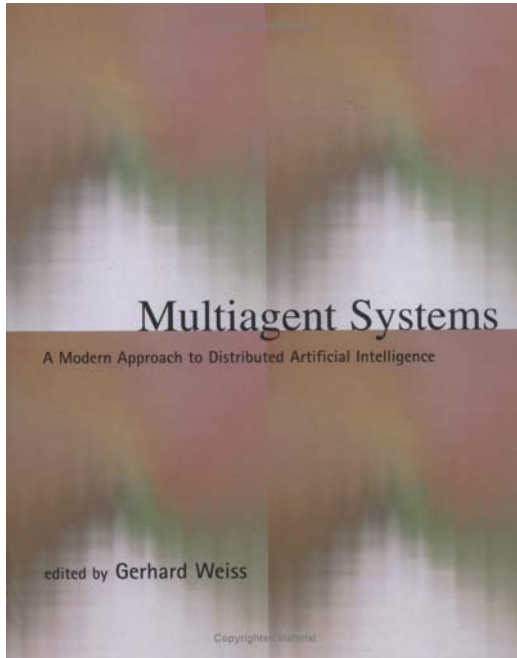
- سیستم چندعاملی از چند پیمانه تشکیل می‌شود که هر کدام به نوعی یک سیستم چند عاملی هستند.
- تقسیم‌بندی بر اساس نیاز بیرونی (مثل توزیع جغرافیایی) یا نیاز درونی (تاکید روی تعامل زیاد عامل‌ها و سرویس‌های داخل یک پیمانه)
- پیمانه‌های جدید می‌توانند در هر زمانی به سیستم موجود اضافه شوند.
- مزیت: بالا رفتن کارایی و کاهش سربار ارتباطی

سیستم های چند عاملی

جامعه‌ی عامل‌ها و سیستم‌های چندعاملی

۲

منابع



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