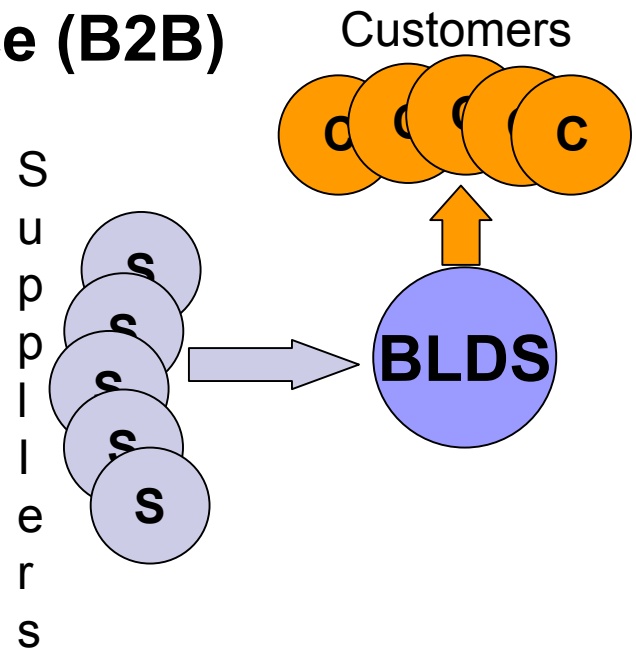
International Scientific–Practical Conference
on Programming (UkrPROG'2002)
May, 22-23, 2002, Kiev, Ukraine

Cooperation Layers in Agent-Enabled Business Process Management

Ermolayev, V. A., Plaksin, S. L.
Dept. of Mathematical Modelling and IT,
Zaporozhye State University
{eva, psl}@zsu.zp.ua (normal business hours)

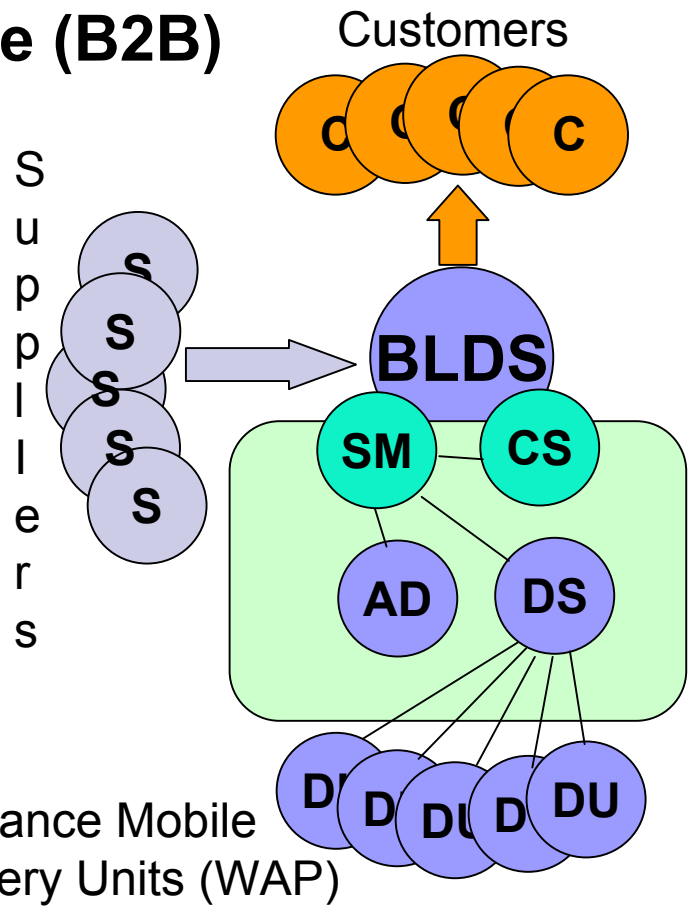
Business Lunch Delivery Service (B2B)

Clients: SME in a City district



Business Lunch Delivery Service (B2B)

Clients: SME in a City district



Business Lunch Delivery Service (B2B)

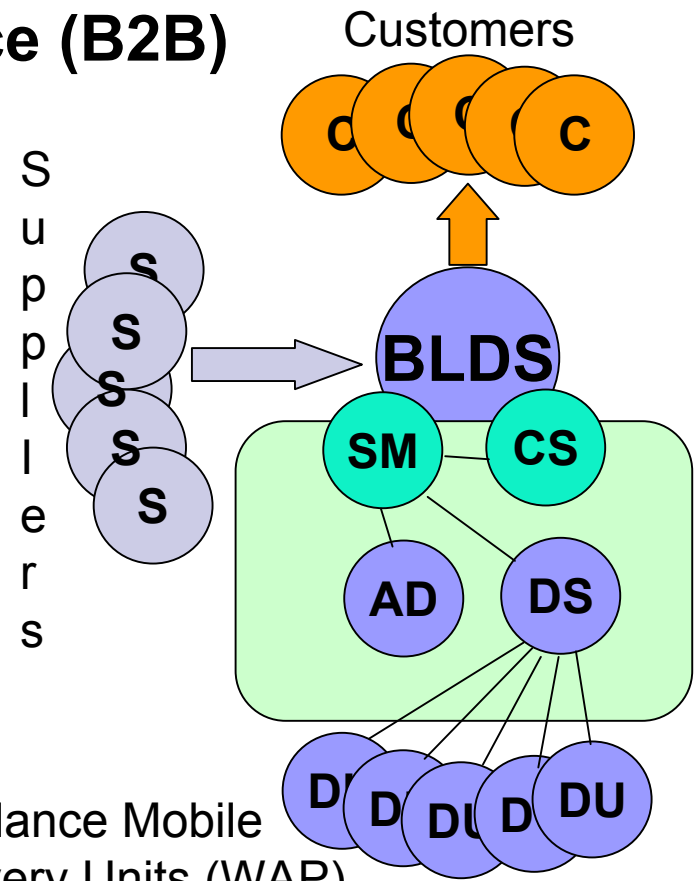
Clients: SME in a City district

Scenario is characterized by intrinsic distributedness, dynamic character and uncertainty:

- Not possible to plan the delivery statically (customer orders are not predictable, BLDS is an open organization - DU)

- No one is capable to perform and even to plan the whole delivery flow on its own (possibly a car repair or a speciality order will be required)

- Activities already allocated may result in failure (e.g., traffic, car is broken, the cooks are on strike, …) – corrective actions needed



Freelance Mobile Delivery Units (WAP)

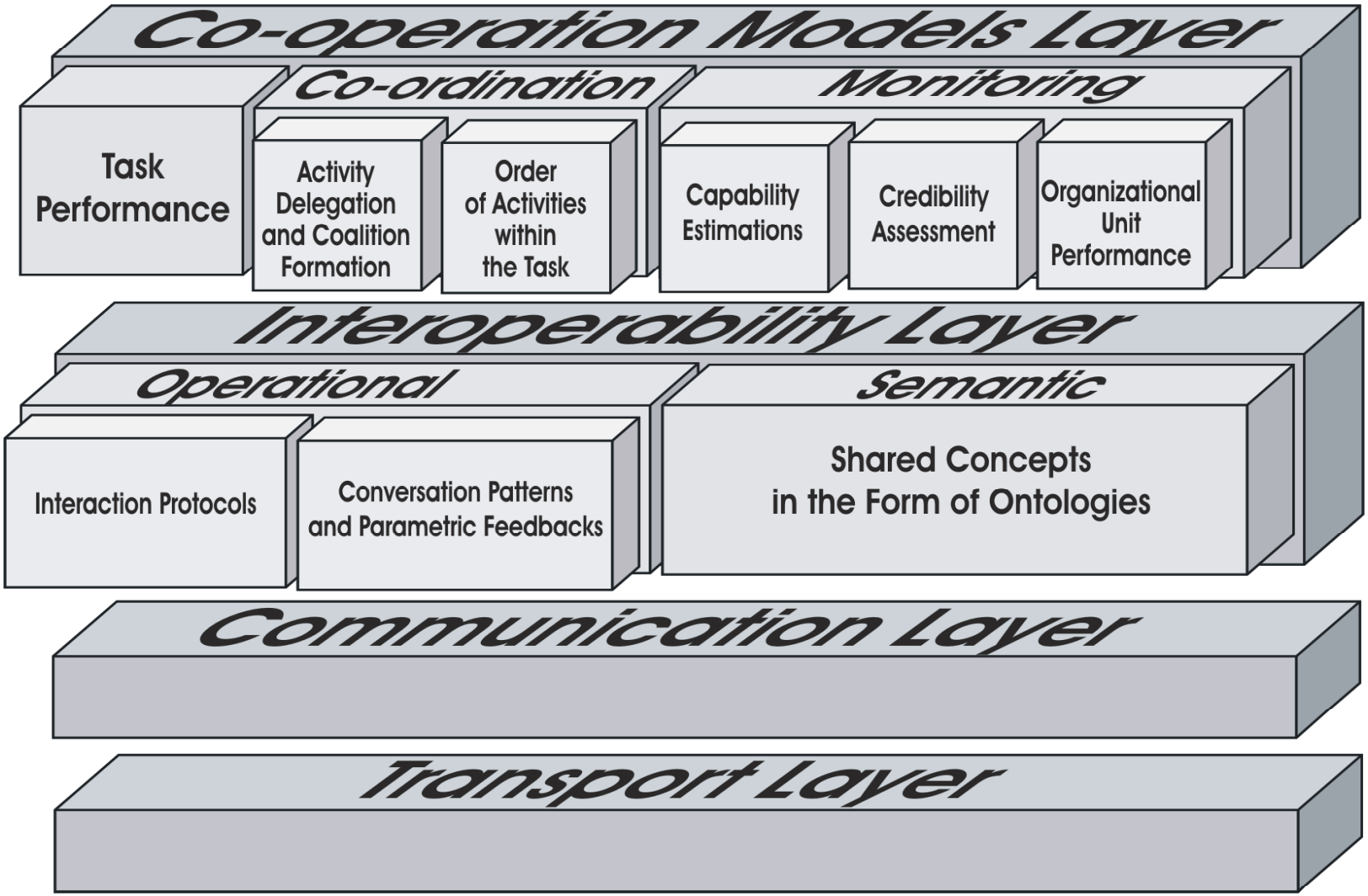
BLDS units: **self-interested**, Can't do without **cooperation**

Why agents: autonomy, situatedness, reactivity, proactivity, adaptability

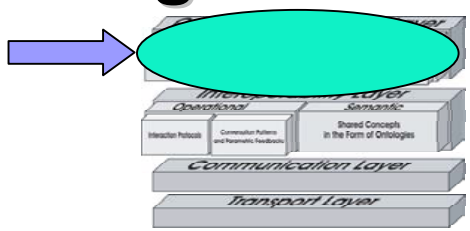
The Emphases of the Talk

- **Framework in a Nutshell:** A proposal of a Layered Approach to the design of agent-based architectures and distributed intelligent software systems for Business Process Management and Performance
- **Our contributions:** What have we done already to provide "plug-ins" to the Framework Slots at different layers
- **Widely accepted and standardized solutions:** What makes the framework open to heterogeneous solutions
- **Conclusions:** What are the results, the lessons learned?
- **Motivation:** Why the research in agent-enabled cooperative business process management and performance is important? – just few slides to justify the answer…

Proposal: Conceptual Framework

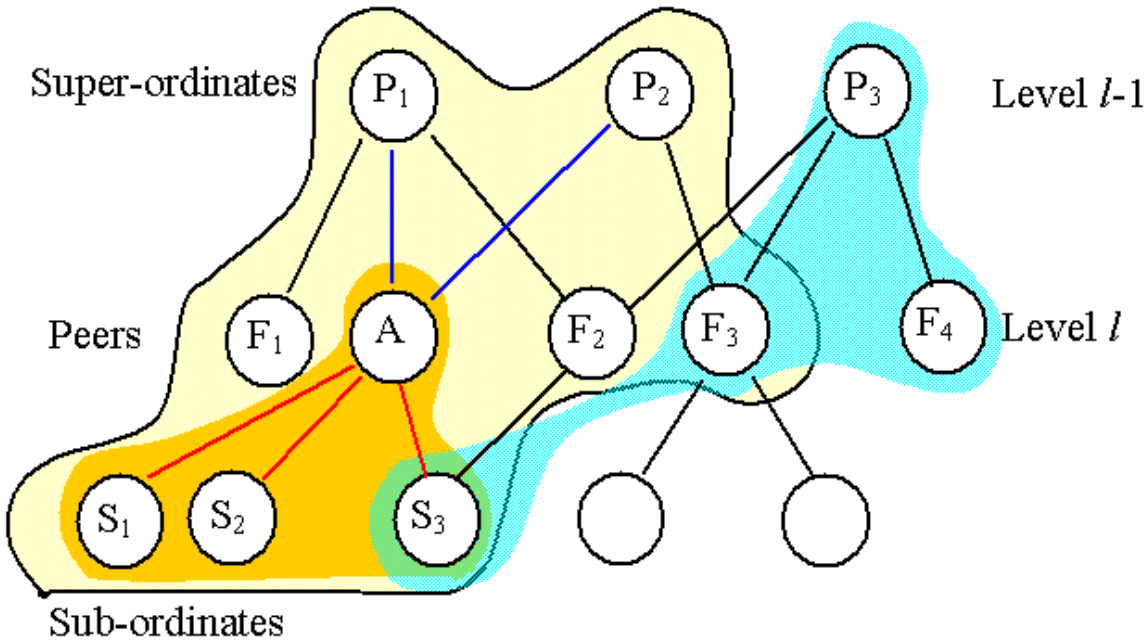


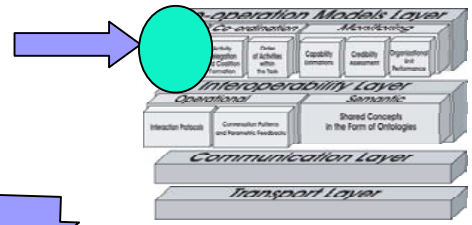
Cooperation Models Layer: Slots and Plug-ins

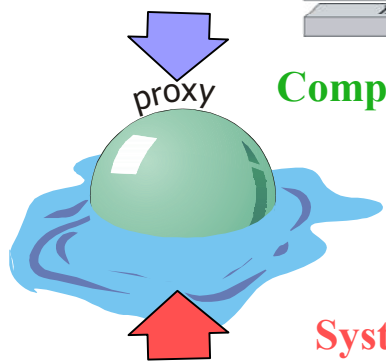


Slots	Plug-ins
Task Performance	- Organization Model - Functional System/Component Model - Task Model
Coordination	- Activity allocation and dynamic task coalition formation mechanism - Mechanism for coordination of the flow of activities within a task
Monitoring, learning from experience	- Fellow Capability Assessment Mechanism - Fellow Credibility Assessment Mechanism - Organizational Unit Performance Monitoring Mechanism

Cooperation Models Layer: Organization





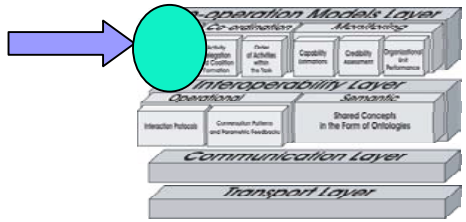


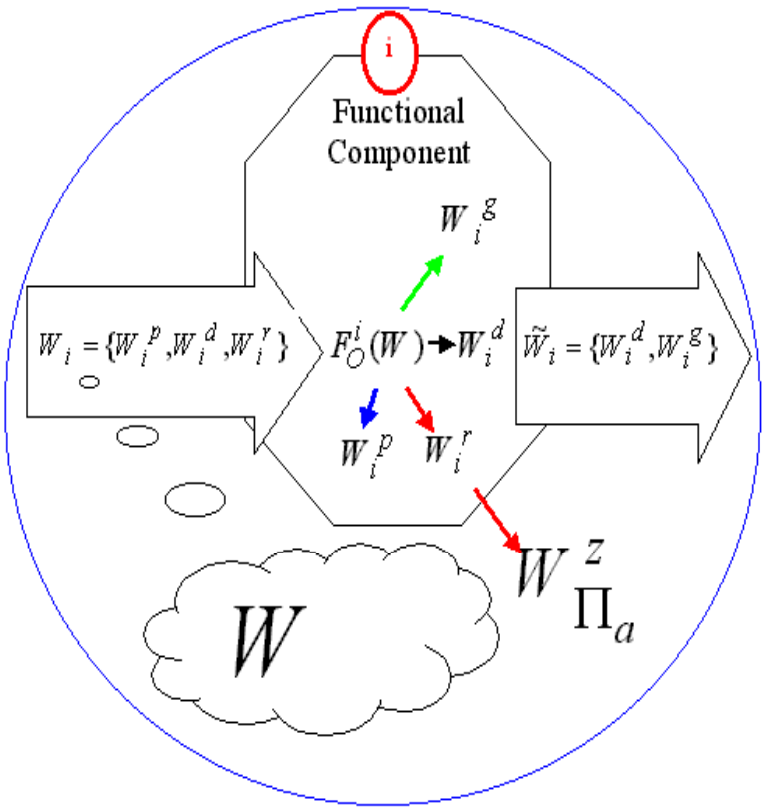
Higher-level Organizational Unit

Lower-level Organizational Unit

- On the higher level **Proxy** is viewed as a functional **component**. It expands to the **functional system** on the lower level of organization
- **Proxies** "wrap" respective **organizational units** (MAS) and are the representative members in the **higher level units** (MAS)

Cooperation Models Layer: Functional System

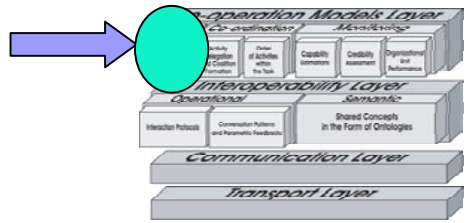




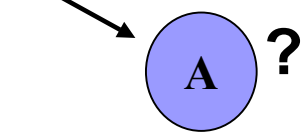
- Actors within organization/unit are considered to be functional (or reactive) components
- The same model is used for a functional system as far as an actor may expand into respective unit at the lower organizational level
- Capabilities are implemented as macro-model programs/methods one per activity
- A component may:
  - **Accept** incoming tasks/activities from the environment
  - **Generate** new tasks/activities in response to environmental events
  - **Reject** incoming tasks/activities
  - **Delegate** activities to subordinates or peers (allocate via negotiation or by directive)
  - **Perform** activities

Cooperation Models Layer: Task

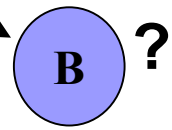
Task – partially ordered set of activities of varying granularity
Granularity: w^i for **A – atomic**, for **B - non-atomic**



$T = \{w^1, w^2, \ldots w^i, \ldots, w^k\}$



w^i



Cascade decomposition and execution by autonomous components at run-time

Task:

w^i - **atomic**
w^i - needs results of w^1, w^2
w^i - **Parameters** X^i **Results' Templates** Y^i comply to my understanding of w^i
w^i - not capable to perform myself
w^i - believe that **B**, **C**, **D** are capable - need to allocate
w^i - have the certain **Budget**, can delegate with the certain **Price**
w^i - results needed before the **Deadline**

Task:

w^i - **non-atomic**, comprises

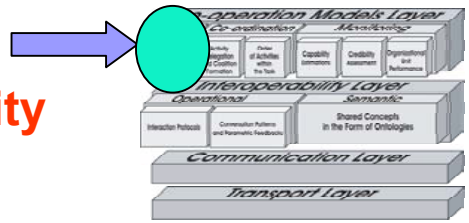
PLP

w_1^i myself
w_2^i need to delegate
w_m^i myself

w_1^i - can't perform now, need results of w_j^i, w_l^i before

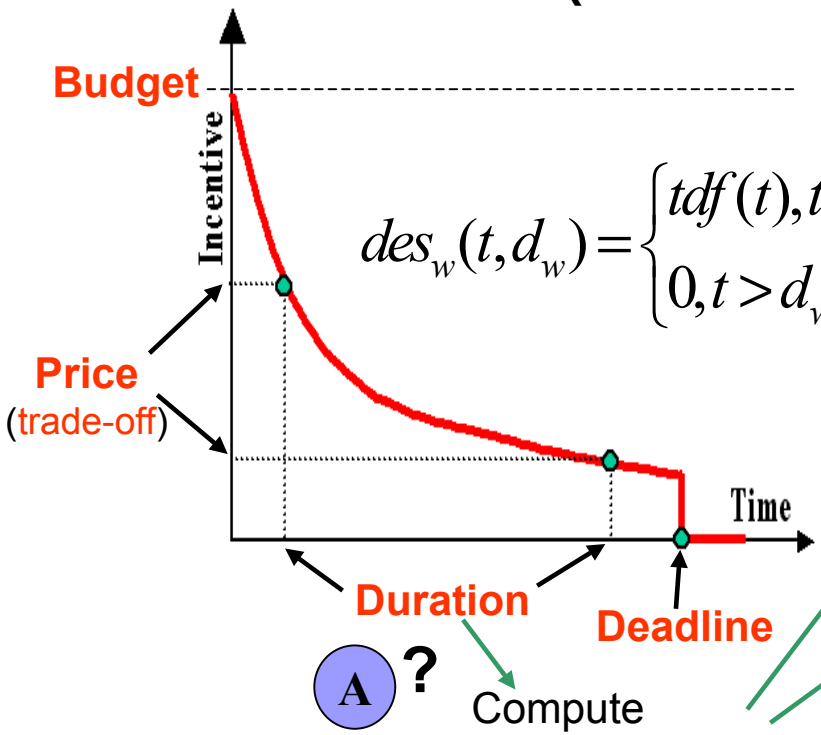
w_1^i - need to spend certain **Effort** to fit the **Deadline**

Cooperation Models Layer: Task

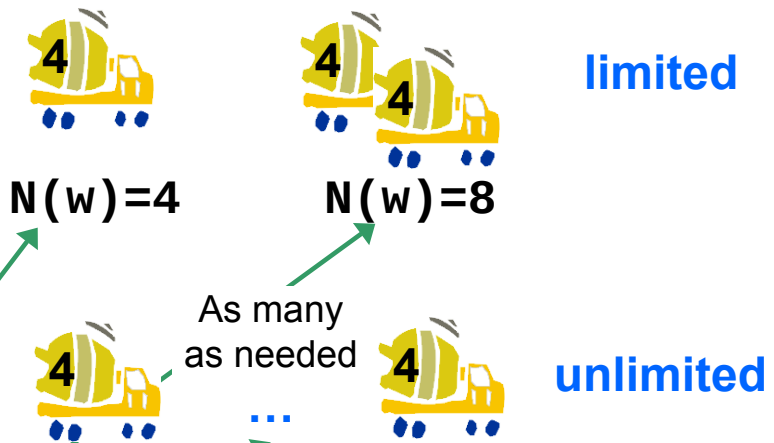


Activity Results' Desirability **Effort and Capacity**

`w=(DeliverPizza,X,Y)`



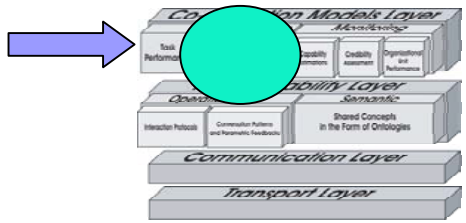
$$des_w(t,d_w)=\begin{cases} tdf(t), t \le d_w \\ 0, t > d_w \end{cases}$$



Capacity – how many pizzas can be delivered by **A's** delivery units to a certain location per unit time interval (eg.: 1 hour, 1 day…)

Effort (Capacity Share) – which part of **A's** delivery units will be busy to accomplish the activity within desired **Duration**

Cooperation Models Layer: Coordination



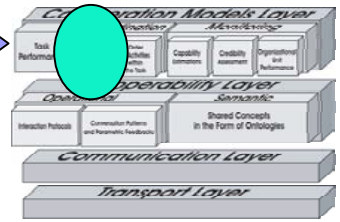
Coordination models (plug-ins):

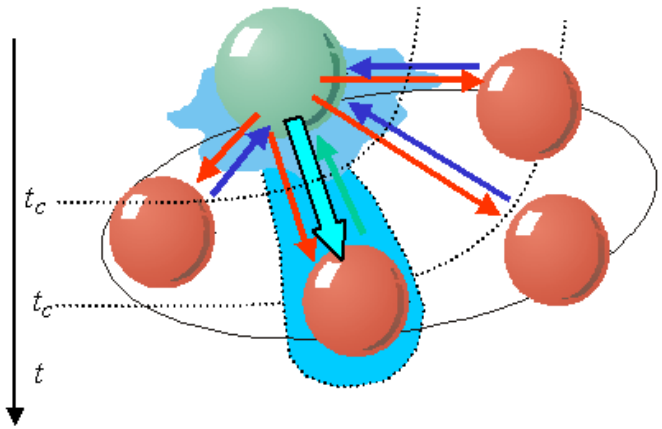
-Activity allocation and dynamic task coalition formation

-Coordination of the flow of activities within a task

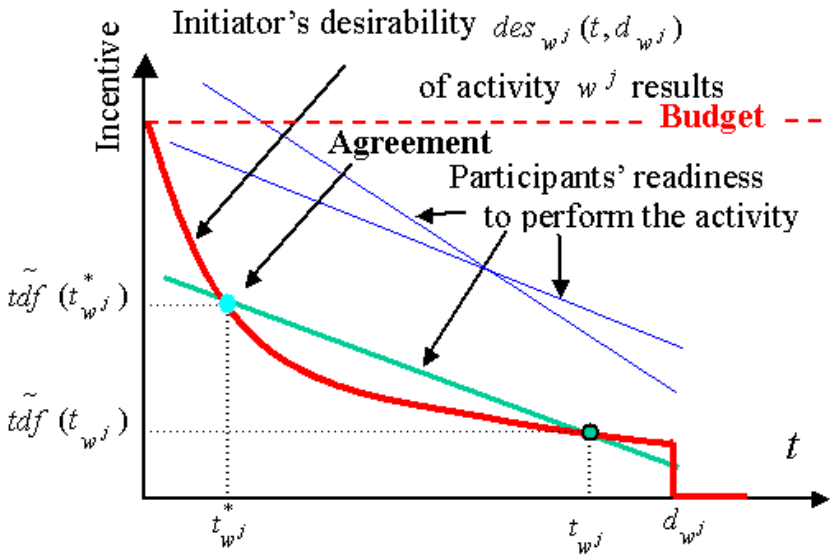
Cooperation Models Layer: Coordination

Activity Allocation and Dynamic Task Coalition Formation





a. Coalition Formation



b. Proposition and Feedbacks

Social laws:

-Relative cooperation commitment
-Activity arrangement convention
-Results delivery commitment

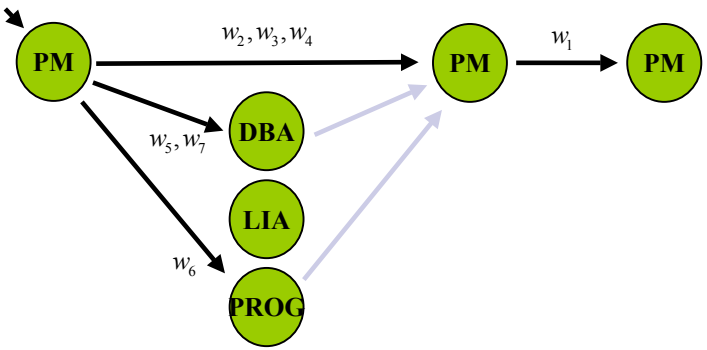
… authors are available
at a coffee break for details…

Even more details may be found at:
http://www.zsu.zp.ua/racing/list/e-pubs.htm

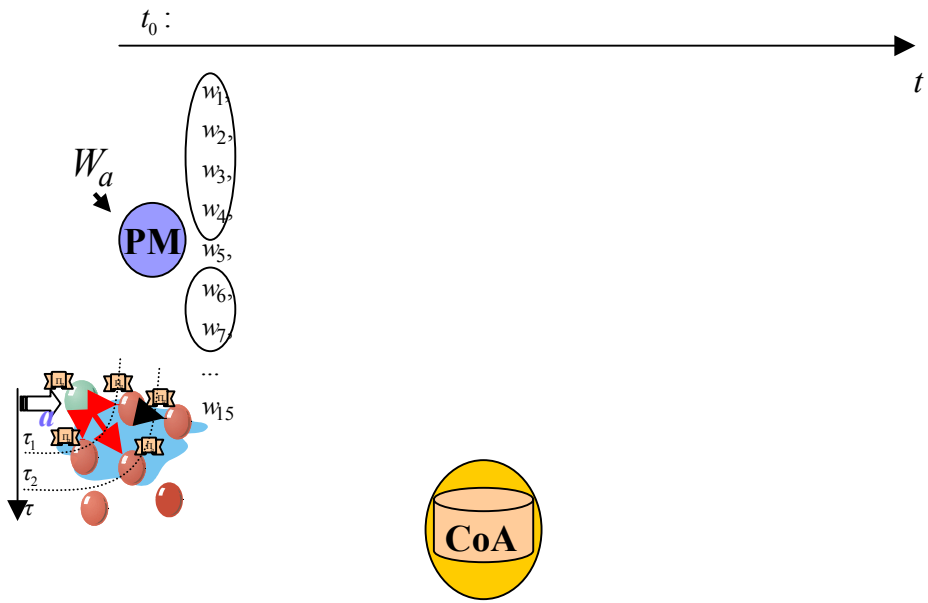
Coordination: Dynamic Task Coalition generates Workflow on the fly

Source: Our example from UkrPROG'2000 paper

Pre-planning: eg., WfMC PDL, Petri Net, ..



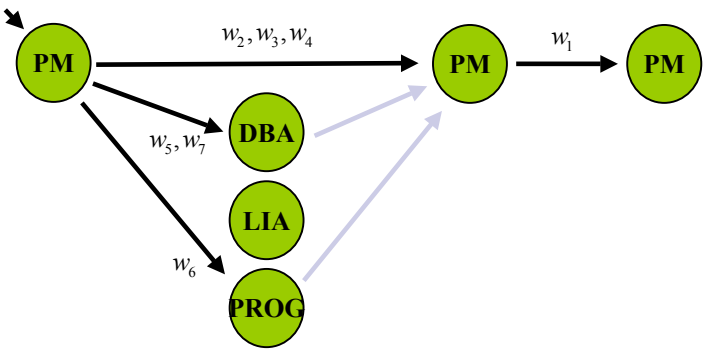
On-the-Fly: presented approach



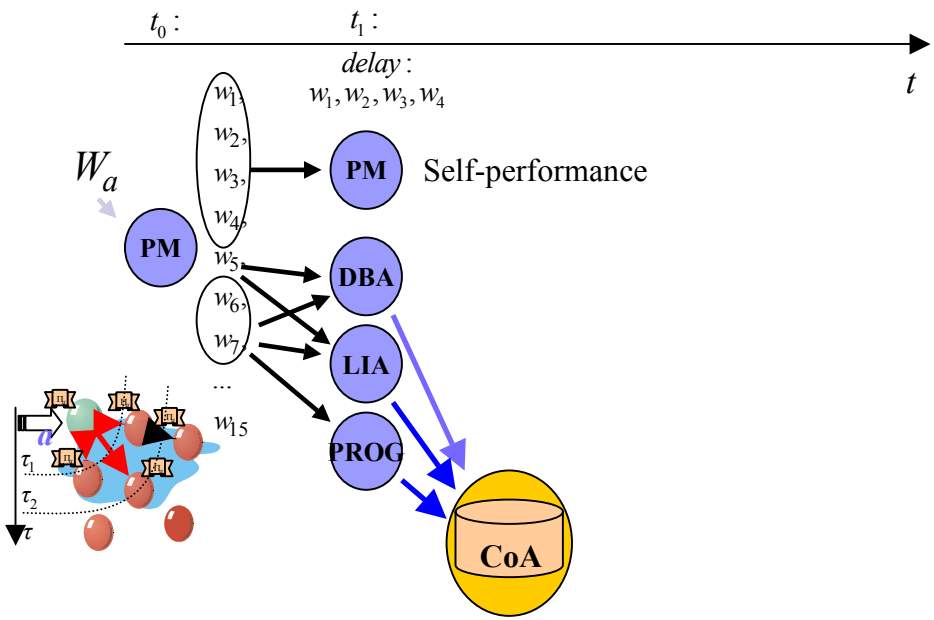
Coordination: Dynamic Task Coalition generates Workflow on the fly

Source: Our example from UkrPROG'2000 paper

Pre-planning: eg., WfMC PDL, Petri Net, ..



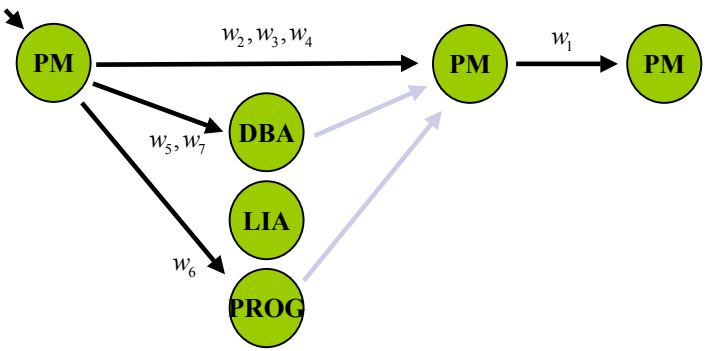
On-the-Fly: presented approach



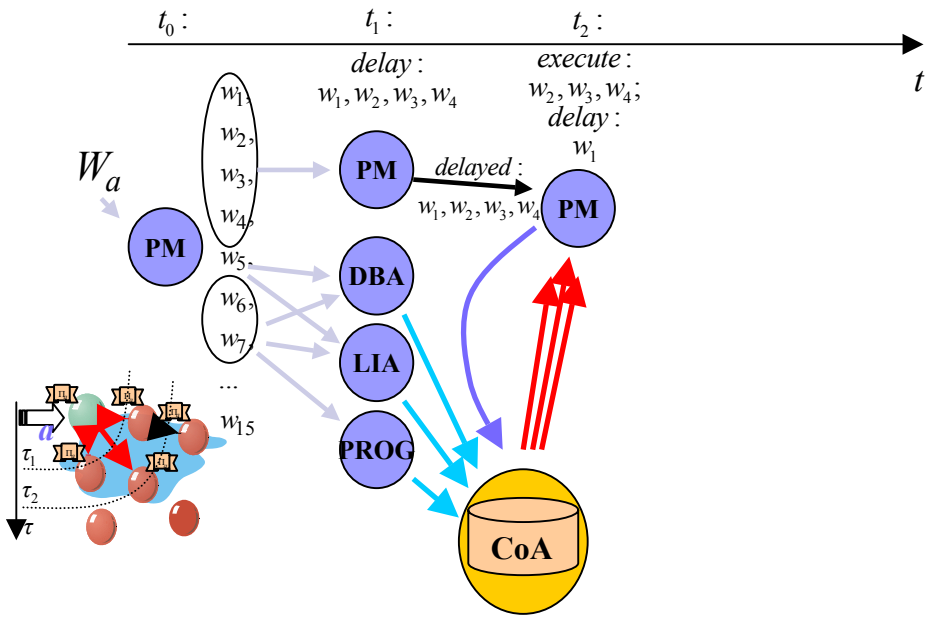
Coordination: Dynamic Task Coalition generates Workflow on the fly

Source: Our example from UkrPROG'2000 paper

Pre-planning: eg., WfMC PDL, Petri Net, ..



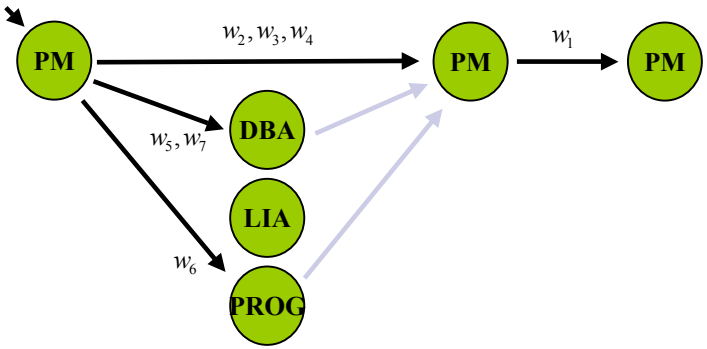
On-the-Fly: presented approach

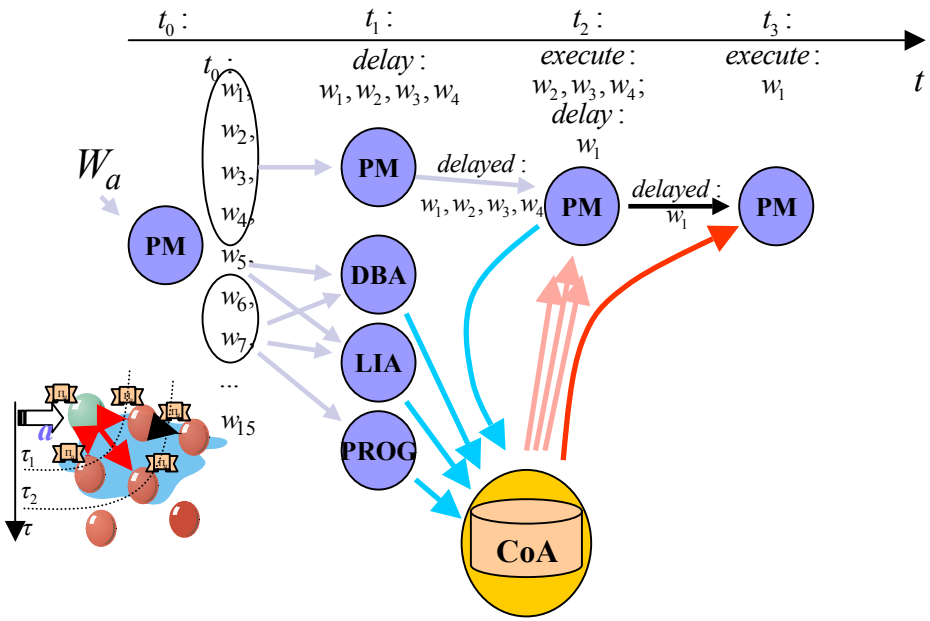


Coordination: Dynamic Task Coalition generates Workflow on the fly

Source: Our example from UkrPROG'2000 paper

Pre-planning: eg., WfMC PDL, Petri Net, … vice **On-the-Fly:** presented approach



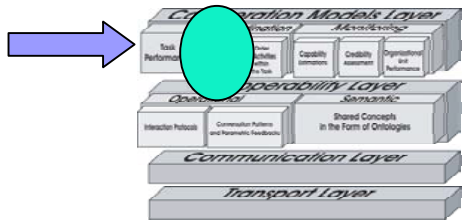


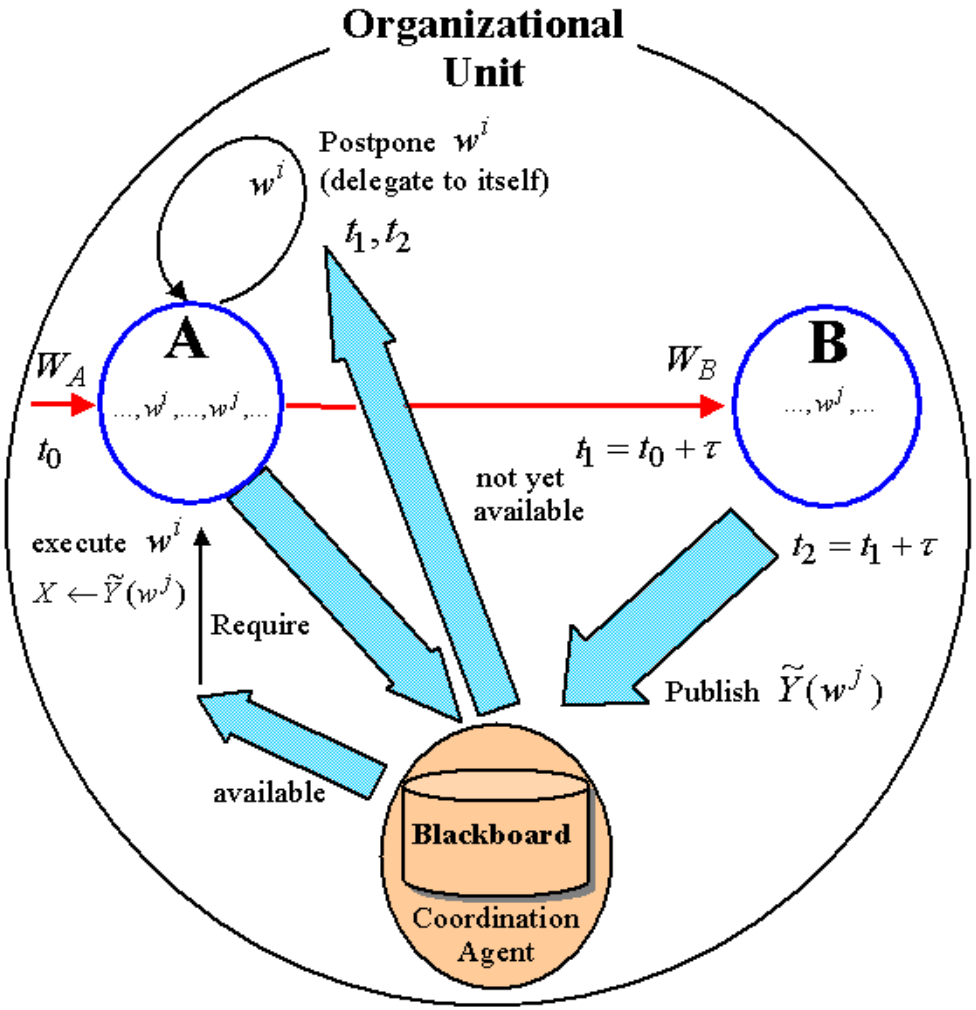
- Workflow is pre-defined long before the process has started
- No means to consider the current executive's state, workload, capacity, capability, trustworthiness, rational interest
- Predefined workflow plans are far from being effective

- Workflow is developed step-by-step in the course of task execution by coalition of distributed autonomous rational actors
- Optimal workflow branch (activity performer) is chosen each time as the result of contracting negotiations

Cooperation Models Layer: Coordination

Coordinating the flow of Activity Performance within a Task

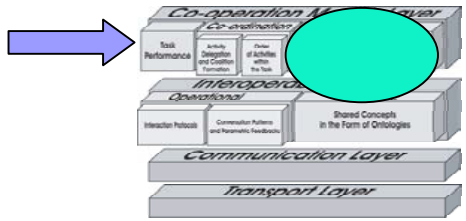




- LINDA-like Tuple Space coordination model is used as the basics
- A dedicated Utility Coordination Agent manages the process
- Activities which need the results of other activities as parameters are postponed until necessary data is published to the Blackboard

… love to discuss the details at a coffee break …

Cooperation Models Layer: Monitoring



Plug-ins:

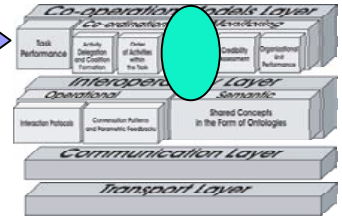
-Adjusting Social Behavior: Fellow Capability Assessment

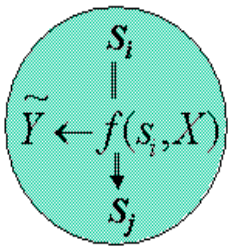
-Adjusting Social Behavior: Fellow Credibility Assessment

-Organizational Unit Performance Monitoring: further adaptability to typical tasks

Cooperation Models Layer: Monitoring

Adjusting Social Behavior: Fellow Capability Assessment





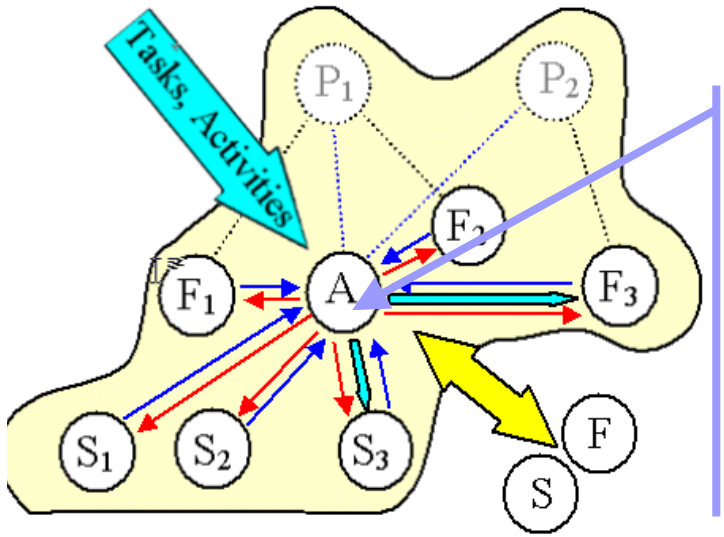
$S_i, S_j \in S_A$

$s_i = \{r(X_A), q(F_A), t(F)\}$

$r(X_A)$ - *parameter constraints*

$q(F_A)$ - *work constraints*

$t(F)$ - *transition function*



Advertise

Propose (Accept/Reject)

Allocate

$$\mathbf{C} = \begin{array}{c} \\ F_1 \\ \dots \\ F_n \\ S_1 \\ \dots \\ S_m \end{array} \begin{array}{c} \begin{array}{ccccc} w^1 & \dots & w^j & \dots & w^k \end{array} \\ \begin{bmatrix} c_1^1 & & c_1^j & & c_1^k \\ & & \dots & & \\ & \dots & c_i^j = (q_i^j, p_i^j) & \dots & \\ & & \dots & & \\ c_{n+m}^1 & & c_{n+m}^j & & c_{n+m}^k \end{bmatrix} \end{array}$$

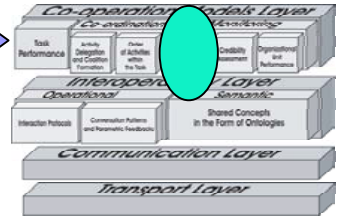
1. $p_i^j \leftarrow p_i^j + \frac{r}{q_i^j}$,
2. $q_i^j \leftarrow q_i^j + 1$

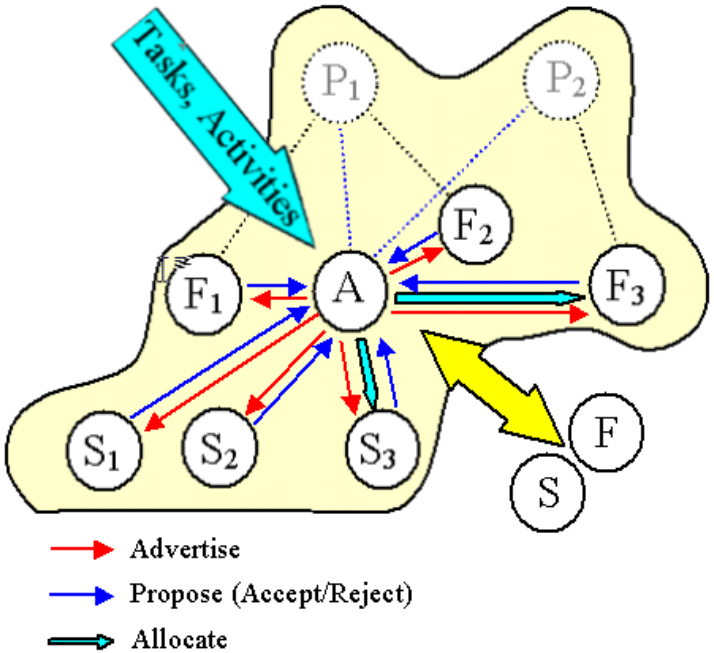
$r =$
- 0 if the fellow rejected the activity,
- 0.5 if the fellow replied that it can accept the activity
- 1 if the activity was finally delegated to the fellow

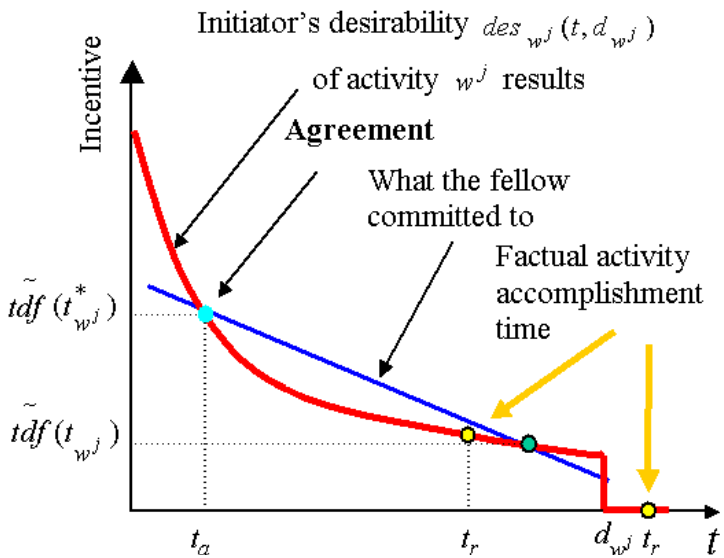
- Activities are advertised to the fellows possessing relevant capabilities
- Knowledge about changing fellows' capabilities is adjusted dynamically
- Agents benefit from cooperative work by adjusting their beliefs about the fellows

Cooperation Models Layer: Monitoring

Adjusting Social Behaviour: Fellow Credibility Assessment







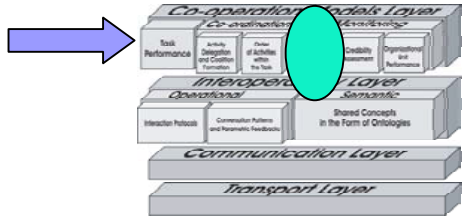
$$Cr_{i,j} := Cr_{i,j} \times \begin{cases} 1, t_r \le t_a \\ p_{w^j}(t_a / t_r), t_a < t_r \le d_{w^j} \\ 0, t_r > d \end{cases}$$

Social laws:
-Relative cooperation commitment
-Activity arrangement convention
-Results delivery commitment

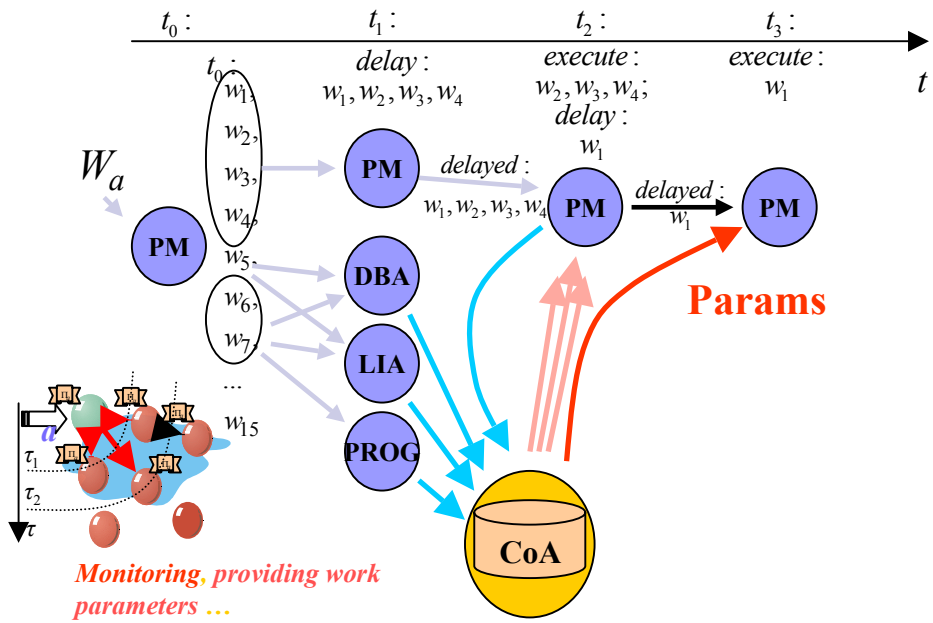
Fellows with higher credibility value w.r.t. the certain activity have better opportunities to get the next contract and, thus, to increase their own utility

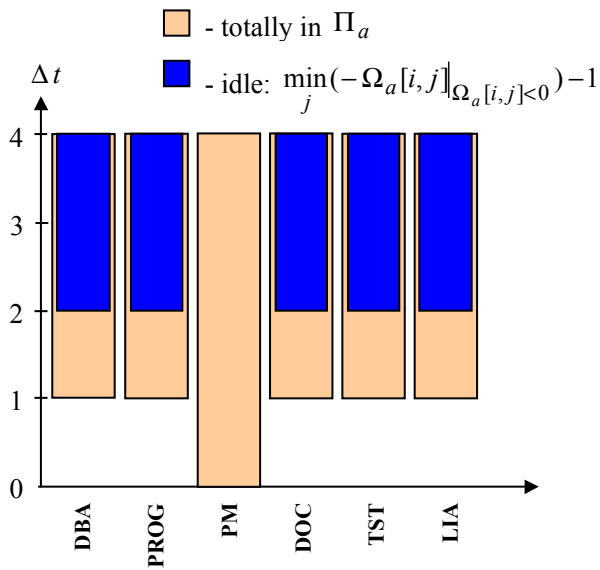
Cooperation Models Layer: Monitoring

Organizational Unit Performance Monitoring:



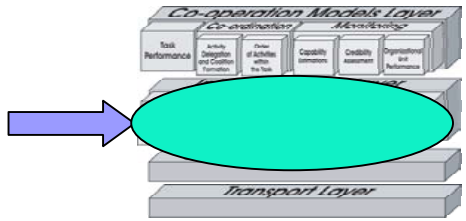
Source: Our example from UkrPROG'2000 paper





- Monitoring information (rejected activities, idle state durations) is collected by Coordination Agent
- It may be further used by human administrators to fine-tune the organization by adjusting agents' capabilities, capacities, organizational units' staff
- Organization thus becomes more optimized to the performance of the typical tasks

Interoperability Layer:



Operational:

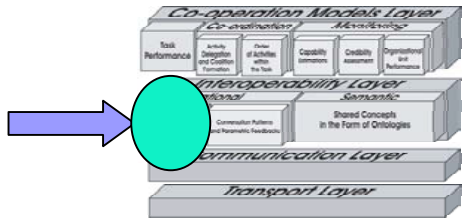
- Interaction Protocols
- Conversation Patterns

Semantic:

- Ontologies

Interoperability Layer: Operational

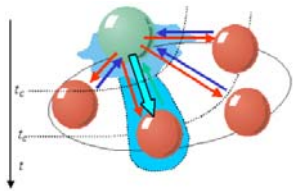
Interaction Protocols

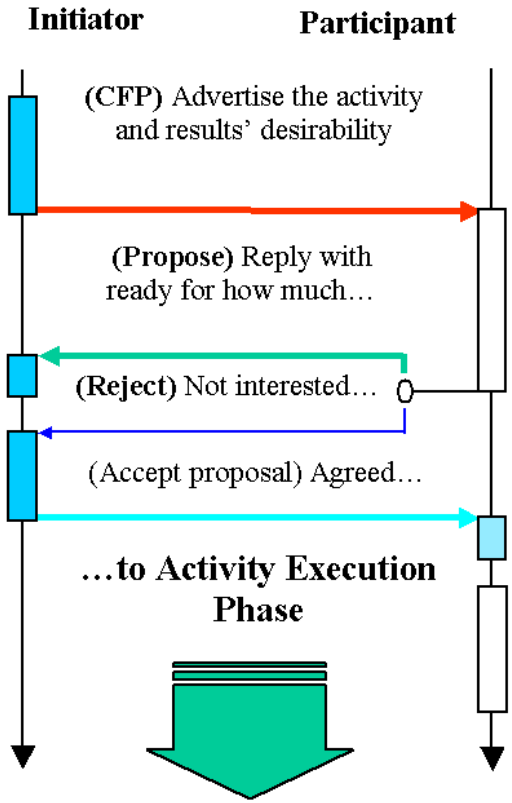


-Any relevant widely accepted interaction protocol (e.g., FIPA) providing the common frame for inter-agent operation may be used to facilitate to agents cooperative task performance

-It is considered that a protocol versus a conversation pattern is a more complex and a more purpose-specific construct and may be assembled of conversation patterns and communicative patterns (performatives) of the Communication Layer

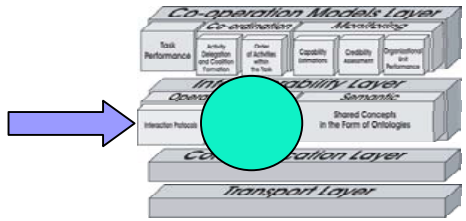
-Slightly modified FIPA Contract Net Protocol was used to arrange negotiations on activity allocation

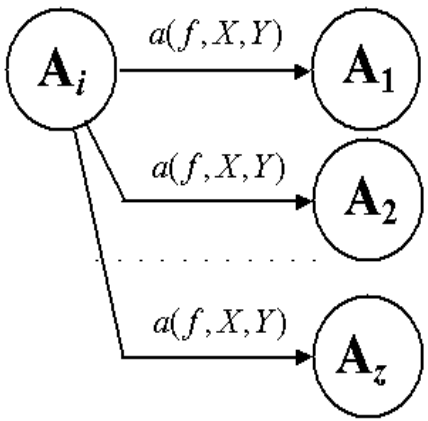




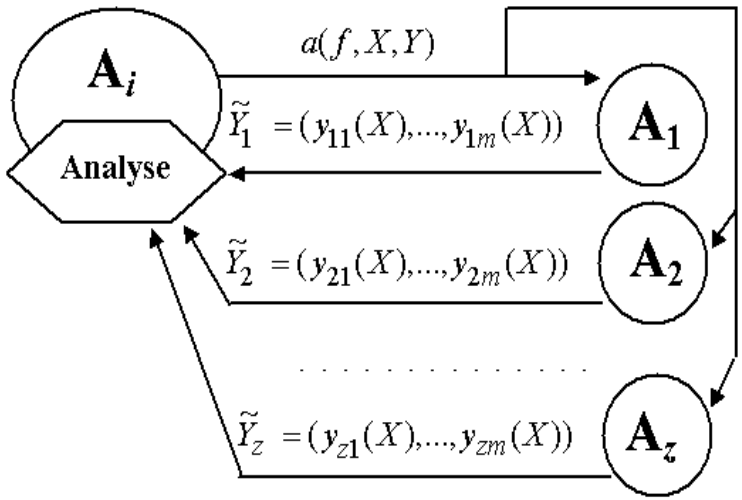
Interoperability Layer: Operational

Conversation Patterns and Parametric Feedbacks

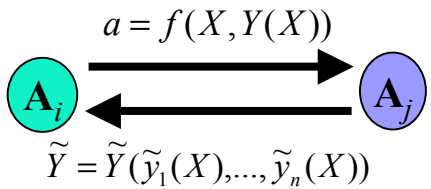


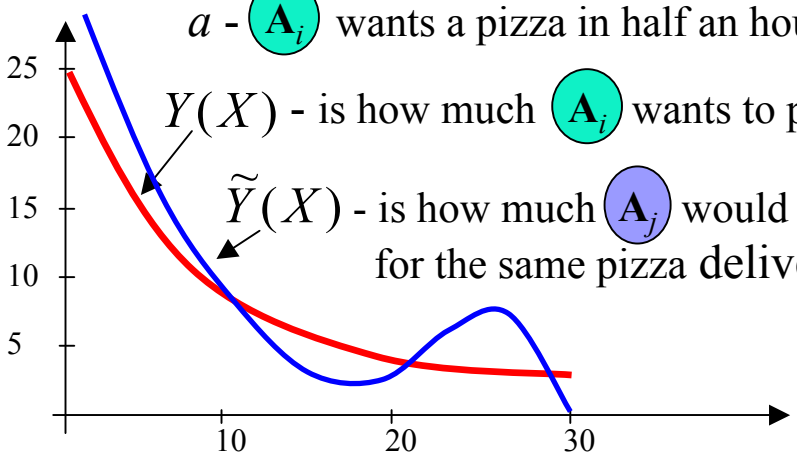


a. a directive



b. a parametric query with results analysis





a - A_i wants a pizza in half an hour from A_j

$Y(X)$ - is how much A_i wants to pay for pizza delivery in time

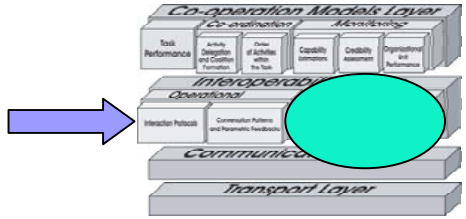
$\tilde{Y}(X)$ - is how much A_j would like to receive for the same pizza delivery

Parametric feedbacks –

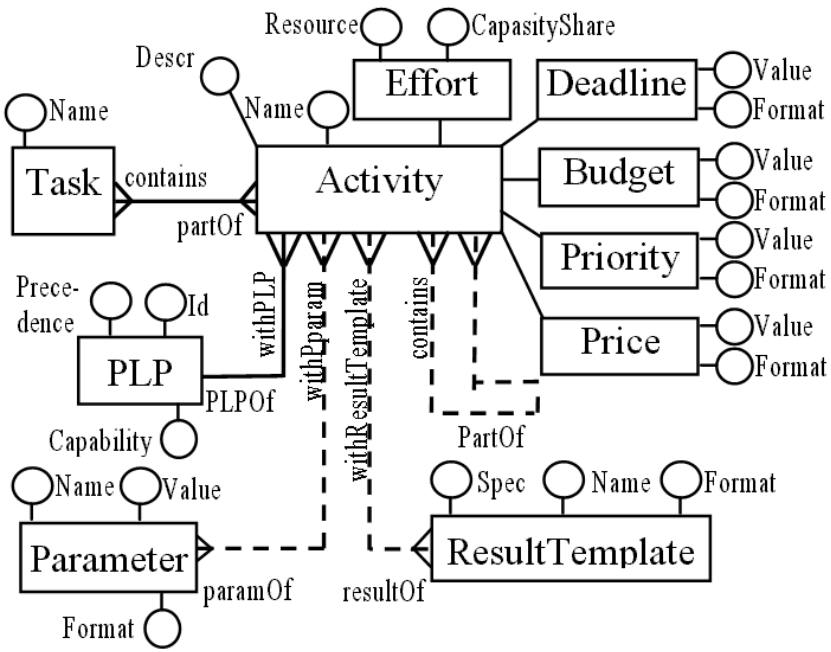
expressed **capability** and **commitment** to perform requested activity (service) at a **certain state** with respect to the **self-interest of a service provider**

Interoperability Layer: Semantic

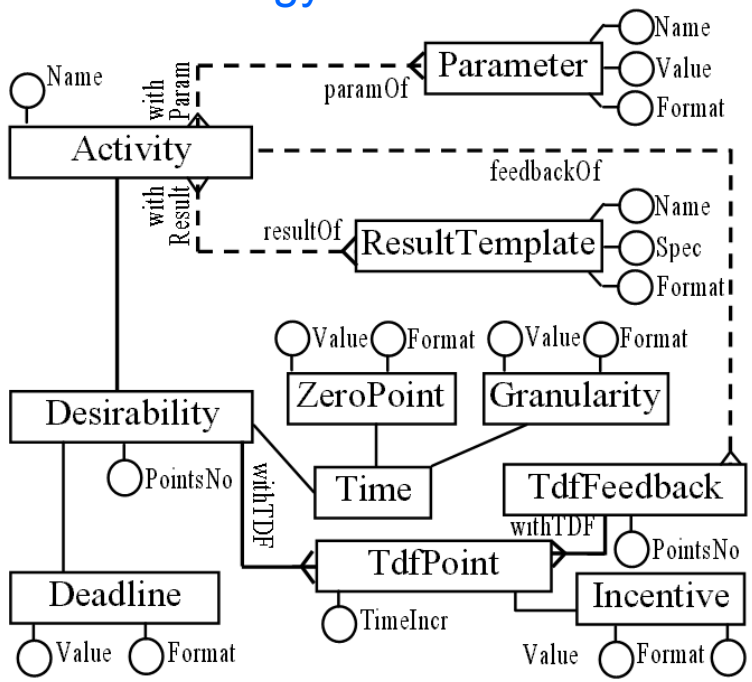
Ontologies - shared common specification of a conceptualization



Task Ontology



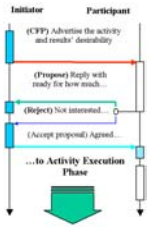
Negotiation Ontology



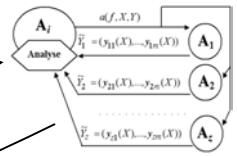
OilEd 2.2a[1] and FACT[2] reasoner were used for ontologies design and expressiveness check. OIL, RDFS, DAML and SHIQ versions of Task and Negotiation Ontologies are available at http://eva.zsu.zp.ua/eva_personal/ontologies/

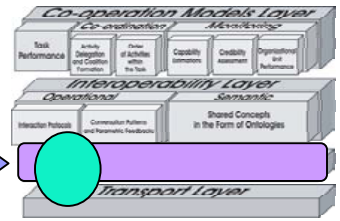
[1] http://img.cs.man.ac.uk/oil/, [2] http://www.cs.man.ac.uk/~horrocks/FaCT/.

Communication Layer:

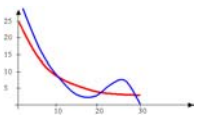


KQML performatives ask-one and tell used for conversation patterns and (further on) for CNP protocol construction





A_i T={('DeliverPizza', X, Y)} A_j

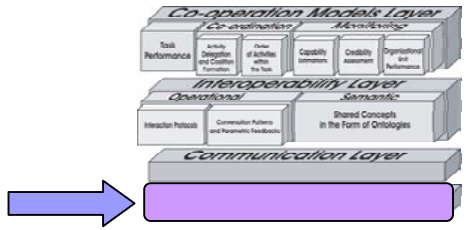


A_i Ready for how much A_j

```
(ask-one
:sender              "I3"
:receiver            "M"
:in-reply-to         Null
:reply-with          DeliverPizza-TDF
:language            (XML)
:ontology            (Negotiation)
:contents            (
<Desirability>
  <Activity> <Name>DeliverPizza</Name> </Activity>
  <Deadline> <Value>23.05.2002/20.00</Value>
             <Format>datetime</Format> </Deadline>
 <Time> <ZeroPoint>  <Value>27.10.2001/08.00</Value>
                     <Format>datetime</Format></ZeroPoint>
        <Granularity><Value>2</Value>
                     <Format>hours</Format></Granularity>
 </Time>
 <PointsNo>6</PointsNo>
 <TdfPoint> <TimeIncr>0</TimeIncr> <Incentive><Value>25</Value>
                   <Format>Money</Format></Incentive> </TdfPoint>
 …
 <TdfPoint> <TimeIncr>30</TimeIncr> <Incentive><Value>5</Value>
                   <Format>Money</Format></Incentive> </TdfPoint>
</Desirability>       )
)
```

```
(tell
:sender              "M"
:receiver            "I3"
:in-reply-to         DeliverPizza-TDF
:reply-with          Null
:language            (XML)
:ontology            (Negotiation)
:contents            (
<TdfFeedback>
 <activity> <name>DeliverPizza</name> </activity>
 <PointsNo>2</PointsNo>
 <TdfPoint>
   <TimeIncr>   </TimeIncr> <Incentive><value>        </value>
                            <format>Money</format></Incentive>
 </TdfPoint>
 <TdfPoint>
   <TimeIncr>   </TimeIncr> <Incentive><value>        </value>
                            <format>Money</format></Incentive>
 </TdfPoint>
</TdfFeedback>
)  )
```

Transport Layer:



Transport Layer elements should be strictly standardized to ensure wide acceptance and usage in open systems with heterogeneous components

A general consensus on the transport environment is that it should provide the shell for agent **naming**, **location** and **message delivery** mechanisms.

According to **FIPA Transport Service Reference Model** Agents in an open organization are bound to **Agent Platforms** (AP) and exchange messages via the **Transport Services** of their AP-s.

The mechanism FIPA proposes as the standard to cope with various network protocols is the use of the **message Envelopes**.

For the moment FIPA has provided specifications for **IIOP** and **WAP** protocols

Conclusions: Results and Lessons Learned

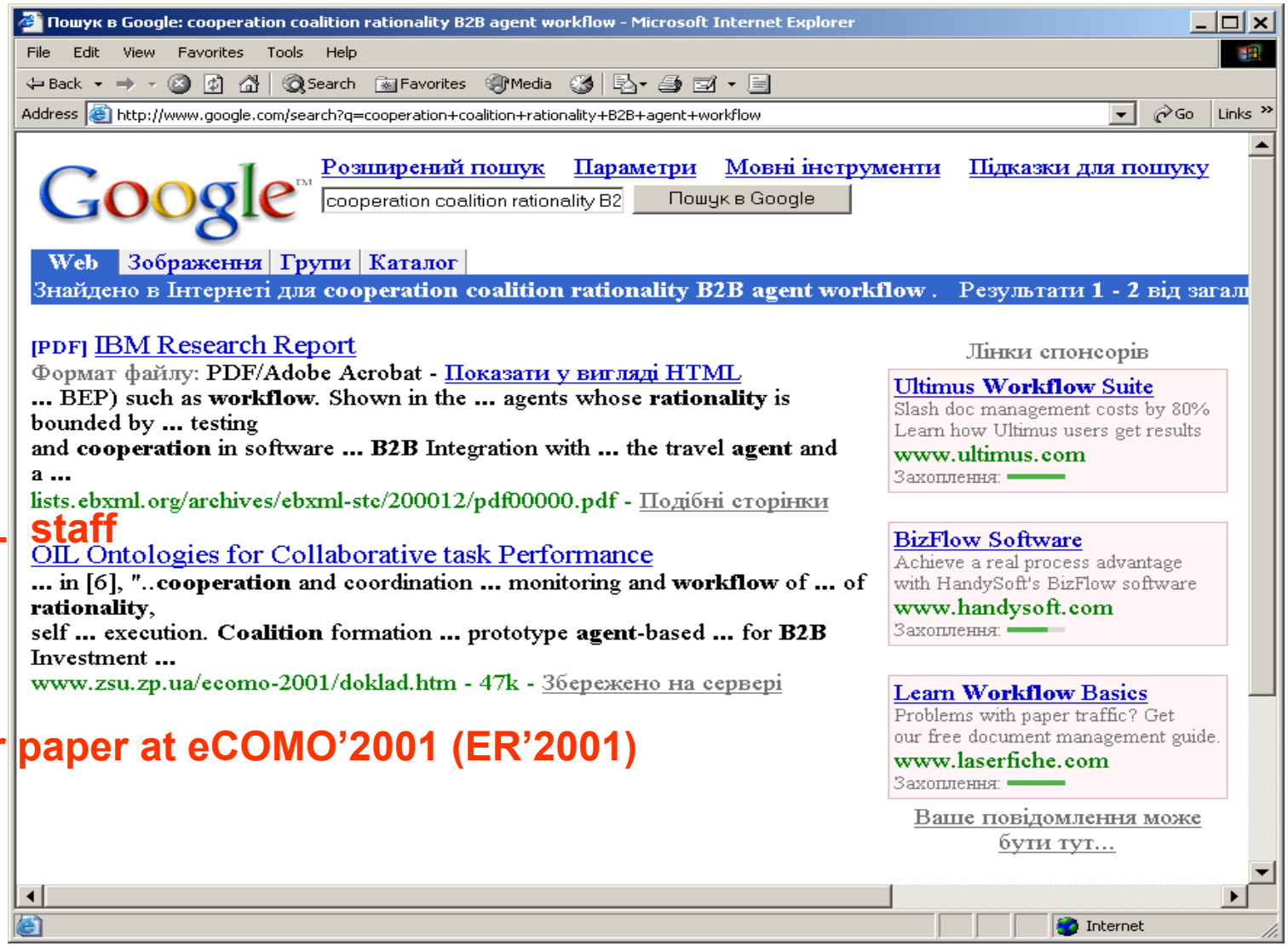
- Cooperation while performing business processes by autonomous, distributed actors possessing rational, uncertain and, sometimes, contradictory behaviors within an open organization is rather a complex utility
- There are still lots of open issues in the domain: e.g., the lack of widely accepted consensus on how all this staff should be structured and organized
- The contribution of the presented research is the proposal of a four-layer formal cooperation framework for agent-enabled business process management
- The paper is not so ambitious as to claim the final solid word in the domain, but rather to analyze the trends, to try to put it to the reasonable places within a conceptual hierarchy
- Presented results to some extent prove that there is some sense in the proposed layering, especially in the domain of business process management and performance
- Review of related work (in the paper) provides no vital contradictions to the presented layering proposition

Domain Keywords:

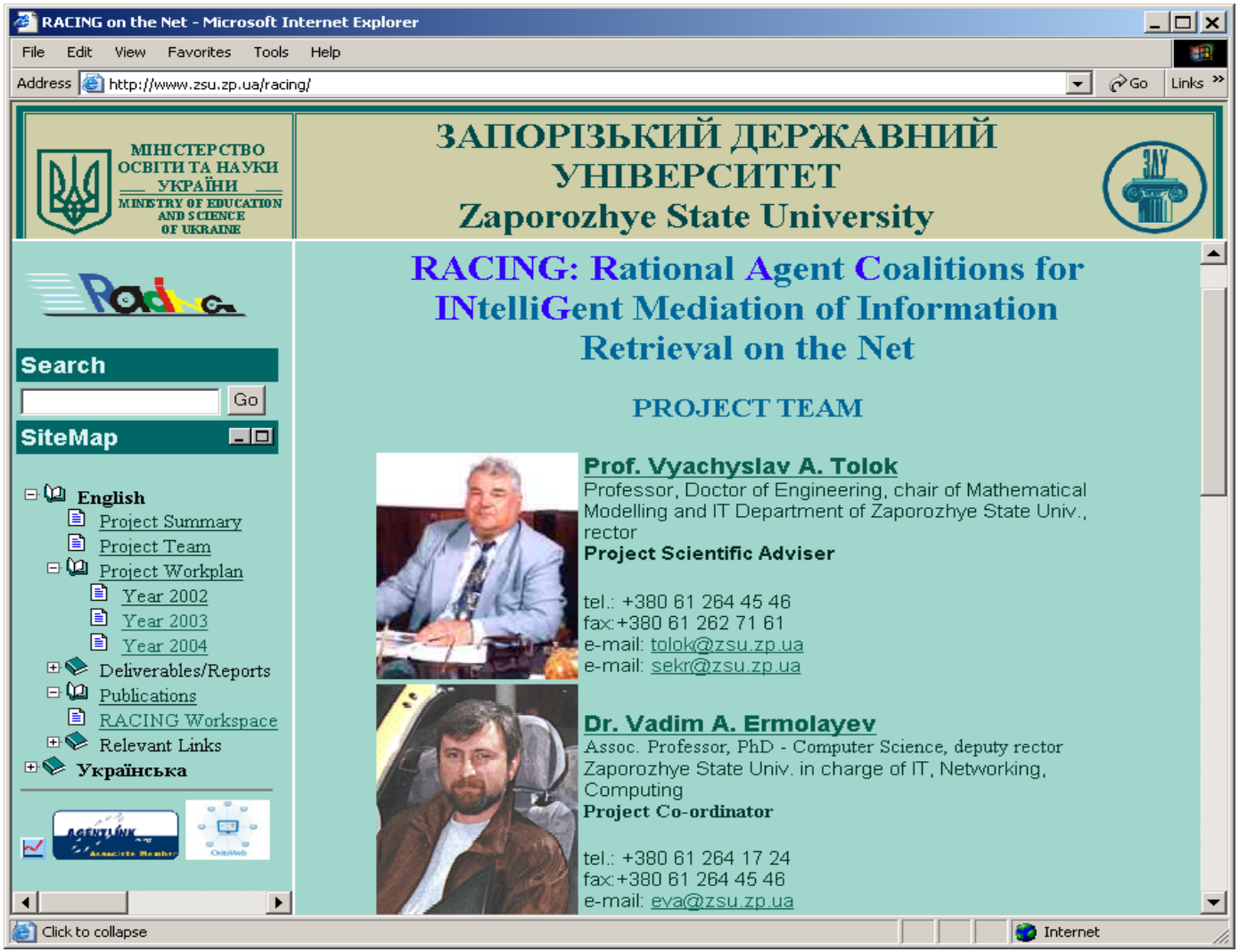
-Cooperation
-Coalition
-Rationality
-B2B
-Agent
-Workflow

Google search with these keywords returns: …

…funny, but: The Domain is still HOT for Further Research



RACING: towards the next step ahead – http://www.zsu.zp.ua/racing/



… Just filling in Google blank spaces