

**APPLICATION OF DIAKOPTICAL  
MAS FRAMEWORK TO PLANNING PROCESS  
MODELLING**

*Borue, S. U., Ermolayev, V. A. Tolok, V. A.*

Dept. of Mathematical Modelling and Information Technologies,

Zaporozhye State University,  
66, Zhukovskogo st., 330600, Zaporozhye, Ukraine,  
tel/fax:+380 61 264 17 24,

E-mail: [bsu,eva,tolok@zsu.zaporizhze.ua](mailto:bsu,eva,tolok@zsu.zaporizhze.ua)

**What is presented:**

- Agent-Based theoretical Framework  
for modelling business processes  
as the processes of information interchange among  
the members of dynamically formed, scalable  
and evolving intellectual agent communities
- Possible Applications and Application Domains
- Case study of the modelling  
of one of practically important types  
of business processes - a planning process

### **What is presented (continued):**

- Case study analysis showing:
  - that the framework is practically applicable to modelling of planning processes
  - the benefits of the methodology
- The hypothesis that the framework is applicable to another types of business processes

### **What is NOT presented (to fit within 30 min):**

- Framework Models in details  
(Generic Agent Model, Community Model,  
Communication and Co-Ordination Models,  
Evolution Model)
- Architectural Solutions
- Symantical, Ontological, Introperability issues
- Implementation issues

### Assumptions:

**Business Process** in real life is the procedure of doing something practically important and performed in time by means of **Co-operated and Co-ordinated Activities** of various **Autonomous Entities** having their own **Specialisations**

**Business Process** is modelled as a process of the execution of the set (*Task*) of **Atomic Works**

by the **Agents** -- the members of the task **Community**  
**Atomic Works** are merely the **Actions** (*Policies*) the Agents are capable and autorised to perform having certain **information** as their **sensory input**

and producing some **information as the result**

**Business Process** is therefore understood as the **Process of Information Interchange**

### Application Domains (outline)

**- Intelligent Information Discovery and Integration (eg. DARPA I\*3, ...)**

«...Information agents are computational software systems that have access to multiple, heterogeneous and geographically distributed information sources. One of their main **tasks** is to perform active searches for relevant information in non-local domains on behalf of their users or other agents.

This includes **retrieving, analyzing, manipulating, and integrating** information available from multiple **autonomous information sources...**»

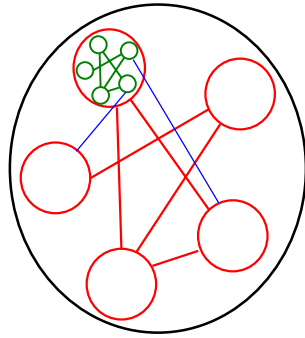
*...Technological Roadmap of the ESPRIT AgentLink I2A SIG*

**- Enterprise Modelling**

Processes of information interchange representing Real Business processes for real/virtual enterprise management, decision making support, manufacturing and supply chain simulation and control, EC applications, traffic control, ....

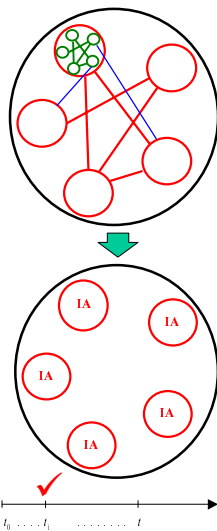
**- Processes of Monitoring in various domains (eg. Environment (Pollution) Control, ...)**

### Basic Idea: Traditional Approaches (Pitfalls)



- based on the usage of static structured collections of rigid relationships among the participating functional nodes (eg. Petri Nets, Task execution plans, ...).
- do not fully cover the things we deal with in real life: brainstorming, informal discussion, negotiation,...
- rigid relationship models are not really scalable.

### Basic Idea: Proposed Approach



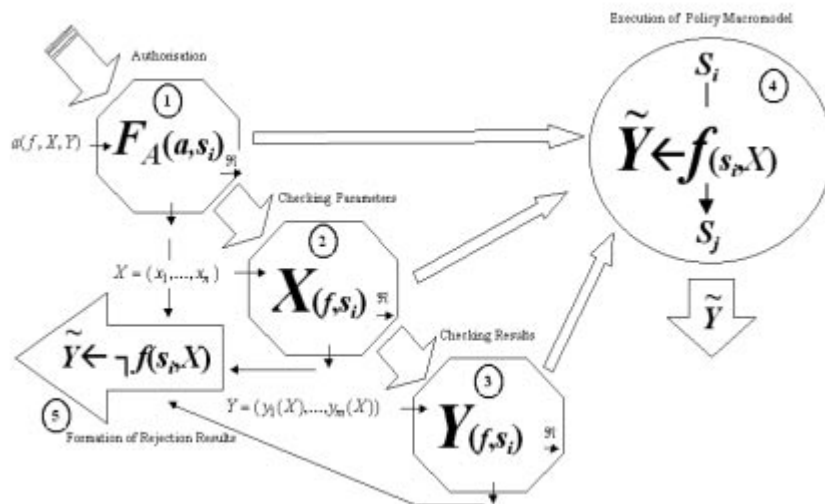
**Diacoptical framework** for modelling of the processes of information interchange among the distributed components (intellectual agents) of a dynamic information system:

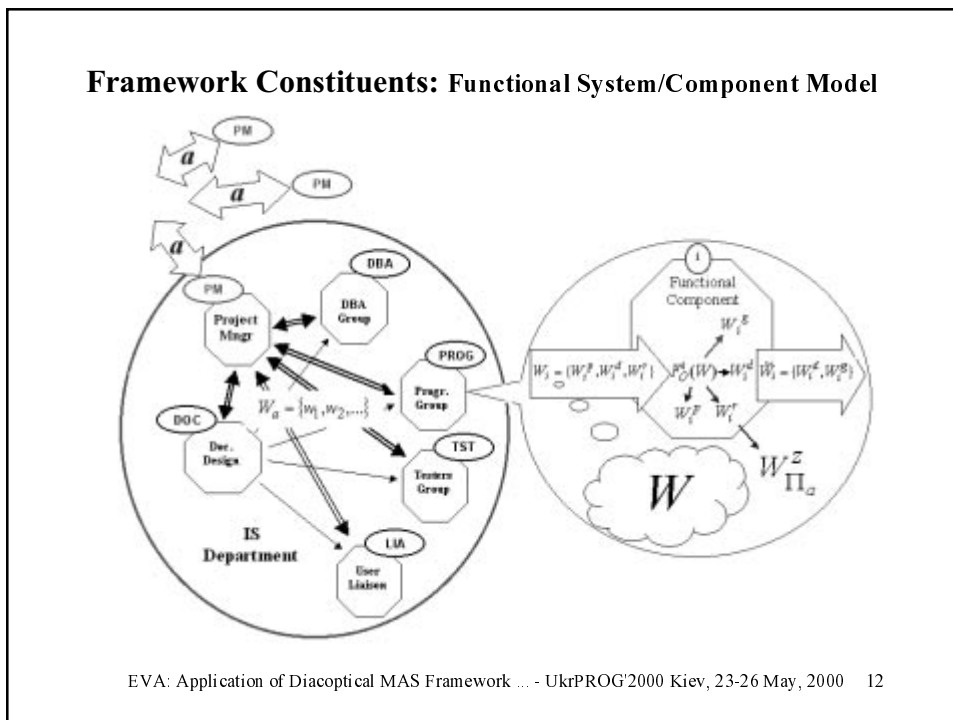
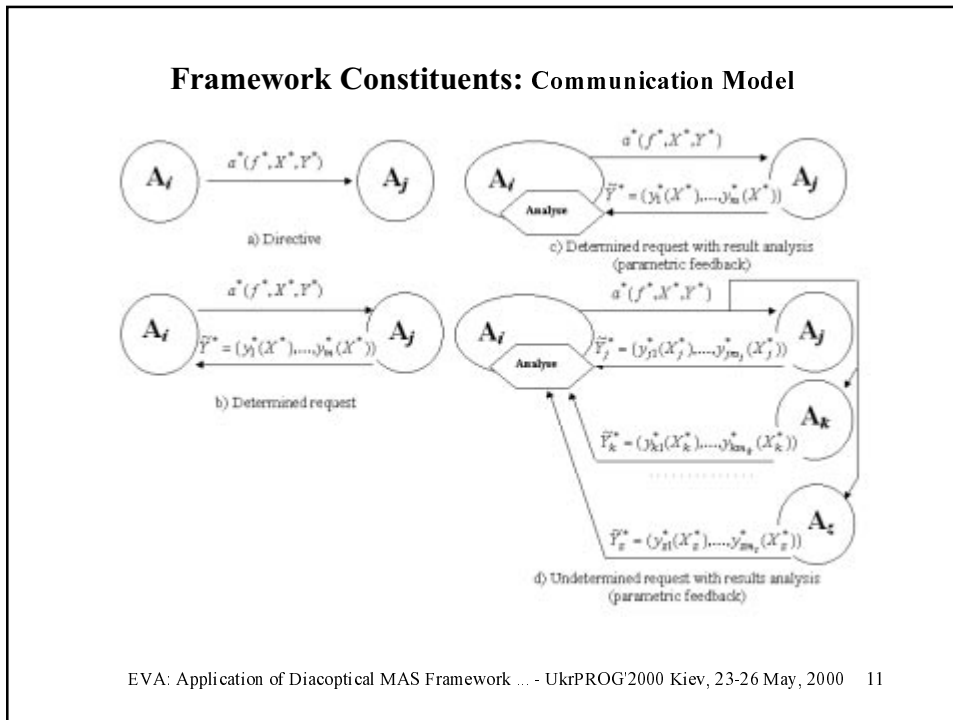
- **Main focus:** the **associations** between participating nodes/actors **are soft and are changing in time**
- Representation of a system and system components by means of the **functionally equivalent, unified and simplified model** (reactive intellectual agents with their behaviour models presented by macromodel programs)
- Components' interrelationship **topology** is embedded into component **macromodel**
- Components' **specialisation** is embedded into component **macromodels**

**Framework Constituents:**

- **Generic Agent's Model**
- **Communication Model**
- **Functional System/Component Model**
- **Process Model**
- **Co-ordination Model**
- **Evolution Model - (will be presented at IOI'200)**

**Framework Constituents: Generic Agent's Model**





### Framework Constituents: Process Model

The diagram illustrates the process model for  $\Pi_a$ . It shows a process  $\Pi_a$  (represented by a cylinder) interacting with various tasks. A task  $W_a$  is shown as a large arrow pointing towards the process. Smaller tasks  $W_j^g$  (generation),  $W_j^d$  (absorption),  $W_j^p$  (performance), and  $W_j^z$  (inexecutable) are shown as arrows between components. A timeline at the bottom shows discrete time points  $t, \dots, t_{n+1} = t_n + \Delta t, t_n, \dots, t_0$ .

Process  $\Pi_a$  starts with generation of the new task  $W_a \subseteq W$ . Task  $W_a$  as well as the additional tasks  $\tilde{W}_a$  are considered to be linked to process  $\Pi_a$  and labelled with the unique identifier of this process. The component is considered to be **linked to process**  $\Pi_a$  in case it has absorbed the part of  $W_a, \tilde{W}_a$ , or has generated  $W_a^g$ . Process  $\Pi_a$  is **completed** in case all the components stopped to absorb the atomic works of the tasks linked to process  $\Pi_a$ . The set of works  $W_{\Pi_a}^z$  not absorbed in  $\Pi_a$  is denoted as the set of **inexecutable** works. Task execution is modelled as a two-level process performed sequentially at discrete time points  $t_n, t_{n+1} = t_n + \Delta t$ .

EVA: Application of Diacoptical MAS Framework ... - UkrPROG'2000 Kiev, 23-26 May, 2000 13

### The rules for Component State and System State Matrixes generation

**$K_i$  is generated by each component:**

- $k_{lj}, :$  **1** - component  $i$  allocates work  $w_j$  to component  $l$ ,  $l \neq i$
- 0** - otherwise
- $k_{ij}^:$  **-1** - component absorbs work  $w_j$  within interval  $[t_n, t_n + \Delta t]$
- 1** - component  $i$  allocates work  $w_j$  to itself,
- 0** - component  $i$  is not capable to perform ... within  $[t_n, t_n + \Delta t]$

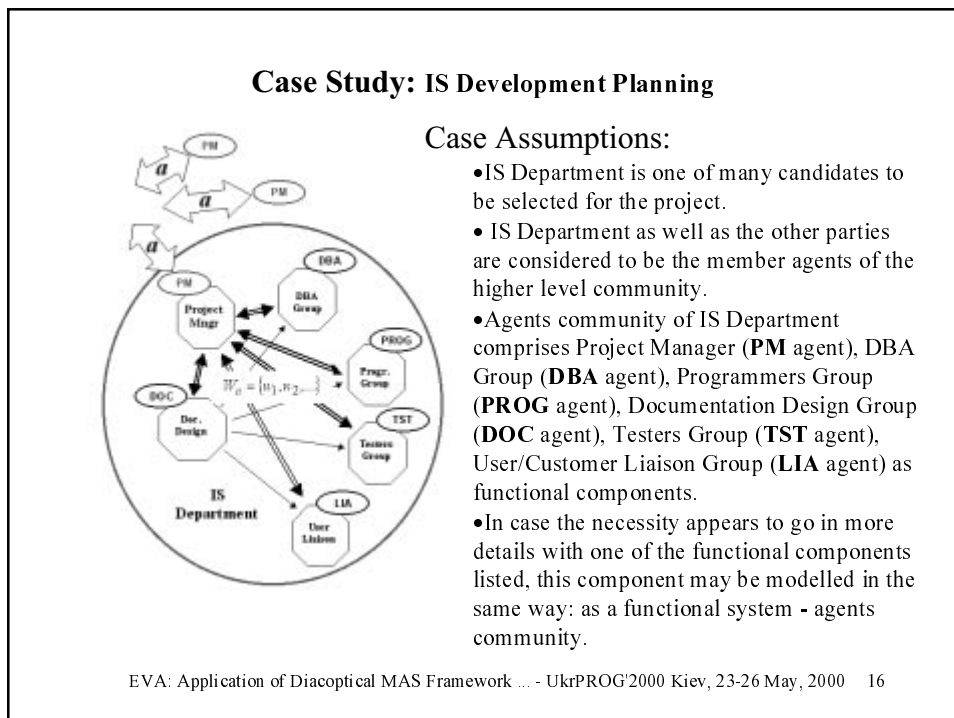
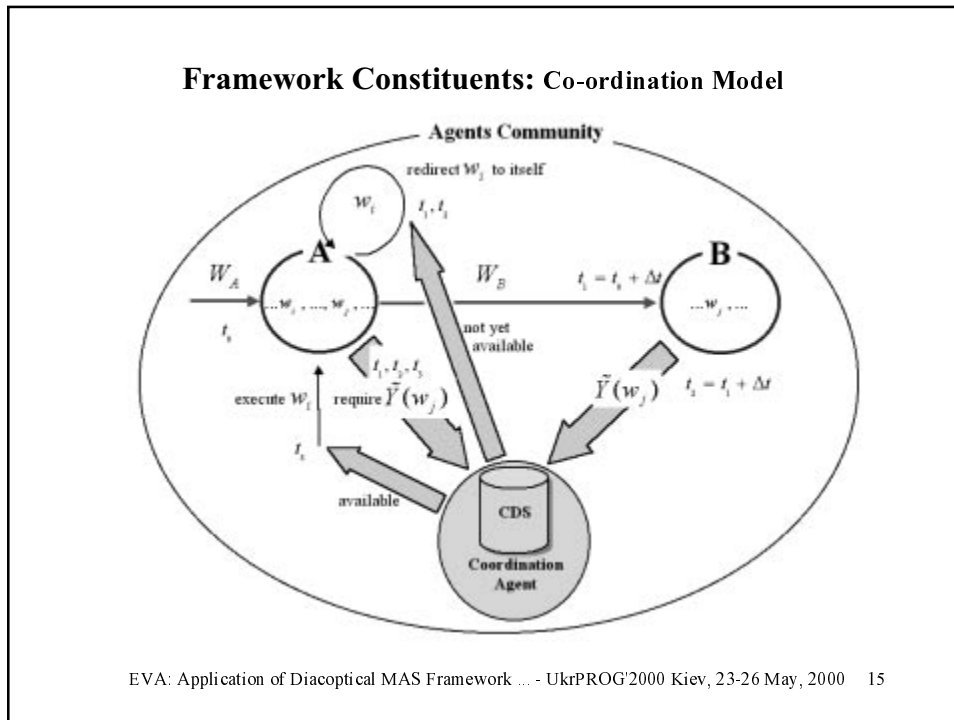
Component 1	$w_1$	$w_2$	$\dots$	$w_\sigma$
Component 2	$k_{11}$	$k_{12}$	$\dots$	$k_{1\sigma}$
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
Component $i$	$k_{i1}$	$k_{i2}$	$\dots$	$k_{i\sigma}$
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
Component $n$	$\dots$	$\dots$	$\dots$	$\dots$

$\Omega(t_n + \Delta t) = \begin{bmatrix} \Theta_1 \\ \dots \\ \Theta_i \\ \dots \\ \Theta_m \end{bmatrix} = \begin{bmatrix} k_{11}, k_{12}, \dots, k_{1\sigma} \\ \dots \\ k_{i1}, k_{i2}, \dots, k_{ij}, \dots, k_{i\sigma} \\ \dots \\ k_{m1}, k_{m2}, \dots, k_{m\sigma} \end{bmatrix}$

$t \dots t_{n+1} = t_n + \Delta t \quad t_n \quad \dots \quad t_0$

$\Omega(t_n + \Delta t)$  is maintained by Community Co-ordination Agent taking component states  $K_i$  as inputs:  $\Omega = \sum_{i=1}^m K_i$ .

EVA: Application of Diacoptical MAS Framework ... - UkrPROG'2000 Kiev, 23-26 May, 2000 14





### Case Study: IS Development Planning Process - Initial Phase

#### $t_0$ : Input influence:

$W_a = \{w_0 = 'Propose\_IS\_Development\_Plan', X_0, Y_0\}$

with the parameters and the result descriptions

$X_0 = \{budget = < figure >, duration = < figure >, Proposal\_Template = < file\_name >,$

$Proposal\_Descr = < file\_name >\}$ ;

$Y_0 = \{Possibility(budget, duration), Proposal(Proposal\_Template)\}$

comes to **PM's** sensory input. **PM** performs at least the following actions:

#### 1. Input influence verification. PM's Verificationation FSMs check out if

- the input influence,
- its parameters
- and result descriptions

comply with **PM's** role and current state constraints.

The further behaviour of PM (i.e. macromodel program) is chosen from two alternatives:

- the influence is accepted
- or the influence is rejected.

### Case Study: IS Development Planning Process - Initial Phase

#### $t_0$ : 2. Input work execution and agent state change: PM's macromodel $F_O^{PM}(W_a)$ (as defined by software project planning ontology) decomposes incoming work - i.e. generates the set of works corresponding to the project stages:

$W_{PM}^p = \{w_0 = 'Propose\_IS\_Development\_Plan', X_0, Y_0\}$  ,  $W_{PM}^d = \emptyset$  ,  $W_{PM}^r = \emptyset$  ,  
 $\dots, g = \{$   
 $w_1 = ('Assemble Project Proposal', X_1, Y_1),$   
 $w_2 = ('Choose best DB Schemata Plan Bid', X_2, Y_2),$   
 $w_3 = ('Choose best Software Model Plan Bid', X_3, Y_3),$   
 $w_4 = ('Choose best REQ Analyses Plan Bid', X_4, Y_4),$   
 $w_5 = ('Analyse requirements', X_5, Y_5),$   
 $w_6 = ('Design database schemata', X_6, Y_6),$   
 $w_7 = ('Design software model', X_7, Y_7),$   
 $w_8 = ('Program the software', X_8, Y_8),$   
 $w_8 =$   
 $\dots$   
 $w_{15} = ('Perform customers training ', X_{15}, Y_{15}).$   
 $w_{15} =$

Change of **PM's** state to the state by adding state constraints may be performed if:

- requested by the behaviour<sup>80</sup> rule provided by appropriate ontology
- or is encapsulated within the macromodel program.

### Case Study: IS Development Planning Process - Initial Phase

$t_0$  : 3. Component state matrix  $\mathbf{K}_{PM}(t_0)$  generation:

	$w_1$	$w_2$	$w_3$	$w_4$	$w_5$	$w_6$	$w_7$	$w_8$	...	$w_{15}$
DBA	0	0	0	0	1	1	1	0		0
PROG	0	0	0	0	0	1	1	1		0
PM	1	1	1	1	0	0	0	0	...	0
DOC	0	0	0	0	0	0	0	0		0
TST	0	0	0	0	0	0	0	0		0
LIA	0	0	0	0	1	1	1	0		1

$\mathbf{K}_{PM}(t_0)$  may be interpreted as follows:  
for the next modelling step  $t_0 + \Delta t$ :

- works  $w_1, w_2, w_3, w_4$  are planned for **self-performance**;
- works  $w_5, w_6, w_7$  are **redirected to more than one** functional component:  
 $w_5$  - to **DBA** and **LIA** agents,  
 $w_6$  - to **DBA, PROG** and **LIA** agents,  
 $w_7$  - to **DBA, PROG** and **LIA** agents.

**Reason:** the attempt to obtain optimal task performance by choosing the component with the best bid.

**Another reason:** PM doesn't really know well what are the duties of **DBA, PROG, LIA** and wants to gain some experience for the future.

$\Theta_{DBA}$	0	0	0	0	1	1	1	0		0
$\Theta_{PROG}$	0	0	0	0	0	1	1	0		0
$\Theta_{PM}$	1	1	1	1	0	0	0	0	...	0
$\Theta_{DOC}$	0	0	0	0	0	0	0	1		0
$\Theta_{TST}$	0	0	0	0	0	0	0	0		0
$\Theta_{LIA}$	0	0	0	0	1	1	1	0		1

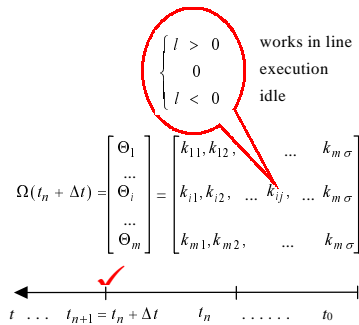
Functional System state matrix  $\Omega_a(t_0 + \Delta t)$  related to the process  $W_a$  will be the same as  $\mathbf{K}_{PM}(t_0)$

### The rules: (just to remind)

$\mathbf{K}_i$  is generated by each component:

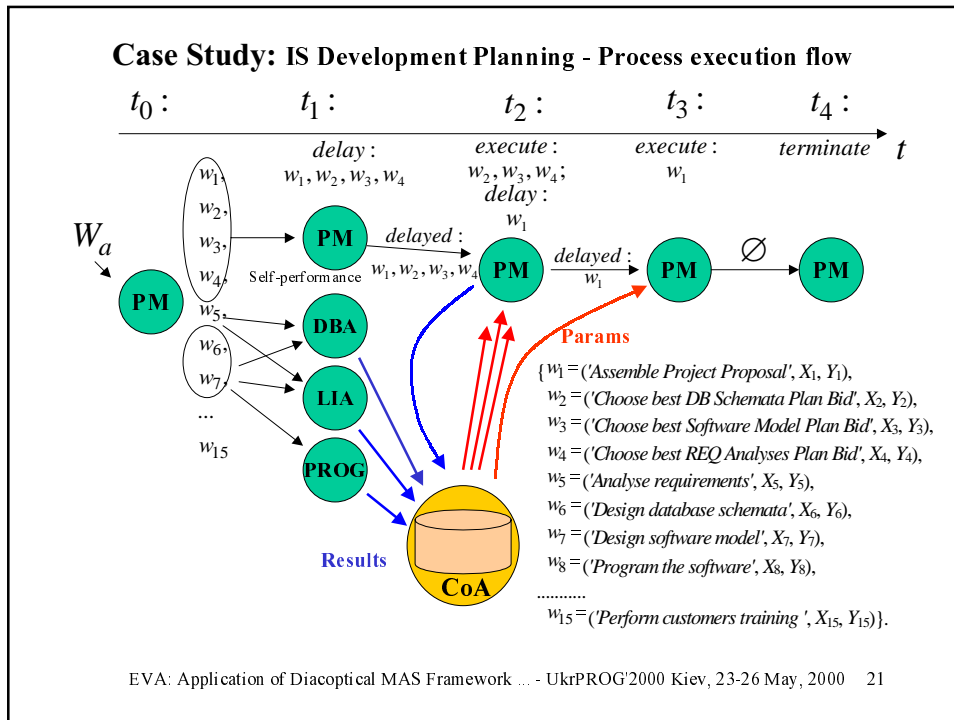
- $k_{lj}$  : **1** - component  $i$  allocates work  $w_j$  to component  $l$ ,
- 0** - otherwise
- $k_{ij}$  : **-1** - component  $i$  absorbs work  $w_j$  within interval  $[t_n, t_n + \Delta t]$ ,
- 1** - component  $i$  allocates work  $w_j$  to itself,
- 0** - component  $i$  is not capable to perform ... within  $[t_n, t_n + \Delta t]$

Component 1	$w_1$	$w_2$	...	$w_\sigma$
Component 2	$k_{11}$	$k_{12}$	...	$k_{1\sigma}$
...	...	...	...	...
Component $i$	$k_{i1}$	$k_{i2}$	...	$k_{i\sigma}$
...	...	...	...	...
Component $n$	$k_{n1}$	$k_{n2}$	...	$k_{n\sigma}$

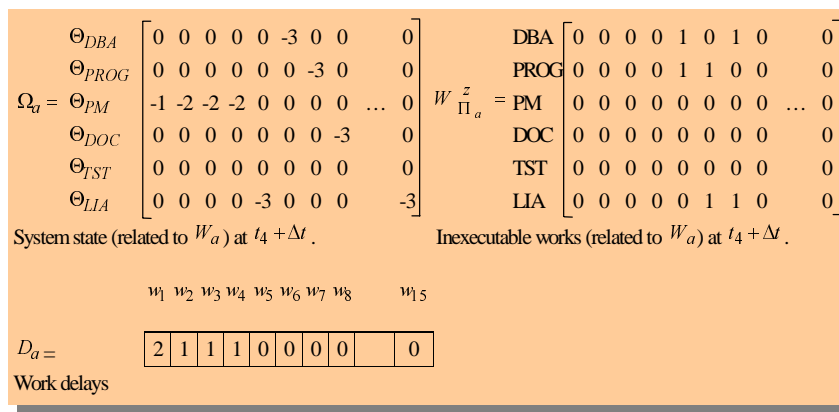


$\Omega(t_n + \Delta t)$  is maintained by Community Co-ordination Agent taking component states  $\mathbf{K}_i$

as inputs: 
$$\Omega = \sum_{i=1}^m \mathbf{K}_i$$



### Case Study: Final Results



### Case Study: What do we have here?...

#### 1. Parametric feedbacks for co-ordination and negotiation modelling:

Remind about the execution of  $w_2$ :

$t_2$ : PM

$w_2 = ('Choose\ best\ DB\ Schemata\ Plan\ Bid', X_2, Y_2),$

$X_2 = \{\tilde{Y}_6^{DBA}, \tilde{Y}_6^{PROG}, \tilde{Y}_6^{LIA}\},$

$Y_2 = \{y_2^1 = (Possibility(budget, duration)),$

$y_2^2 = (Plan\_File\_Name(Plan\_Template))\};$

$y_2^1 = \min(y_6^{1,DBA}, y_6^{1,PROG}, y_6^{1,LIA})$

$y_2^2$  - Plan corresponding to the best bid ( $y_2^1$ )

	0.5d	0.8d	1.0d	1.5d	1.8d
0.5b	p	p	p	p	p
0.8b	p	p	p	p	p
1.0b	p	p	p	p	p
1.5b	p	p	p	p	p
1.8b	p	p	p	p	p

Possibility: estimated values  $p \in [0,1]$

The resulting *Possibility* points to the best bid for *Database Schemata Design Plan* as well as provides the outgoing *Possibility* for this work depending from changing *budget* (0.5b-1.8b) and *duration* (0.5d-1.8d).

To say generally, parametrical feedbacks provided by the framework make modelling of co-ordination (like CNP or brokering) or negotiation (like auction) protocols by means of work sequences quite simple.

### Case Study: What do we have here?...

#### 2. Learning and behaviour evolution:

$w_i^z$	DBA	[0 0 0 0 1 0 1 0 0]	0
	PROG	[0 0 0 0 1 1 0 0 0]	0
	PM	[0 0 0 0 0 0 0 0 0 ...]	0
	DOC	[0 0 0 0 0 0 0 0 0]	0
	TST	[0 0 0 0 0 0 0 0 0]	0
	LIA	[0 0 0 0 0 1 1 0 0]	0

Inexecutable works ( $w_a$ ) at  $t_4 + \Delta t$ .

Works  $w_5, w_6, w_7$  had first been rejected, though they have finally been performed.

#### Inferences:

-  $w_{\Pi^a}^z$  may be used as a tip for more accurate work assignments:

- agent **PM** learns about the capabilities of other community members related to the process discussed and thus adapts its behaviour (updating a kind of incidence matrix).

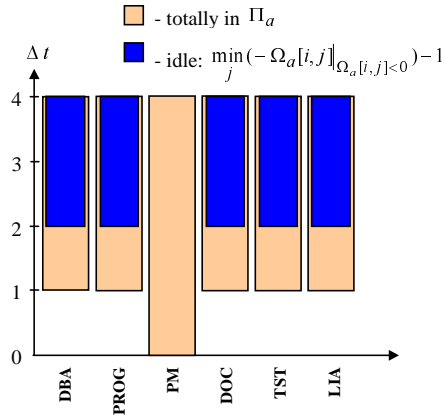
- Work  $w_i$  is really inexecutable by the community ( $w_i$  team) only if the column  $i$  of  $w_{\Pi^a}^z$  doesn't contain zero values

- The problem of dynamic change of  $\Pi^a$  team constitution is however opened

### Case Study: What do we have here?...

#### 3. Agents' load evaluation:

Evaluation of agents' activity related to process  $\Pi_a$  shows that average agents' load is 37.5%.



### Case Study: What do we have here?...

#### 4. Planning by simulation:

One of the accessory results which may be automatically obtained from Planning Process simulation is the draft of proposal preparation plan

Task/Work	$t_1$	$t_2$	$t_3$	$t_4$
Tasks Specification	PM			
REQ Analyses Planning		LIA		
DB Schemata Design Planning		DBA		
Software Model Design Planning		PROG		
Programming Planning		PROG		
.....		...		
Customers' Training Planning		LIA		
REQ Analyses Plan Optimisation			PM	
Software Modelling Plan Optimisation			PM	
DB Schemata Design Plan Optimisation			PM	
Project Plan Assembly				PM

### Case Study: What do we have here?...

**5. Communication acts as work sequences:**

a) communication act diagram

b) corresponding work sequence model

**t<sub>1</sub>**: DBA, PROG, LIA produce  $\tilde{Y}_6^{DBA}, \tilde{Y}_6^{PROG}, \tilde{Y}_6^{LIA}$  as requested by parameter descriptions (templates):

$$Y_6 = \{ y_6^1 = (\text{Possibility}(\text{budget}, \text{duration})), y_6^2 = (\text{Plan\_File\_Name}(\text{Plan\_Template})) \};$$

$$\tilde{Y}_6^{DBA} = \begin{bmatrix} 0.1 & 0.1 & 0.4 & 0.6 & 0.35 \\ 0.3 & 0.35 & 0.4 & 0.8 & 0.35 \\ 0.3 & 1.0 & 1.0 & 0.8 & 0.35 \\ 0.35 & 1.0 & 1.0 & 0.85 & 0.35 \\ 0.4 & 1.0 & 1.0 & 0.9 & 0.35 \end{bmatrix}$$

"DBSCH\_PLAN.XLS"

**t<sub>2</sub>**: PM

$w_2 = (\text{'Choose best DB Schemata Plan Bid'}, X_2, Y_2)$

$X_2 = \{ \tilde{Y}_6^{DBA}, \tilde{Y}_6^{PROG}, \tilde{Y}_6^{LIA} \}$

$Y_2 = \{ y_2^1 = (\text{Possibility}(\text{budget}, \text{duration})), y_2^2 = (\text{Plan\_File\_Name}(\text{Plan\_Template})) \};$

The works are executed in **INVERSED** order

EVA: Application of Diacoptical MAS Framework ... - UkrPROG'2000 Kiev, 23-26 May, 2000 27

### Case Study: What do we have here?...

**6. FIPA compliance:**

The work sequence model is notably similar to FIPA Contract Net Protocol.

**Conjectures:**

**1-st** - we may model FIPA ACL communicative acts by means of presented framework

**2-nd** - we may possibly use ACL as the transport language for inter - agent communication within the communities modelled in frame of the proposed approach.

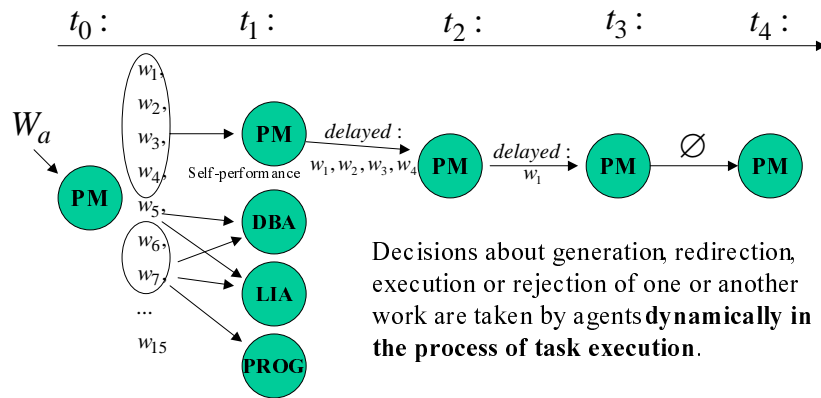
a) communication act diagram

b) corresponding work sequence model

EVA: Application of Diacoptical MAS Framework ... - UkrPROG'2000 Kiev, 23-26 May, 2000 28

### Case Study: What do we have here?...

#### 7. Absence of statically defined workflows, task plans:



The Workflow (Task Plan) is thus **created in step-by-step execution process**

### Case Study: What do we have here?...

#### 8. Manual preparation work and interaction with a Human "MASTER":

In practice we need to apply lots of inhouse work before the functional system is ready to operation:

- Macromodels,
- ontologies,
- state constraints,
- parameters', results' and role descriptions

are to be **prepared manually in advance** and stored into the agent/community knowledge base.

**Positive thing:** the major part of this work **is performed once** and the result may be **re-used by another agents/communities** (performing different tasks).

**Another aspect:** **not all of the atomic works may be executed automatically** by macromodel programs.

The agents are used as personal assistants providing substantial aid to the execution of the routine part of work. Execution of works in this Case Study assumes the participation of a human 'master' with his creative mind for at least the informal part of the plan preparation.

### Summary, Hypothesis, Future Work

**Summary:** Presented case study has shown that the framework may be applied to modelling business processes as tasks presented by sets of atomic works.

Case study analysis has provided some results valuable both from application and theoretical generalisation points of view.

The framework provides good opportunities for further fine-tuning of community members behaviour (ontologies, state constraints and macromodel programs) based on the utilisation of the results obtained while modelling of one or another task execution process.

**Hypothesis:** The framework also seem to provide visible possibilities to model known communication protocols, communicative acts (contract net, negotiation, ...) and another types of business processes by means of appropriately configured work sequences.

**Future work:**

- **More case studies** in outlined Application Domains - I\*3, Enterprise Modelling, Monitoring,
- Development of the Full-Scale **Architectural Framework**
- **Implementation** of Multi-Agent Systems based on the frameworks