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.zaporizhzhe.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**...»

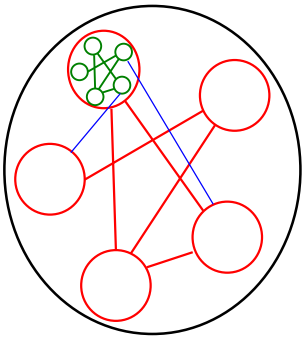
...Technological Roadmap of the ESPRIT AgentLink I2A SIG

- 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)

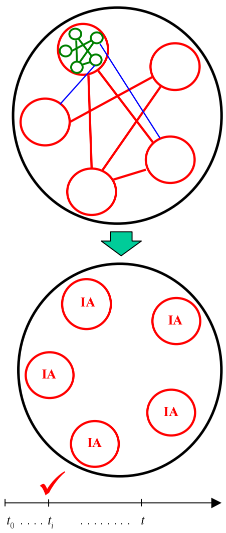


- based on the usage of static structured collections of rigid relationships among the participating functional nodes (eg. Petri Nets, Task execution plans, ...).

- 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



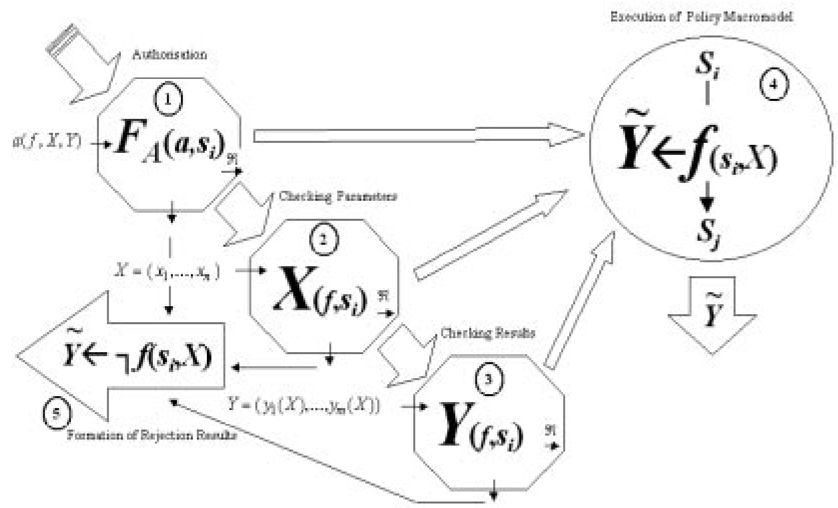
Diakoptical framework for modelling of the processes of information interchange among the distributed components (intellectual agents) of a dynamic information system:

- **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**

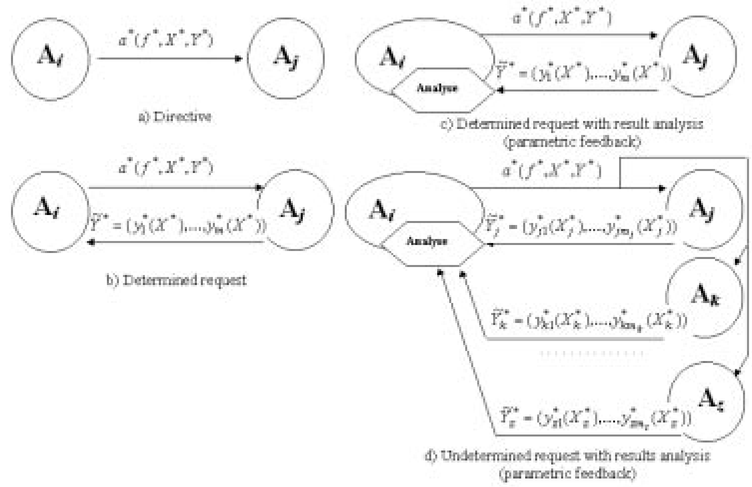
Framework Constituents:

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

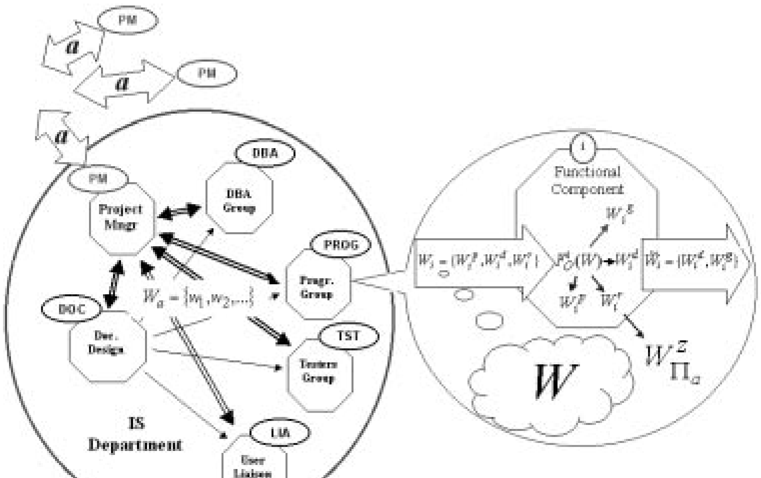
Framework Constituents: Generic Agent's Model



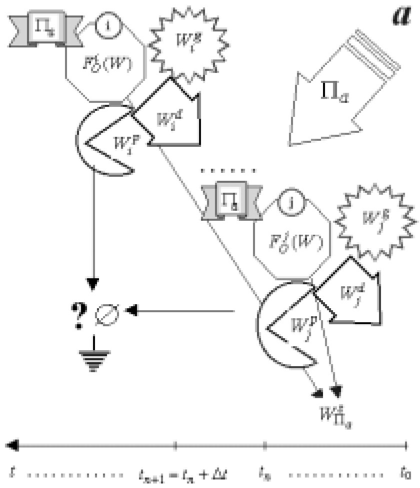
Framework Constituents: Communication Model



Framework Constituents: Functional System/Component Model



Framework Constituents: Process Model



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

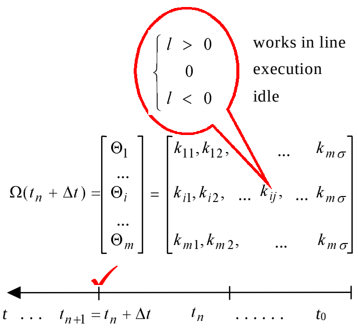
The rules for Component State and System State Matrixes generation

K$_i$ **is generated by each component:**

k_{lj}, $l \neq i$: **1** - component i allocates work w_j to component l,
0 - otherwise

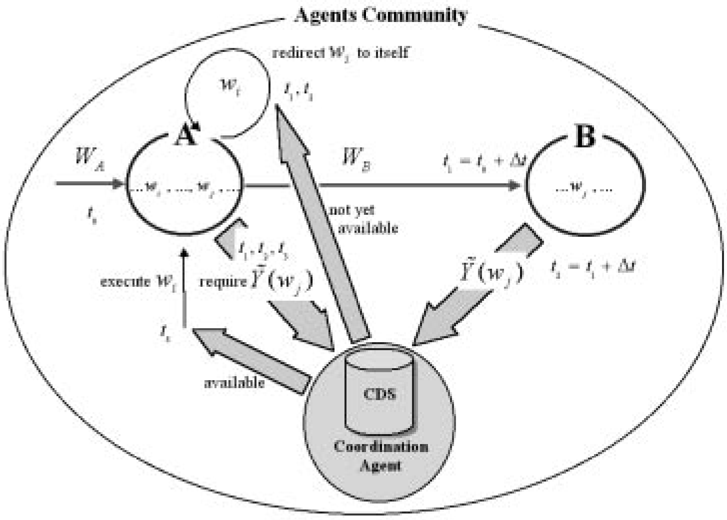
k_{ij}: **-1** - component absorbs work w_j within interval $]t_n, t_n + \Delta t]$,
1 - component i allocates work w_j to itself,
0 - component i is not capable to perform w_j within $]t_n, t_n + \Delta t]$

$$\mathbf{K}_i = \begin{array}{c} \\ \text{Component 1} \\ \text{Component 2} \\ \dots \\ \text{Component } i \\ \dots \\ \text{Component n} \end{array} \begin{bmatrix} w_1 & w_2 & \dots & w_\sigma \\ k_{11} & k_{12} & \dots & k_{1\sigma} \\ k_{21} & k_{22} & \dots & k_{2\sigma} \\ & & & \\ k_{i1} & k_{i2} & \dots & k_{i\sigma} \\ & & & \\ k_{n1} & k_{n2} & \dots & k_{n\sigma} \end{bmatrix}$$

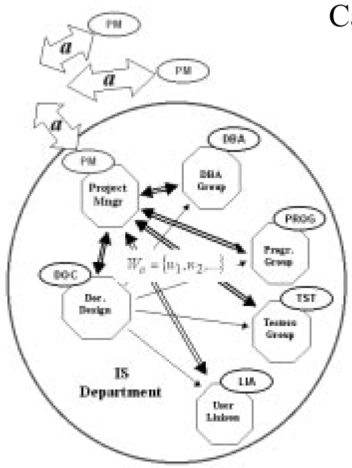


$\Omega(t_n + \Delta t)$ is maintained by Community Co-ordination Agent taking component states $\mathbf{K}_i$ as inputs: $\Omega = \sum_{i=1}^{m} \mathbf{K}_i$.

Framework Constituents: Co-ordination Model



Case Study: IS Development Planning



Case Assumptions:

- IS Department is one of many candidates to be selected for the project.
- IS Department as well as the other parties are considered to be the member agents of the higher level community.
- Agents community of IS Department comprises Project Manager (**PM** agent), DBA Group (**DBA** agent), Programmers Group (**PROG** agent), Documentation Design Group (**DOC** agent), Testers Group (**TST** agent), User/Customer Liaison Group (**LIA** agent) as functional components.
- In case the necessity appears to go in more details with one of the functional components listed, this component may be modelled in the same way: as a functional system - agents community.

Case Study: IS Development Planning Process - Initial Phase

t_0 :

Input influence:

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

with the parameters and the result descriptions

$X_0 = \{\textbf{budget} =< figure >, \textbf{duration} =< figure >, \textbf{Proposal\_Template} =< file\_name >,$
$\textbf{Proposal\_Descr} =< file\_name >\}$;

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

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:

$W_{PM}^p = \{w_0 = \text{'}Propose\_IS\_Development\_Plan\text{'}, X_0, Y_0\}$, $W_{PM}^d = \varnothing$, $W_{PM}^r = \varnothing$,

$W^g = \{$
$w_1 =$ ('*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),
....
$w_{15} =$ ('*Perform customers training* ', X_{15}, Y_{15})$\}$.

Change of **PM**'s state $S0$ to the state $S1$ by adding state constraints may be performed if:
- requested by the behaviour 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 $K_{PM}(t_0)$ generation**:

$$K_{PM} = \begin{array}{c} \\ DBA \\ PROG \\ PM \\ DOC \\ TST \\ LIA \end{array} \begin{array}{c} w_1\ w_2\ w_3\ w_4\ w_5\ w_6\ w_7\ w_8\ \dots\ w_{15} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & & 1 \end{bmatrix} \end{array}$$

$$\Omega_a = \begin{array}{c} \Theta_{DBA} \\ \Theta_{PROG} \\ \Theta_{PM} \\ \Theta_{DOC} \\ \Theta_{TST} \\ \Theta_{LIA} \end{array} \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & & 1 \end{bmatrix}$$

$K_{PM}(t_0)$ 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.

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

EVA: Application of Diacoptical MAS Framework ... - UkrPROG'2000 Kiev, 23-26 May, 2000 19

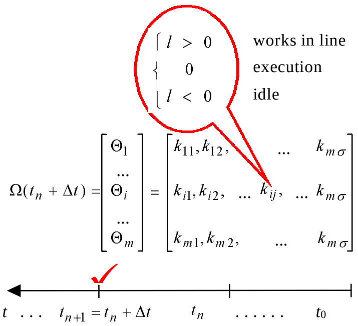
The rules: (just to remind)

K_i is generated by each component:

k_{lj}, $l \neq i$: **1** - component i allocates work w_j to component l,
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 w_j within $]t_n, t_n + \Delta t]$

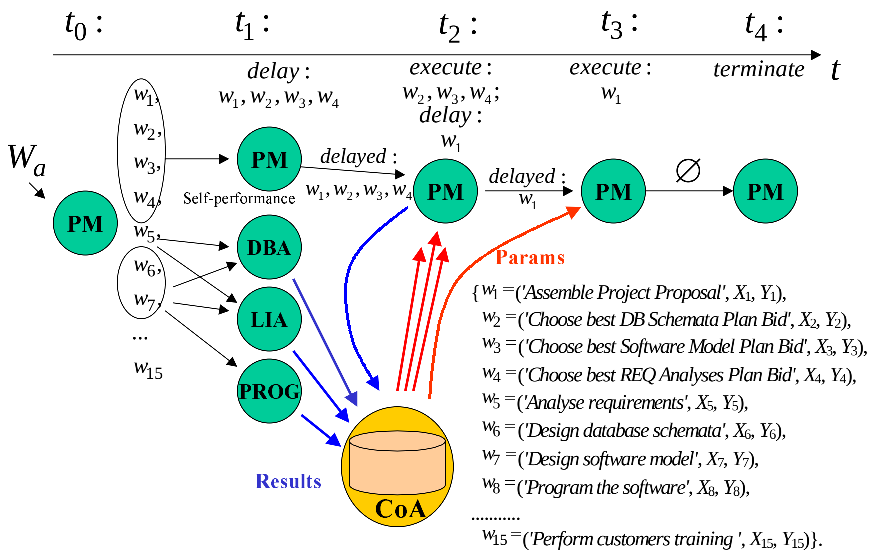
$$K_i = \begin{array}{c} \\ \text{Component 1} \\ \text{Component 2} \\ \dots \\ \text{Component } i \\ \dots \\ \text{Component n} \end{array} \begin{bmatrix} w_1 & w_2 & \dots & w_\sigma \\ k_{11} & k_{12} & \dots & k_{1\sigma} \\ k_{21} & k_{22} & \dots & k_{2\sigma} \\ & & & \\ k_{i1} & k_{i2} & \dots & k_{i\sigma} \\ & & & \\ k_{n1} & k_{n2} & \dots & k_{n\sigma} \end{bmatrix}$$



$\Omega(t_n + \Delta t)$ 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 20

Case Study: IS Development Planning - Process execution flow



Case Study: Final Results

$$\Omega_a = \begin{matrix}\Theta_{DBA}\\ \Theta_{PROG}\\ \Theta_{PM}\\ \Theta_{DOC}\\ \Theta_{TST}\\ \Theta_{LIA}\end{matrix}\begin{bmatrix} 0&0&0&0&0&-3&0&0& &0\\ 0&0&0&0&0&0&-3&0& &0\\ -1&-2&-2&-2&0&0&0&0&\dots&0\\ 0&0&0&0&0&0&0&-3& &0\\ 0&0&0&0&0&0&0&0& &0\\ 0&0&0&0&-3&0&0&0& &-3\end{bmatrix}$$

System state (related to W_a) at $t_4 + \Delta t$.

$$W^{z}_{\Pi_a} = \begin{matrix}DBA\\ PROG\\ PM\\ DOC\\ TST\\ LIA\end{matrix}\begin{bmatrix} 0&0&0&0&1&0&1&0& &0\\ 0&0&0&0&1&1&0&0& &0\\ 0&0&0&0&0&0&0&0&\dots&0\\ 0&0&0&0&0&0&0&0& &0\\ 0&0&0&0&0&0&0&0& &0\\ 0&0&0&0&0&1&1&0& &0\end{bmatrix}$$

Inexecutable works (related to W_a) at $t_4 + \Delta t$.

	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8		w_{15}
$D_a =$	2	1	1	1	0	0	0	0		0

Work delays

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),

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

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

y_2^2=(*Plan_File_Name*(***Plan_Template***))};

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

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

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

$$W_{\Pi_a}^z = \begin{array}{l} \text{DBA} \\ \text{PROG} \\ \text{PM} \\ \text{DOC} \\ \text{TST} \\ \text{LIA} \end{array} \begin{bmatrix} 0&0&0&0&1&0&1&0& &0 \\ 0&0&0&0&1&1&0&0& &0 \\ 0&0&0&0&0&0&0&0&\dots&0 \\ 0&0&0&0&0&0&0&0& &0 \\ 0&0&0&0&0&0&0&0& &0 \\ 0&0&0&0&0&1&1&0& &0 \end{bmatrix}$$

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.

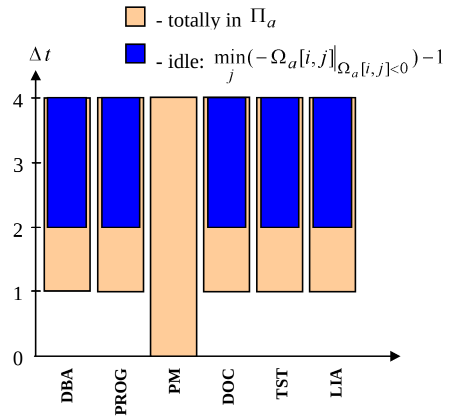
- Work w_i is really inexecutable by the community (Π_a team) only if the column *i* of $W_{\Pi_a}^z$ doesn't contain zero values

- The problem of dynamic change of Π_a team constitution is however opened

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

3. Agents' load evaluation:

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



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

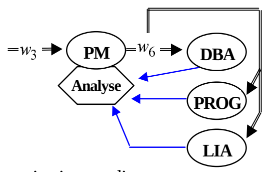
4. Planning by simulation:

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

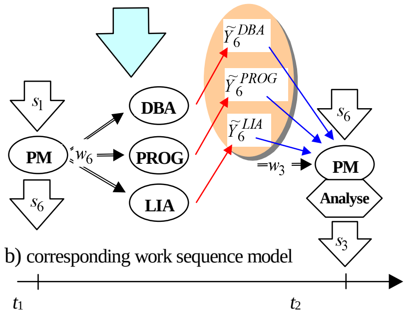
Task/Work	t_1	t_2	t_3	t_4
Tasks Specification	**PM**			
REQ Analyses Planning		**LIA**		
DB Schemata Design Planning		**DBA**		
Software Model Design Planning		**PROG**		
Programming Planning		**PROG**		
....		...		
Customers' Training Planning		**LIA**		
REQ Analyses Plan Optimisation			**PM**	
Software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

t1: **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=(*Possibility*(***budget***, ***duration***)),
y_6^2=(*Plan_File_Name*(***Plan_Template***)) }:

$$\tilde{Y}_6^{DBA} = \{ \begin{bmatrix} 0.1 & 0.1 & 0.4 & 0.6 & 0.35 \\ 0.3 & 0.35 & 0.4 & 0.8 & 0.35 \\ 0.3 & 1.0 & 1.0 & 0.8 & 0.35 \\ 0.35 & 1.0 & 1.0 & 0.85 & 0.35 \\ 0.4 & 1.0 & 1.0 & 0.9 & 0.35 \end{bmatrix}, "DBSCH\_PLAN.XLS" \}$$

t2: **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***))};

The works are executed in **INVERSED order**

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

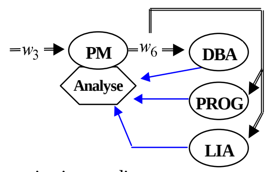
6. FIPA compliance:

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

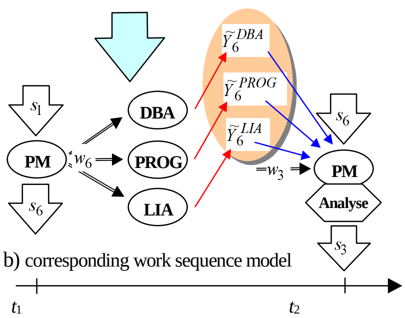
Conjectures:

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.



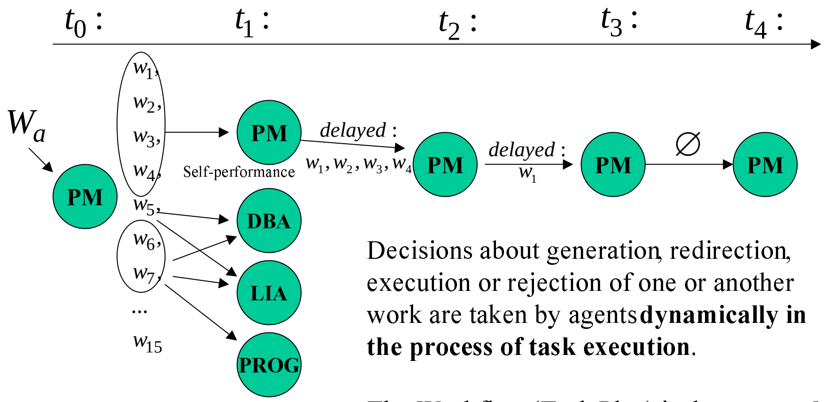
a) communication act diagram



b) corresponding work sequence model

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

7. Absence of statically defined workflows, task plans:



Decisions about generation, redirection, execution or rejection of one or another work are taken by agents **dynamically in the process of task execution**.

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