Cooperative Cognition and Its Implementation under Web Environment

Wanchun DOU[1,2,3*], Guihai CHEN[1,2], S.C.CHEUNG[3], and Shijie CAI[1,2]
1.*State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing,P.R.China,*
2.*Department of Computer Science and Technology, Nanjing University, Nanjing, P.R.China,*
3.*Department of Computer Science, Hong Kong University of Science and Technology, Hong Kong*
douwc@nju.edu.cn, gchen@nju.edu.cn, scc@cs.ust.hk, sjcai@nju.edu.cn

Abstract

Problem cognition is often the first step to problem solving. Complex problem solving is often promoted by cooperative cognition. Our research work presented in this paper is initialed by a motivated example. Then meta-cognition and cooperative cognition are explored by defining their Cognition Logic Context and Cognition Resource Context. Cognition evolution is discussed from the transformation of problem state space. Moreover, the knowledge flow principle engaged in cooperative cognition is explored from the perspective of learning and cognition evolution. To promote cooperative cognition under Web environment, a context application paradigm is proposed based on P2P scenarios. The conclusions and our future works are presented at last.

1. Introduction

Now, knowledge management (KM) is a topic of considerable interest [1, 2]. The majority of extant KM research often treats knowledge as a static object; whereas very little attentions are paid to the dynamic feature of knowledge processing and knowledge management, which is often called as knowledge flow [3]. Currently, the concept of knowledge flow is often employed to model and steer the knowledge interaction, knowledge innovation, knowledge transfer, and knowledge sharing among participators in teamwork [4]. For instance, H.Zhuge [4] defined the knowledge flow as a process of knowledge passing between people or knowledge processing mechanism, and pointed out that direction, content and carrier are three crucial attributes of knowledge flow. Moreover, M.E.Nissen and R.E.Levitt [3] summarized that notional Knowledge-Flow trajectories can be delineated by a curvilinear vector sequence S-E-C-I corresponding to the processes of socialization, externalization, combination, and internalization, from the aspect of cognition. The methodology employed in knowledge flow research can be popularized in distributed problem-solving process, in which problem cognition is often the first step to problem solving and complex problem solving is often promoted by cooperative cognition [5]. Upon a certain problem, cognition process aims at externalizing and formalizing the internal logic contained inside the problem, which is often a recursive process in a progressive way [1, 6, 7].

Traditionally, studies in cognitive psychology have focused on individual cognition [8]. As a tendency, Internet technologies have enabled problem solving to be globally distributed for upgrading the competitiveness and promoting team's innovation. For distributed and complex problem solving is often a cognition- and learning-compliant knowledge interaction among individuals or teams, cognition and learning are two basic aspects that are interdependent each other during problem solving. The methods to promote cognition and learning provide foundations for exploring the principle of knowledge flow from different behavioral perspectives.

Upon a certain problem, its solving process is often executed under a certain Problem Solving Environment (PSE) that is a complete, integrated environment for composing, compiling, and running applications in a specific area [9, 10]. Furthermore, to facilitate Web-based knowledge propagation and sharing during distributed cooperative cognition, Knowledge Grid and semantic Web attract more and more attentions upon complex problem solving [11, 12], in which different PSEs provide backend computational resources of Knowledge Grid. In this paper, we would concentrate on exploring cognition contexts and related knowledge flow principle engaged in cooperative cognition from ***software engineering*** perspective.

2. A motivated example

Here, a cognitive pattern derived from New Product Development (NPD) would be analyzed to facilitate our further discussion. Roughly, NPD is a knowledge-driven action that concentrates on transformation from tacit knowledge to explicit knowledge, through which the expected product is materialized, step by step, from an obscure and vague state to an evident and plain state [6, 7]. The tacit knowledge includes not only the designer's basic perception or consciousness of the expected product, but also some knowledge stored in enterprise's repository. To promote NPD, the conceptualizing and formalizing processes with the goal to reify tacit knowledge from enterprise repository could be identified by following 4 stages: (1) Conceptualize the expected product and existed product; (2) Frame the case structure and gather related decision cases; (3) Learn from feedback; (4) Develop solutions through iteration of ideas exchange.

Here, the expected product could be specified in form of *ExpectedProduct* = $\{fa_1, fa_2, \ldots, fa_n\}$ through qualitative analysis around its fundamental features, where fa_i stands for a basic functional attributes related to the expected product. Once the expected product could be described in this formalization, NPD has achieved its first step with cognition evolution. At this stage, fa_i is still a candidate object to be cognized, which depending on further quantitative analysis in details. Let *ReferredAnalogue* = {*Product-1*, *Product-2*, … , *Product-n*} be a set of existed products related to the expected product, i.e. there are n feasible design patterns that could be referred to in later product development. These accumulated effective design proposals could be treated as auxiliary knowledge resource to enlighten concrete product design.

Moreover, let *SetofInherited* be a subset derived from *ExpectedProduct*, which stands for a set of functional attributes that could be mapped onto *ReferredAnalogue*. Please note that the mapping image of element fa_i is indicated by fa_i', in this paper. In practice, there could be more than one image in conformance with fa_i. Let *SetofInnovated* be a subset derived from *ExpectedProduct*, which stands for the set of original functional attributes that has no images in existed products. The number of elements contained in *SetofInnovated* measures the originality of NPD. The smaller it is, the less originality NPD owns. If *SetofInnovated*=*ExpectedProduct*, the NPD could be treated as an absolute originality. If *SetofInherited* = *ExpectedProduct*, the NPD could be treated as an absolute inherited design. Here, the absolute originality could be treated as an unstructured cognition process; while the absolute inherited process could be treated as a structured cognition process. If ($SetofInherited \neq \Phi$) $\wedge$ ($SetofInherited \subset ExpectedProduct$), the NPD pattern could be treated as a semi-structured cognition process. Obviously, there are some relations among those two sets:

i)$ExpectedProduct = SetofInherited \cup SetofInnovated$

ii)$SetofInherited \cap SetofInnovated = \Phi$.

Figure 1 illustrates the relations among those sets. Please note that the association between *SetofInherited* and *ReferredAnalogue* is an aggregation, while the relation between *SetofInherited* and *ExpecteProduct*, and the relation between *SetofInnovated* and *ExpecteProduct* is a composition. To facilitate our later discussion, a basic concept related to NPD would be defined as follows.

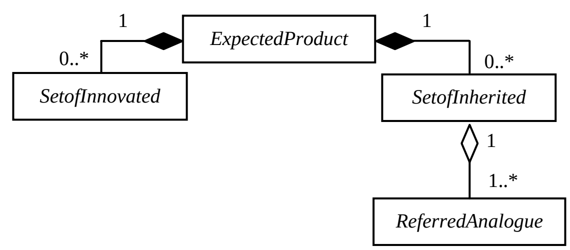

Figure 1. Relations among *ExpectedProduct*, *SetofInherited*, *SetofInnovated*, and *ReferredAnalogue* in form of UML

Definition1. Let $fa_i \in SetofInherited$, the collection of fa_i' is viewed as an auxiliary catalyzer for reifying tacit knowledge, from *ReferredAnalogue*, to explicit knowledge, which is indicated by fa_i/*ReferredAnalogue*.

In this paper, the fa_i/*ReferredAnalogue* stands for the Resource Context (RC) around fa_i. Besides RC, there also stands another kind of context upon fa_i, i.e., Logic Context (LC). For example, the parameters driven by the assembly relation are typical issues engaged in LC around fa_i and fa_j, if an assembly relation between fa_i and fa_j arises during NPD. Here, the contents of RC are derived from static enterprise repository, and the contents of LC are determined dynamically during design process. The RC plays the role to pilot knowledge acquisition from static knowledge resource, which could be treated as a static knowledge configuration upon NPD. The LC derived from assembly relation points out the future knowledge interaction relation among collaborators when required. Both RC and LC are indispensable to piloting knowledge flow execution from direction, content and carrier during NPD [4]. In view of the dynamics of NPD, some unexpected context information could be produced during NPD for piloting dynamic knowledge interactions.

3. Scenarios on cognition for promoting problem solving

Similarly, let *GlobalProblem* = $\{LP_1, LP_2, \ldots, LP_i, \ldots, LP_n\}$, where LP_i stands for a local problem derived from global problem decomposition, based on problem characteristics. Let *InnovatedProblem* be a subset derived from *GlobalProblem*, in which each LP_i has no referred analogue to promote its solving, let *InheritedProblem* be a subset derived from *GlobalProblem,* in which each LP_i could find some referred proposals from existed repository to promote its solving. According to the hypothesis, some basic concepts related to cognition are defined as follows.

Definition2. LP_i's cognition logic context (CLC) consists of local problems that have direct input, output or other directive relations with LP_i in global problem logic or in problem solving process; LP_i's cognition resource context (CRC) consists of all the proposals derived from existed repository that could be referred to during its solving process.

Here, if the CLC and CRC could be predefined in advance before problem solving, we call them static CLC and static CRC. With cognition's evolution, some new content would be recruited to enrich the predefined CLC and static CRC.

Definition3. Let $CLC\text{-}LP_i$ be a set that consists of local problems engaged in LP_i's CLC specification. Here, $CLC\text{-}LP_i \subseteq GlobalProblem$, and $LP_i \notin CLC\text{-}LP_i$. If $CLC\text{-}LP_i = \Phi$, we believe that the solving process around LP_i is a meta-cognition. If $CLC\text{-}LP_i \neq \Phi$, we believe that the solving process around LP_i would be evolved based on a cooperative cognition fashion.

Definition4. Upon two local problems LP_i and LP_j, if $CLC\text{-}LP_i \cap CLC\text{-}LP_j \neq \Phi$, we define the overlapped content of $CLC\text{-}LP_i$ and $CLC\text{-}LP_j$, i.e., $CLC\text{-}LP_i \cap CLC\text{-}LP_j$, is the shared cognition objects to steer their cooperative cognition.

Those shared cognition objects defined in Def.4 are basic problems to be settled, which just underlies the cooperatrive cognition. The cognition consistency around those objects lays a foundation for the consistency of further problem solving. Dynamic knowledge interactions often occur around those problems in form of knowledge flow. Typically, we would explore evolution pattern around meta-cognition from the transformation of problem state space.

Definition5. LP_i-around meta-cognition consists of spectrums of evolutionary cognition behaviors or evolutionary stages, which could be formalized by a triple form of *MetaCognition* = <*PreviousState*, $\Rightarrow$, *SucceedState* >, where, *MetaCognition* stands for a round of cognition, *PreviousState* stands for a set of problem elements to be settled before this round of cognition; '$\Rightarrow$' stands for introspective-based learning and reflecting process around *PreviousState*, which arms at externalizing the internal logic contained inside the problems; *SucceedState* stands for the set of left problem elements after this round of cognition.

Here, some conclusions would be drawn from Def.5 as follows:

1) If *PreviousState*=*SucceedState*, we believe that this round of cognition achieves no evolutional progress and the cognition process stands in a logjam state.

2) If $SucceedState \subset PreviousState$, we believe that this round of cognition has achieved some evolutional progress. $PreviousState - SucceedState$ stands for the problem elements which have been settled.

3) Let lp_i be a problem element, if $(lp_i \in SucceedState) \wedge (lp_i \notin PreviousState)$, we believe that this round of cognition get some progress on an unexpected way.

4) If $SucceedState = \Phi$, we believe that this round of cognition has be finished successfully and achieve all the expected result, i.e. all the problem elements contained in *PreviousState* have been solved absolutely.

During the cognition process, a peer should try his/her best to capture and exploit required knowledge resources, from repository or other peers, to enhance its cognition efficiency or cognition ability. It is a typical learning process, through which it is helpful for inspiring each other among collaborators, especially for alternate evolution. Besides, cognition process around complex problem solving could consist of some rounds of cognition evolutions in form of Def.5.

4. Knowledge flow principle engaged in cooperative cognition

Let $Agent_i$ and $Agent_j$ be two collaborators engaged in cooperative cognition, <*CognitionPrecondition-i*: *PreviousState-i*, $\Rightarrow$, *SucceedState-i*> is a round of cognition performed by $Agent_i$ based on set of preconditions of *CognitionPrecondition-i*. <*CognitionPrecondition-j: PreviousState-j*, $\Rightarrow$, *SucceedState-j*> is a round of cognition performed by $Agent_j$ based on set of preconditions of

CognitionPrecondition-j. Let lp_i' be the cognitive result of lp_i, if $lp_i \in$(PreviousState-*i*-SucceedState-*i*)$\cap$(lp_i'$\in$CognitionPrecondition-*j*), we believe that there stands a dynamic learning relation between $Agent_i$ and $Agent_j$, in which $Agent_j$ is a learner during their cognitive cooperation. Similarly, cooperative cognition around complex problem solving could be defined by Def.5'.

Definition5'. Cooperative cognition around complex problem solving process consists of spectrums of evolutionary *MetaCognition*, which could be formalized by a triple form of *CooperativeCognition* = <*PreviousState*, $\Rightarrow$, *SucceedState* >.

where, *PreviousState* stands for the set of problems to be settled before cooperative cognition,'$\Rightarrow$' stands for cooperative cognition process around *PreviousState*, *SucceedState* stands for the set of left problems after a round of cooperative cognition. Similar conclusions could be achieved as drawn in Def.5, which is overleaped here.

The learning relation between $Agent_i$ and $Agent_j$ presented here is just a kind of learning-based knowledge flow pattern with cognition intentions. Accordingly, a general learning-based knowledge flow pattern is defined as follows.

Definition 6. Let f_i be a cognitive function against cognized object of LP_i; θ_i be a set that consists of referred objects for learning, which is often specified by LP_i's CLC, LP_i's CRC, and LP_i's definitions (LPD_i). Here, θ_i plays as cognition ground. Accordingly, a general learning-based cognition pattern performed by $Agent_i$ could be formalized as $f_i(\theta_i: LP_i \rightarrow \Theta_i)$, where Θ_i stands for the cognition output after a round of cognition.

For highlighting different cognized objects, θ_i could be divided into θ_i=CLC_i, θ_i=CRC_i, and θ_i = LPD_i, in which CLC_i and CRC_j stands for LP_i's CLC and LP_i's CRC, respectively. Besides, $f_i(\theta_i)$ stands for θ_i- and learning-based cognition process. For example, f_i(CLC) stands for CLC- and learning-based cognition function, f_i(CRC) stands for CRC- and learning-based cognition function, and $f_i(LPD_i)$ stands for problem specification-based cognition function. The learing-based cooperation cognition could be formalized as follows if there stands a learning relation between $Agent_i$ and $Agent_j$:

$$\begin{cases} LP_i \xrightarrow{f_i(CLC_i)\wedge f_i(CRC_i)\wedge f_i(LPD_i)} \Theta_i \\ LP_j \xrightarrow{f_j(CLC_j)\wedge f_j(CRC_j)\wedge f_j(LPD_{i\,j})\wedge f_j(\Theta_i)} \Theta_j \end{cases}$$

where $f_j(\Theta_i)$ stands for Θ_i- and learning-based cognition process performed by $Agent_j$.

Accordingly, a general cognition process would be defined as follows based on Def.6.

Definition7. A general cognition process could be formalized by (*GlobalProblem*, θ, F, Θ; R_1, R_2, R_3). Where *GlobalProblem* has the same meaning as it presented in section 3; θ stands for the set of cognition background of θ_i; F is the set of cognition function of f_i, Θ is the set of Θ_i; R_1 stands for the relation between *GlobalProblem* and θ; R_2 stands for the relation between F and θ; while R_3 stand for the relation between *GlobalProblem* and Θ.

The transformation from obscurity or vagueness to evident and plain representation characterizes the cognition process toward problem solving. Learning process is an efficient way to degrade the obscurity. Accordingly, the knowledge flow engaged in cooperative cognition could be formalized as follows based on Def.7:

$$(GlobalProblem\,R_1\,\theta,\ F R_2\,\theta) \rightarrow (GlobalProblem\,R_3\,\Theta)$$

Once the global problem's internal logic could be formalized externally based on cognition evolutions, concrete tasks could be assigned and carried out among participants or teams based on their knowledge background [13].

5. A context-aware application paradigm based on P2P scenarios

In this section, we would focus on discussing cooperative cognition's implementation under Web environment from ***software engineering*** perspective. With recent advances in pervasive devices and communication technologies, networked services extend support from Web browsers on personal computers to handheld devices and sensor networks. Knowledge interaction piloted by CLC and resource access piloted by CRC could also be treated as CLC-aware and CRC-aware context application in form of knowledge service or resource service. Accordingly, we believe that an ideal PSE for promoting cooperative cognition should have two basic abilities related to collaboration: (1) ***how*** to find and locate required resource efficiently, and (2) ***how*** to manipulate the resource efficiently after it is located. Therefore, knowledge service under an open PSE should facilitate cognition evolution from two aspects of efficient ***knowledge location service*** and ***knowledge manipulation service***. Here, the collaborator location is covered by ***knowledge location service***.

Here, we would find that the features presented here share some attributes with P2P system, and the cooperative cognition under Web environment could be treated as a P2P application. According to P2P scenarios, a peer could be cut out for depicting local problem solving process; while P2P approach could be exploited to promote cooperative cognition under Web environment. More specifically, the introspection of meta-cognition is similar to the activity of a peer; and the socialization and combination of cooperative cognition is similar to service-based overlay architecture, which underlie the distributed, decentralized, and self-managing cooperative cognition under Web environment [14]. A peer could publish his/her information or knowledge like a server to other peers to meet some requirements; whereas, it would also require and capture some information or knowledge from other peers like a client.

In view of setting up an efficient context index mechanism is important for cooperative cognition [15], we would focus on exploring a ***context-aware*** application related to ***knowledge location service***, in this section, based on P2P scenarios. Here, four kinds of entities should be discriminated from cooperative cognition, around which a PSE should provide necessary services when required. The four kinds of entities are referred object (RO), local problem (LP), PEER, and Web-based cooperative cognition context mechanism (WBCCCM) as demonstrated in Figure 2. The attributes and methods contained in those classes are depicted in Table 1.

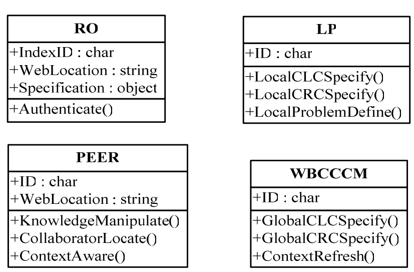

Figure 2. The classes engaged in cooperative cognition

Table 1. Specifications of the attribute/methods that are defined in Figure 2

No.	Item	Specifications
1	RO.IndexID	Attribute for indicating the ID of a referred object.
2	RO.WebLocation	Attribute for indicating the location of a referred object under Web environment, such as its IP address of a server name.
3	RO.Specification	Attribute for depicting a referred object from aspects of functional attributes, knowledge type, knowledge domain, and ontology description, etc.
4	RO.Authenticate()	Function for granting a certificate to a valid consumer for resource access, if needed.
5	LP.ID	Attribute for indicating the ID of LP.
6	LP.LocalCLCSpecify()	Function for depicting LP's CLC.
7	LP.LocalCRCSpecify()	Function for depicting LP's CRC.
8	LP. LocalProblemDefine()	Function for depicting LP's definitions and its specification.
9	PEER.ID	Attribute for indicating the ID of a peer.
10	PEER.Weblocation	Attribute for indicating the location of a peer under Web environment, such as its IP address or the workgroup a peer belongs to.
11	PEER. KnowledgeManipulate()	Function for manipulating the knowledge resource with certain tools or equipments to promoting its problem solving.
12	PEER. ResourceLocate()	Function for locating, communicating, interacting with a peer's collaborator.
13	PEER.ContextAware()	Function for apperceiving a peer's PSE based on specifications of CLC and CRC, some AI toolsets, and WBCCCM.ContextRefresh().
14	WBCCCM.ID	Attribute for indicating the ID of a WBCCCM.
15	WBCCCM. GlobalCLCSpecify()	Function for specifying CLC around global problem based on each LP's CLC definition.
16	WBCCCM. GlobalCRCSpecify()	Function for specifying CRC around global problem based on each LP's CRC definition.
17	WBCCCM. ContextRefresh()	Function for updating global CLC and CRC if needed under runtime environment.

In conformance with those specifications as listed in Table 1, Figure 3 illustrates the associations among these four kinds of entities based on class diagram in UML. Please note that Figure 3 is an extended class diagram, in which the symbol (for example, '1..*') on the left of the slash stands for the multiplicity of the association between two classes, and the number enclosed in a brace on the right of the slash stands for the number of the functions as listed in Table 1. For instance, 1..*/{11, 13} points out that the multiplicity of the association is 1..*, and the association could be set up by the functions of PEER.CollaboratorLocate() and PEER.ContextAware() that are assigned with 11 and 13 as numbered in Table 1.

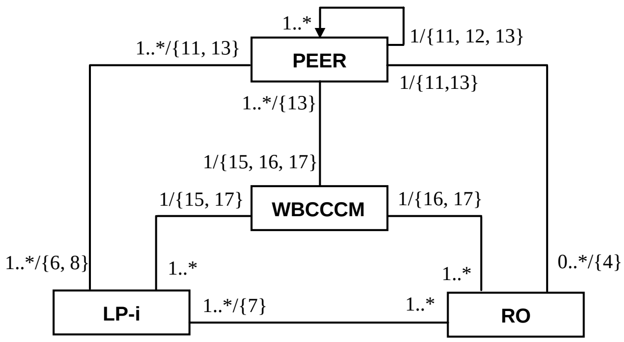

Figure 3. Specification of the associations between classes of RO, SP-i, PEER and WECCCM

6. Conclusions and future work

In this paper, cooperative cognition and its implementation under web environment is explored from software engineering perspective, rather than from AI perspective. Problem solving process is always accompanied by series of cognition evolution and knowledge interaction. Knowledge flow characterizes the knowledge interaction engaged in cooperative cognition. Context-aware knowledge interaction could efficiently promote cooperative cognition along positive way. In our future research, we would focus on exploring efficient knowledge index mechanism and intelligent methods for enhancing context-aware ability, especially for dynamic context-aware ability. Our expected target is to develop a prototype of intelligent service space to promote problem solving based on certain domain ontology and method ontology.

Acknowledgements

This paper is based on the Project 60303025 supported by NSFC, Jiangsu Provincial NSF research funds (No. BK2004411 and No. BK2005208), and "973" Project (NO. 2006CB303000).

References

[1]R.A.Wilson and F.C.Keil, The MIT Encyclopedia of the Cognitive Sciences*, The MIT Press Cambridge, Massachusetts London, England, 1999.*

[2]A.Satyadas, U.Harigopal, and N.P.Cassaigne, "Knowledge Management Tutorial: An Editorial Overview", *IEEE Transaction on Systems, Man, and Sybernetics (PartC)*, 2001, Vol.31, No.4, pp.429-437.

[3]M.E.Nissen and R.E.Levitt, "Agent-Based Modeling of Knowledge Flows: Illustration from the Domain of Information Systems Design", *Proceedings of the 37th Hawaii International Conference on System Sciences,* Hawaii, 2004.

[4]H.Zhuge, "A Knowledge Flow Model for Peer-to-Peer Team Knowledge Sharing and Management", *Expert Systems with Applications*, 2002, Vol.23, No.1, pp.23-30.

[5]V.Chiew and Y.X.Wang, "Formal Description of the Cognitive Process of Problem Solving", *Proceedings of the Third IEEE International Conference on Cognitive Informatics (ICCI'04)*, University of Victoria, Victoria, British Columbia, Canada, 2004.

[6]R.Sun, E.Merrill, and T.Peterson, "From Implicit Skills to Explicit Knowledge: A Bottom-up Model of Skill Learning", *Cognitive Science*, 2001, Vol.25, No.1, pp.203-244.

[7]P.Frasconi, M.Gori, M.Maggini, and G.Soda, "Unified Integration of Explicit Knowledge and Learning by Example in Recurrent Networks", *IEEE Transactions on Knowledge and Data Engineering*, 1995, Vol.7, No.2, pp.340-346.

[8]E.S.Patterson, D.D.Woods, N.B.Sarter, and J.Watts-Perotti, "Patterns in Cooperative Cognition", *Proceedings of COOP '98, Third International Conference on the Design of Cooperative Systems*, Cannes, France, 1998.

[9]I.Taylor, M.Shields, I.Wang, and O.Rana, "Triana Application within Grid Computing and Peer to Peer", *Journal of Grid Computing*, 2003, Vol.1, No.2, pp.199-217.

[10]S.Varadarajan and N.Ramakrishnan, *"Novel Runtime Systems Support for Adaptive Compositional Modeling in PSEs",* Future Generation Computer Systems*, 2005, Vol.21, No.6, pp.878-895.*

[11]H.Zhuge, "China's E-Science Knowledge Grid Environment", *IEEE Intelligent Systems*, 2004, Vol.19, No.1, pp.13-17.

[12]F.Berman, "From TeraGrid to Knowledge Grid", *Communications of the ACM*, 2001, Vol.44, No.11, pp.27-28.

[13]D. Fensel and E. Motta, "Structured Development of Problem Solving Methods", *IEEE Transactions on Knowledge and Data Engineering*, 2001, Vol.13, No.6, pp.913-932.

[14]D.Doval and D.O'Mahony, "Oveylay Networks: A Scalable Alternative for P2P", *IEEE Internet Computing*, 2003, Vol.7, No.4, pp.79-82.

[15]G.J.Fakas and B.Karakostas, "A Peer to Peer (P2P) Architecture for Dynamic Workflow Management", *Information and software technology*, 2004, Vol.46, No.6, pp.423-431.