HAL
archives-ouvertes.fr

# Collaboration in Agent Grid Based on Dynamic Description Logics

Limin Chen, Zhongzhi Shi

▶ To cite this version:

Limin Chen, Zhongzhi Shi. Collaboration in Agent Grid Based on Dynamic Description Logics. 6th IFIP TC 12 International Conference on Intelligent Information Processing (IIP), Oct 2010, Manchester, United Kingdom. pp.6-15, 10.1007/978-3-642-16327-2_5 . hal-01060353

HAL Id: hal-01060353
https://hal.inria.fr/hal-01060353

Submitted on 4 Sep 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution| 4.0 International License

# Collaboration in Agent Grid Based on Dynamic Description Logics

**L. Chen*, Z. Shi****

** The Key Laboratory of Intelligent Information Processing, Institute of Computing Technology Chinese Academy of Sciences Beijing 100190,China*
*Graduate University of Chinese Academy of Sciences, Beijing 100049, China*
*chenlm@ics.ict.ac.cn*

*** The Key Laboratory of Intelligent Information Processing, Institute of Computing Technology Chinese Academy of Sciences Beijing 100190,China*
*shizz@ics.ict.ac.cn*

ABSTRACT: *The global expansion of the Web brings the global computing; and the increasing number of problems with increasing complexity & sophistication also makes collaboration desirable. In this paper, we presented a semantics-based framework for collaborative problem solving in agent grid by coupling joint intention and dynamic description logics (DDL), our previous work to extend description logics (DL) with a dynamic dimension to model the dynamic world. Capabilities and attitudes of agents were captured by actions, and formulas in DDL respectively. Thus representation components in our framework were conferred with well-defined semantics by relating them to some domain ontologies. We could employ reasoning on actions in DDL to help agents to find proper colleagues when collaboration is necessary, and the philosophy underlying Joint Intention to bind their actions to achieve their overall goal. The main strengths of our framework include: i) finding probably helpful agents in a semantically accurate way due to the employment of semantic information; ii) going much closer to industrial implementations while retaining the main express power of classical joint intention model.*

KEYWORDS: *agent grid, dynamic description logics, joint intention, collaborative computing*

## 1. Introduction

To solve the problems with increasing complexity and sophistication, substantial time, effort and finances have been devoted to developing complex and sophisticated software systems, which places greater demand on the knowledge content and executing power of software agents. Collaboration might be a promising way to ease this tension. Furthermore, the need for collaboration among intelligent systems has also been fuelled by the global expansion of the Web and the advent of the paradigm of service oriented computing (SOC).

The emerging computational grid as well as agent technologies provides a good basis for building super collaboration frames for complex problem solving. The grid community has historically focused on infrastructure, tools, and applications for reliable and secure resource sharing within dynamic and geographically distributed virtual organizations, while the agent community has focused on autonomous problem solvers that can act flexibly in uncertain and dynamic environments [1]. As the scale and ambition of both grid and agent deployments increase, a convergence of interests in the agent and grid communities emerges: agent systems require robust infrastructure and grid systems require autonomous, flexible behaviours.

The past decades have witnessed researchers' attempts to apply agent technologies to realize the grid vision that enables resource sharing, provides the basic mechanisms for forming and operating dynamic distributed collaborations, or virtual organizations [2], and facilitates the unification of geographically dispersed computing systems to form a more powerful one. The most interesting work might be DARPA ISO's Control of Agent-Based Systems (*CoABS*) program, which firstly proposes the concept of 'Agent Grid' [3]. Shi et al. propose a model for agent-based grid computing from a point view of implementation, and develop *AGEGC*, an agent grid using a multi-agent environment platform based on the model [4]. Due to the merits inherited from agent and grid, agent grid has some advantages over the traditional approaches dealing with more demanding problems.

This paper addressed collaboration in agent grid by coupling dynamic description logics and joint intention. Sec.2 was devoted to a brief overview of dynamic description logics (*DDL*), a dynamic extension of *DL* [5]-[6]. In Sec.3, we proposed a collaboration model based on joint intention and dynamic description logics. Sec.4 concluded the paper with a discussion on the future work.

## 2. An overview of dynamic description logics

*DDL* is PDL-like dynamic extensions of *DL* [7]. Actually it is a family of languages, depending on the underlying *DL*. When necessary to be specific, we write *D-ALC*, *D-SHOIQ*, and the like. For simplicity, we choose ALCO as the underlying logic, and refer to the resulted *DDL* as *D-ALCO*.

***(Synatax)*** Primary alphabets of *D-ALCO* include: i) $N_R$ for role names; ii) $N_C$ for concept names; iii) $N_I$ for individual names; and iv) $N_A$ for atomic action names.

The concepts and roles in *D-ALCO* are the same as that in *ALCO* with "$C,D \rightarrow C_i \mid \{o\} \mid \neg C \mid (C \sqcap D) \mid \exists R.\, C$" & "$R \rightarrow P$", where $C_i \in N_C$, $o \in N_I$, $P \in N_R$. We use $\perp$, $\top$, $(C \sqcup D)$, and $\forall R.\, C$ to shorthand $(C \sqcap \neg C)$, $\neg\perp$, $\neg(\neg C \sqcap \neg D)$, and $\neg\exists R.\, \neg C$, resp..

Formulas in *D-ALCO* are built with: $\varphi,\psi \rightarrow C(u) \mid R(u,v) \mid \neg\varphi \mid \varphi \vee \psi \mid <\pi>\varphi$, where $u$, $v \in N_I$, $R \in N_R$ and $\pi$ is an action defined later. We define the logical connectives "$\rightarrow$" and "$\leftrightarrow$" in terms of "$\neg$", "$\vee$", as usual, and define "$[\pi]\varphi$" as "$\neg<\pi>\neg\varphi$". **ABox** and **TBox** in *D-ALCO* are the same as that in *ALCO*. Here we assume that readers have some familiarity with DL and omit some details of them.

An atomic action in *D-ALCO* is defined as $<\alpha, C, Pre, Eff>$, where i) $\alpha \in N_A$ is the name of the atomic action; ii) $C$ is a concept denoting the category the atomic action belongs; iii) *Pre* is a finite set of formulas; and iv) *Eff* is a finite set of formulas of the form $C(a)$ or $R(a,b)$, or their negations. *D-ALCO* actions are built with: $\pi, \pi' \rightarrow a \,|\varphi?|\pi\cup\pi'|\, \pi\,;\pi'|\pi^*$, where $a$ is an atomic action, and $\varphi$ is a formula i. An action box is a finite set of atomic actions in *D-ALCO*.

A domain specification is defined as: $DS$ = <*T, A, ActBox, SSet*>, where $T$ is a TBox, consisting of domain constraints; $A$ is an ABox for the initial world; *SSet* is the set of Aboxes for the world in a time slice; *ActBox* is an action box, the dynamic aspects of the world evolvements.

***(Semantics)*** A *D-ALCO* interpretation is a pair $(I,\mathcal{W})$, where $I=(\Delta^I,\cdot^I)$ is an *ALCO- interpretation* and $\mathcal{W}$ is a set of *ALCO- interpretations*. $I$ consists of nonempty domain $\Delta^I$ and mapping $\cdot^I$ that assigns atomic concepts to a subset of $\Delta^I$, each $a\in N_I$ to an $\Delta^I$-element, and each role to a subset of $\Delta^I\times\Delta^I$. The semantics of *D-ALCO* is formally defined in Table 1, where $A$ is an atomic concept, $R$ a role, $C$, $D$ concepts, $\alpha$ is an atomic action, and $\pi_1$, $\pi_2$ denote actions:

Table 1: Semantics of *D-ALCO*

1) $A^{I,\mathcal{W}}=A^I$; 2) $R^{I,\mathcal{W}}=R^I$; 3) $\{u\}^{I,\mathcal{W}}=\{u^I\}$;

4) $(\neg C)^{I,\mathcal{W}}=\Delta^I\backslash C^{I,\mathcal{W}}$; 5) $(C\sqcap D)^{I,\mathcal{W}}=C^{I,\mathcal{W}}\cap D^{I,\mathcal{W}}$;

6) $(\forall R.\,C)^{I,\mathcal{W}}=\{x\in\Delta^I|\forall y.(x,y)\in R^{I,\mathcal{W}}$ implies $y\in C^{I,\mathcal{W}}\}$;

7) $(\exists R.\,C)^{I,\mathcal{W}}=\{x\in\Delta^I|\exists y.(x,y)\in R^{I,\mathcal{W}}$ and $y\in C^{I,\mathcal{W}}\}$;

8) $(\alpha)^{I,\mathcal{W}}=(Pre,Eff)^{I,\mathcal{W}}=\{(I,I') \mid I$ satisfies each $\varphi_i\in Pre$, and $C^{I',\mathcal{W}}=(C^{I,\mathcal{W}}\cup\{u^I \mid C(u)\in Eff\})\backslash u^I \mid \neg C(u)\in Eff\}$; $R^{I',\mathcal{W}}=(R^{I,\mathcal{W}}\cup\{(u^{I,\mathcal{W}},v^{I,\mathcal{W}}) \mid R(u,v)\in Eff\})\backslash\{(u^{I,\mathcal{W}},v^{I,\mathcal{W}}) \mid \neg R(u,v)\in Eff\}\}$;

9) $(\varphi?)^{I,\mathcal{W}}=\{(I,I)|\, I$ satisfies $\varphi\}$;

10) $(\pi_1\cup\pi_2)^{I,\mathcal{W}}=(\pi_1)^{I,\mathcal{W}}\cup(\pi_2)^{I,\mathcal{W}}$;

11) $(\pi_1;\pi_2)^{I,\mathcal{W}}=\{(I,I') \mid$ there exists some $I_t\in\mathcal{W}$ such that $(I,I_t)\in(\pi_1)^{I,\mathcal{W}}$ and $(I_t,I')\in(\pi_2)^{I,\mathcal{W}}\}$ ;

12) $(\pi_1^*)^{I,\mathcal{W}}=$ the reflective transitive close of $(\pi_1)^{I,\mathcal{W}}$.

We still need to make precise the meaning of "satisfies" in 8) and 9). The satisfaction of a formula $F$ in $(I,\mathcal{W})$, written as $(I,\mathcal{W})\vDash F$, is defined in Table 2:

Table 2: The Satisfaction of $F$ in $(I,\mathcal{W})$

1) $(I,\mathcal{W})\vDash C(a)$ iff $a^{I,\mathcal{W}}\in C^{I,}$; 2) $(I,\mathcal{W})\vDash R(a,b)$ iff $(a^{I,\mathcal{W}},b^{I,\mathcal{W}})\in R^{I,\mathcal{W}}$;

3) $(I,\mathcal{W})\vDash\neg\varphi$ iff $(I,\mathcal{W})\nvDash\varphi$; 4) $(I,\mathcal{W})\vDash\varphi\vee\psi$ iff $(I,\mathcal{W})\vDash\varphi$ or $(I,\mathcal{W})\vDash\psi$;

5) $(I,\mathcal{W})\vDash<\pi>\varphi$ iff there exists an $I'\in\mathcal{W}$ such that $(I,I')\in\pi^I$ and $(I',\mathcal{W})\vDash\varphi$.

A formula $\varphi$ is satisfiable w.r.t a TBox T, if a *D-ALCO* interpretation models T and $\varphi$. The main reasoning tasks in *DDL* can be reduced to satisfiability checking of formulas, which is *decidable* in *D-ALCO* [5]. Due to the space limitation, we do not elaborate on reasoning tasks in *D-ALCO* (See [5] for further details).

## 3. Collaboration by joint intention and dynamic description logics

In this section, we proposed a collaboration model which employed reasoning on actions in *DDL*s to find probably helpful agents and the philosophy underlying Joint Intention to bind their teamwork.

Intention is the notion to characterize both our action and our mental states in our commonsense psychology [8]. While in a collaboration environment, intentions are not sufficient in that teamwork involves more than just the union of the simultaneous individual actions even if they are coordinated [9]. As a joint commitment to act in a shared belief state, joint intention binds team members, and enables the team to overcome misunderstandings and surmount obstacles [10].

Cohen and Levesque devise a formalism to express joint intentions with strong semantics and a high level of maturity [9, 11-12]. The main drawback of C&L's formalism is that it stays too far from the industrial application. In our model, rather than as modal operators to model the attitudes of community members, beliefs and goals of agents are expressed as *DDL*-formulas and simply grouped into the corresponding classes. Intentions of an agent are defined as its aim to perform some actions to fulfil some goals. The joint attitudes or intentions of a team are defined in terms of those of its members. Capabilities of agents are represented by *DDL*-actions, thus the problem of capability matching, i.e., to find proper peers to do proper jobs, can be solved by *DDL*-reasoning. Then the philosophy underlying joint intention is employed to bind their actions to achieve their overall goal.

### 3.1. *Conceptualizing Agent Grid*

This subsection gives some primary definitions about agents and the specifications of agents and agent grid. In the sequel, we use Greek symbols such as $\alpha$ $\beta$ $\gamma$ , possibly with subscripts, to name agents.

***Defn 3.1 (Belief)*** Let $w$ be an environment, for an agent $\alpha$, a formula $b$ is a belief of $\alpha$ if $\alpha$ believes to be true in $w$, where $s$ is the ABox describing $w$. We denote by $B_{\alpha,s}$ the set of $\alpha$ 's beliefs in $w$.

We assume agents are rational but not logically omniscient. So belief of an agent in any environment is consistent, but may be not close under logical references.

Beliefs of an agent are environment-dependent, i.e., changing along with the evolvement of environments. We assume that agents have records of their past beliefs. The sequence of beliefs of an agent $\alpha$ in an evolvement path is a belief record of $\alpha$.

***Defn 3.2 (Mutual Belief)*** For an environment *w* described by state *s*, formula *b* is a mutual belief of agent $\alpha$ and $\beta$ in *w* if and only if $b \in B_{\alpha,s} \cap B_{\beta,s}$.

Of course, agents have capabilities to change the environments; and we assume that these capabilities are the only factors that cause environment changes. This assumption rules out the possibility of unpredictable changes on environments.

***Defn 3.3 (Capability)*** A capability of agent $\alpha$ is an action $\pi$ in *DDL* describing some ways the agent can change the environments, written as $<\alpha, \pi>$.

We can also safely state that an agent changes the environments with the aim to enter into environments with certain properties. Typically, agents' goals can be classified into two types: maintenance goals and achievement goals. The former are something initially true and are not allowed to be changed over the evolvements of the system, while the latter are currently false and agents attempt to make true later. For example, consider an environment where a frail vase stands on a table, and if only one side is lifted, then the vase slides down and crumbles. If a team of agents have an achievement goal to lift the table, they may simultaneously to keep the table flat in the lift process, which is a maintenance goal. In this paper we focus on achievement goals, and assume that an agent retains a goal until the goal has been fulfilled or the motivation becomes irrelevant (persistent goals).

***Defn 3.4 (Goal)*** A goal of an agent $\alpha$ relative to m is an ordered pair of the form $<m, g>$, where *m* is a formula in *DDL*s stating the motivation, and *g* a formula characterizing the desirable environments of $\alpha$.

Agents' goals direct their attempts and determine the actions agents take to fulfil these goals. Intentions of an agent are persistent goals to do some actions.

***Defn 3.5 (Intention)*** An intention of an agent $\alpha$ relative to *m* is an ordered pair of the form <*m*, *actExp*>, where i) *m* is a DDL-formula stating the motivation(usually, a goal of $\alpha$); and ii) *actExp* is a *DDL*-action.

Agents' intentions are goal-dependent. Given a goal and a set of actions available to perform, the *planning problem* is to find an action list whose execution reaches an environment the goal holds. Such plans may pre-exist as a part of agents' knowledge, or can be computed dynamically.

Joint intentions are a special kind of intentions whose component actions are relative to other agents.

***Defn 3.6 (Joint Intention)*** A joint intention of agent $\alpha$ relative to *m* with agents $\{\beta_1, \beta_2, \ldots, \beta_k\}$ is a ternary <*m*, *π*, *Ags*>, where i) *m* is a formula; ii) $\pi$ is an action; and iii) *AgS* is the set concerned agents.

An agent cannot drop a joint intention arbitrarily, and once the intended action has been performed or becomes irrelevant, the agent must make its new mental state about the intention known to the relative agents before its discard of the intention.

***Defn 3.7 (Agent Specification)*** An agent is described by a tuple: $<\alpha, G, B, C, I, JI, K_{on}PC, K_{on}E>$, where i) $\alpha$: the agent's name; ii) *G*: the set of agent's goals; iii)

*B*: agent's current beliefs; iv) *C*: $\{\pi|<\alpha, \pi>\}$, the set of actions describing $\alpha$'s capabilities; v) *I*: the set of agent's intentions; vi) *JI*: the set of agent's joint intentions; vii) $K_{on}PC$: a set of capabilities of other agents; viii) $K_{on}E$: a subset of the domain specification, the knowledge about the environment.

An agent grid is specified by the following definition:

***Defn 3.8(Agent Grid)*** An agent grid is a tuple: $<W, T, L, AgS>$, where i) *W:* the set of possible environments; ii) *T:* a subset of $W\times W$, the binary relation on *W,* for $e_i$, $e_j \in W$, $< e_i, e_j > \in T$ iff some agent in the agent grid can perform some actions transforming $e_i$ to $e_j$; iii) $L$: $W \rightarrow 2^{SSet}$, a labeling function, i.e., mapping any $w \in W$ to a subset of *SSet*, the set of ABoxes in domain specification (cf. Sec. 2); and iv) *AgS:* the set of community members of the grid.

*T* is intended to model transitions on environments and is transitive, whose transitive close is ***accessibility***, denoted by *A*. Evolvement paths, or paths for short, are defined as chains in the partial order set $<W, A>$.

### 3.2. *Planning for goals*

Once an agent $\alpha$ has a goal *g*, the next thing for $\alpha$ is to get a plan for *g*. As mentioned above, plans may pre-exist as a part of agents' knowledge, or can be computed dynamically. Here, we propose a planning algorithm where the *planning problem* can be reduced to the satisfiability checking of *DDL*-formulas [5].

For agent $\alpha$ with a goal *g*, $\alpha$ computes a plan for *g* based on its knowledge on the current environment, on its capability and on its peers' capabilities. Before presenting the algorithm, we first give some results concerned, and the interested reader can refer to [5] for deeper investigations on topics of reasoning on actions in *DDL*s.

Suppose $PlanCandidate=<\alpha_1, \alpha_2,\ldots, \alpha_k>$ is an action list, and that *s* is the ABox for the initial environment. In the sequel, we denote by *Conj(s)* the conjunction of member formulas in a formula set *s*. Let us explore the following two formulas.

$1)[(a_1 \cup \ldots \cup a_k)^*]\Pi \wedge Conj(s) \rightarrow \langle PlanCandidate \rangle true$ , where $\Pi$ is $\wedge_{i=1}^{k}(Conj(P_i) \rightarrow < a_i > true)$ and $P_i$ the precondition of $a_i$ for each $i$: $1 \leq i \leq k$.

The formula $[(a_1 \cup \ldots \cup a_k)^*]\Pi \wedge Conj(s)$ can be viewed as an evolvement axiom: started from the environment described by *s*, during the process of environment evolvements, any action can be performed if its preconditions are satisfied. The whole formula says that *PlanCandidate* can be executed in the world respecting the evolvement axiom, and its validity requires its negation unsatifiable :

$$[(a_1 \cup \ldots \cup a_k)^*]\Pi \wedge Conj(s) \wedge \neg\langle PlanCandidate \rangle true \quad (1)$$

$2)Conj(s) \rightarrow [PlanCandidate]g$ .

This formula states that after the execution of *PlanCandidate* in the environment described by *s*, the goal *g* holds, whose validity requires the following unsatifiable.

$$Conj(s) \wedge \neg[PlanCandidate]g \quad (2)$$

The above facts can be employed by agent $\alpha$ to compute a plan for a goal from a set of actions. Table 3 shows the algorithm to compute a plan to fulfill a goal *g* from the set *CapSet* of the available actions, which travels all the possible action lists to find a plan, if any. All inputs of the algorithm are from the *G*, *C*, $K_{on}PC$ and $K_{on}E$ of $\alpha$ , including the ABox *s* describing the current environment, TBox *T* capturing the domain axioms, *CapSet*, the set of actions. In the algorithm, <*PlanCandidate*, $\alpha_i$ > denotes the action list resulted by appending $\alpha_i$ to the rear of *PlanCandidate*.

Table 3: Plan computation

```
Algorithm 1 CapabilityComposition(s, T, CapSet, g)
  [Input: state s, TBox T, Capabilities of agents CapSet,and goal formula g.]
  [Output: a unempy action sequence as a plan; or false (as a failure).]
  [Begin]
  Initialize queue QueOfPlanCandidates;
  Enqueue QueOfPlanCandidates with empty action list ⟨⟩;
  while (QueOfPlanCandidates is unempty) do
    Dequeue(QueOfPlanCandidates, PlanCandidate);
    if (Formula (1) is unsatifiable w.r.t T) then

      if (Formula (2) is unsatifiable w.r.t T) then
        Return PlanCandidate as a successful plan;
      else
        for each αi: enqueue QueOfPlanCandidates with new candidate
        ⟨PlanCandidate, αi⟩;
      end if;
    end if
  end while
  Return false;
  [End]
```

### 3.3. *Working collaboratively*

When a task arrives, the agent concerned does a means-end analysis based on its capabilities, current intentions, and environment information to see whether the task can be solved locally. Suppose that $\alpha$ has belief $B_{\alpha,s}$ in the environment described by state *s*. For each <*m*, *g*>, if $K_{on}E$, $B_{\alpha,s} \models m$, then $\alpha$ chooses *g* as a current goal. Then $\alpha$ computes plans for these goals and takes these plans as its current intentions. After a process of capability matching between the intended actions and its capabilities, $\alpha$ realizes the need for collaboration, and a further negotiation with other agents is invoked. If the task can be solved locally, then a set of local objectives are identified. Such objectives may contradict the agent's current intentions, thus a phase to check and resolve the inconsistency is need. *Goals* and *Beliefs* of the agent are also involved in the phase. Failures in this phase can also lead to the considerations of the task's feasibility. Then, a set of consistent intentions are generated and added to the agent's current intentions. Once a new (joint) intention is formed, another kind of *Inconsistency Checker & Resolver* will be invoked to ensure the consistency between individual intentions and joint ones. Figure 1 sketched such a process.

We further investigate what is involved when agents prefer to work collaboratively. The main consideration includes the following two questions:

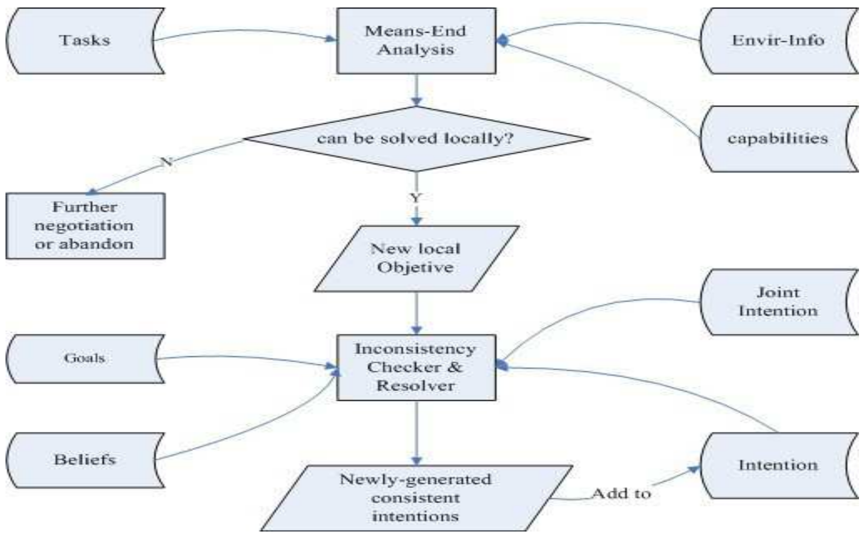


Figure 1 An adoption process of a new task

***What conditions should be satisfied before collaboration can proceed****?* Once an agent decides to achieve a goal through teamwork, the first thing for this originator agent is to figure out the capable and potential interested agents. The key point is to compute a semantically accurate match between the actions to be performed and those peers' capabilities.

The originator employs the classical reasoning tasks in description logics to find the capable agents by a matching between the intended actions and its *knowledge on peers' capabilities*. For the moment, we just simply employ the concepts to categorize the intended actions and agents' capabilities, and the match between agents' capabilities and the intended actions can be solved by a reduction to the subsumption between concepts: the agent with the capability *C* has the ability to perform the action *T* if *C* subsumes *T*. Further, classifying agents' capabilities by relating them to concepts in some ontology also structures the originator's knowledge about agents' capabilities in a hierarchical way, and such a structure can be taken into the originator's account in the process of finding and ranking the potential participants. Communication becomes necessary in finding the interested agents. The originator sends collaboration proposals to its target agents, promises to help in future may also be offered. The recipients undertake some rounds of the process depicted in Figure. 1, and decide the proposals should be accepted or refused. Finally, agreement on the details about the tasks will be reached through negations. Once such an agreement has been reached, the intentions concerned become joint intentions of the agents involved with its co-workers. After the above phase, all the participants should acknowledge their acceptances to their own parts in the final plan. Only after all the participants have promised to contribute to the whole plan and all the details have been fixed, collaboration can proceed.

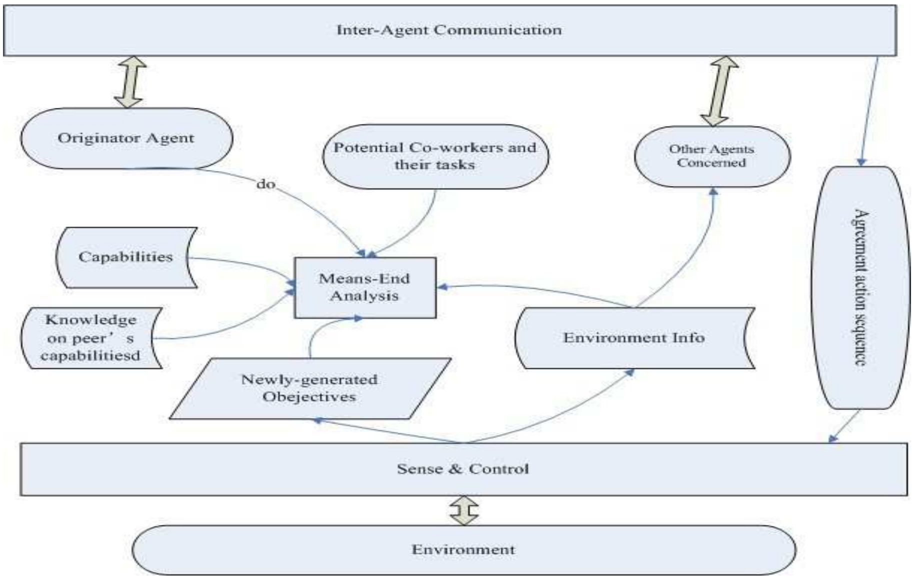

Figure. 2: Collaboration in agent arid

***When and how agents should interact with their co-workers?*** Once cooperation begins, the participants are jointly committed to their individual goals and the overall goal until the collaboration is finished. When a participant comes to believe the goal is finished, or impossible to finish, or the motivations are not relevant any longer but that is not a mutually known, then the participant takes a new persistent goal to make its new discovery a mutual known. Then communications between the observant participant and its fellows are initiated, and the persistent goal terminates if all participants acknowledge their new known about the new status of the goal. The community deals with the newly-emerged situation accordingly, be it a reconsideration of the validity of the goal, a totally discard, a further negation or a new attempt to something less ambitious. Figure 2 sketches the architecture of collaboration in agent grid.

## 4. Conclusions and future work

We have presented a framework toward collaboration in agent grid. The framework is based on our ongoing work on extending description logics with a dynamic dimension and the philosophy behind Cohen & Levesque's joint intention model. Rather than as modal operators to model the attitudes of community members, we just simply group beliefs, goals, and intentions into corresponding classes. The joint attitudes of a community are defined in terms of the joint ones of its members. The problem of capability matching, i.e., finding the peers to do proper jobs, can be solved by *DDL*-reasoning; and the philosophy underlying joint intentions is employed to bind the team. While the collaboration model employs a *DDL*-based planning and capability matching to choose the potential participants and joint intention to bind and coordinate their actions, its higher-level nature in abstraction allows the potential tailors for proper fitness for the problems at hand.

One future work is to investigate the relation among *DDL*-actions further to compute a sophisticated specification of agents' capabilities, rather than the preliminary classifications by relating actions to *DL*-concepts in this paper. It is helpful to capture the partial matching with quantification degrees and to find the potential co-workers.

## 5. Acknowledgements

This work is supported by the National Natural Science Foundation of China (No. 60775035, 60933004, 60903141, 60970088), National Basic Research Priorities Programme (No. 2007CB311004), and National Science and Technology Support Plan (No. 2006BAC08B06).

## 6. References

1. I. Foster, N. Jennings, and C. Kesselman, "*Brain Meets Brawn* " in AAMAS-2004 pp. 8-15

2. I. Foster. and C. Kesselman, "*The Grid*", 2nd ed. Morgan Kaufmann, 2003

3. F. Manola. and C. Thompson, "*Characterizing the Agent Grid*." Ava at http://www.objs.com/agility/tech-reports/990623-characterizing-the-agent-grid.html, 1999

4. Z Shi., H Huang, J Luo, et al. "*Agent-based Grid Computing*" Applied Mathematical Modeling, 30(7): 629-640, 2006

5. L Chang, F Lin, and Z Shi, "*A Dynamic Description Logic for Representation and Reasoning about Actions*." in KSEM-2007, Australia. pp: 115-127

6. Z. Shi, M. Dong, Y. Jiang,et al. "*A Logic Foundation for the Semantic Web*," Science in China, Series F, vol. 48(2):161-178, 2005

7. F Baader, D Calvanese, D McGuinness, et al, "*The description logic handbook*" Cambridge University Press, 2003

8. M. Bratman, "*Two Faces of Intention*," The Phil. Review, 93:375-405, 1984

9. H Levesque, and P Cohen, "*Teamwork*," Nous ,vol. 25, pp. 487-512, 1991

10. P Cohen, & H Levesque, "*Confirmations and Joint Action*," in IJCAI-1991, pp. 951-957

11. P Cohen, & H Levesque, "*Intention is choice with commitment*," AI, 42:213-261, 1990

12. H Levesque, P Cohen, & J Nunes, "*On Acting Together*," in AAAI-1990, pp.94-99