A Self-Organizational Management Network Based on Adaptive Resonance Theory[1]

Ping Jiang and Quentin Mair

Department of Computing, Glasgow Caledonian University
Glasgow G4 0BA, UK
{p.jiang, q.mair}@gcal.ac.uk

Abstract. This paper presents an organizational network for product configuration management within the context of Virtual Enterprise. Actors, from high level strategy making actors to low level physical devices, can advertise their own skill and knowledge and seek for partners to form dynamic alliances in a community. The network is organized based on Adaptive Resonance Theory(ART) which was originally used for unsupervised neural network learning and which allows the organization and cooperation of such product development alliances to be more flexible and adaptable. Some characteristics, which are inherent in real enterprises or society, such as self-organization, unsupervised learning, competition between actors are exhibited in the ART-based organization network and are the keys for evolution and development of enterprises.

1 Introduction

During long product life-cycles, there are huge numbers of actors involved in product design, development, and deployment. They can be distributed at different geographical sites and different information is shared between them. If such a system is designed from a global viewpoint it becomes inflexible, unchangeable and is too difficult to manage. However, take a look at each actor, such as a programmer, a manager, etc., who is involved in the product life cycle. There are not usually too many direct connections to control and usually a very explicit and clear task can be assigned to each actor. An actor-oriented distributed management structure may give a management system more flexibility, changeability, and interoperability, where the actor is any person or device involved in product management, manufacturing and customer service[1]. The UML analysis from an actor's perspective can directly guide people to realize their own cooperation agent that is capable of advertising their knowledge for both seeking for and cooperating with partners. A multi-agent system can then be constructed to realize collaborative work between actors during product life-cycles.

---

[1] This work is supported by DIECoM (Distributed Integrated Environment for Configuration Management), an IST project funded under the EC's Framework V programme (http://www.diecom.org)

R. Kowalczyk et al. (Eds.): Agent Technology Workshops 2002, LNAI 2592, pp. 211–225, 2003.

A key technology of actor-oriented product management is service discovery from a group of actors, where the capability of each actor can be described, advertised, and discovered by other actors who are seeking for partners for a given task. It has a similarity to the service matchmaking among agents for information retrieval on the World Wide Web. However, for information retrieval, current research mainly focuses on passive matching of context and profile by the advertisement[2][3] with less focus on active learning and adaptive ability. Paper [4] attempts to endow WWW information retrieval with learning ability by using BP neural networks that can capture knowledge about users' interests and preferences; although the training of the networks might be quite slow and cumbersome. At the same time other drawbacks such as local optimum, missing semantic relation by hashing encode etc., exist. This kind of active learning and adaptive ability may be more important in product management in Virtual Enterprises. Within the context of a Virtual Enterprise, cross-organizational PCM (Product Configuration Management) faces dynamically changing environments and dynamically changing roles within organization. An actor with more adaptability and flexibility will be more powerful and useful in a competitive society. This requires that actors should have dynamic reorganization ability. The concept of dynamic reorganization allows agents to reconfigure and/or restructure their system in response to environmental or system changes.

This paper proposes an organizational network for dynamic partnership in a dynamic environment with adaptive ability, learning ability, competitive ability. It is based on Fuzzy Adaptive Resonance Theory (Fuzzy ART) that is a neural network model and is proposed by Carpenter, Grossberg, and Rosen[5] for clustering binary or analog data. The ART network is a self-organizational network and is based on a "winner-takes-all" competitive principle. It has unsupervised learning ability and adaptive ability for data clustering. In this paper, each actor involved in product configuration management is considered as a neuron in an ART network that is the proposed organizational network of product configuration management. On presentation of a task (input vector) from a contract provider, who is seeking for an appropriate partner for the task, the actor whose advertisement of its ability (connection weight) is closest to the input vector will become candidate of the contract, and will be allowed to learn the demands of the contract provider, i.e. modify its ability description (called connection weight in ART). After repeatedly advertising the task in the product management community, the network will adapt to the demand of the contract provider and store a prototypical element of each demand in the connection weights. Then, an actor with previous experience in a specific area will have more chance for a task in that area and will have more expertise to do it. In multi-agent system, agents should have information about their environment. Actors should talk each other based on an ontology of a community, where OIL(DAML-OIL)[6] can be used to represent the knowledge and information of each actor. In order to use numeric representation instead of symbolic representation of knowledge in ART networks, a feature vector of a local ontology is defined by using semantic distance of concepts and fuzzy inference is introduced for analysis of similarity of concepts. Simulations are carried out to verify features of the proposed management network.

2 Actor-Oriented Product Management Systems

In the life-cycle of product management, each actor goes through a sequence of partnership creation, configuration, operation, dissolution, repeatedly. At the first stage, an actor may initiate a product development partnership and may advertise requirements of the task to a community. For instance, "I need a software engineer for a driver program coding" and "I need another hardware engineer for communication board design who has experience in IEEE 802.11b". The actor is called an "initiator" of the product development partnership. At the same time, actors as participators are advertising their ability in the community for getting the task assigned by the initiator. After finding a candidate for the desired task, the initiator will create a partnership with it. Both sides can modify the advertised ontology to adapt specific tasks through negotiation. Then the participator who gets the contract may decompose the task into subtasks and become a new initiator within its local community.

Therefore, each actor plays two roles in a system, on one side, it is an initiator of a partnership in a multi-agent community who seeks for qualified candidate to complete a given task, and on the other side, it is a participator in another multi-agent community who advertises its ability and seeks for tasks from other actors. Suppose an actor a_j belongs to two communities, as a task initiator in $C_I = \{a_{I1} \quad a_{I2} \quad \dots \quad a_{IL}\}$ and as a task participator in $C_P = \{a_{P1} \quad a_{P2} \quad \dots \quad a_{PM}\}$, respectively. An actor is capable of connecting two communities together and decomposes a given task $t_{input}(C_p)$ from C_p into a series of subtasks $t_{output}(C_I) = [t_{o1} \quad t_{o2} \quad \cdots]$ in C_I and local tasks t_{local} as

$$\{t_{output}, t_{local}\} = S_j(t_{input}) \tag{1}$$

where the task t_{input} is assigned by actors in C_p , i.e., $t_{input}=t_o(a_{pi})$, $i=1\dots M$, and t_{output} is the decomposed subtasks of the actors in C_I. The t_{local} is the tasks completed locally.

Within a community C, in order to facilitate cooperation and understanding amongst agents, there is an ontological definition Ω_C that defines the semantics of commonly used concepts and terminologies. The ontology can be defined using DAML-OIL(http://www.daml.org/), Suppose that the ontology in a community C can be defined as:

$$\Omega_C \equiv R(e_1, e_2, \dots, e_N) \tag{2}$$

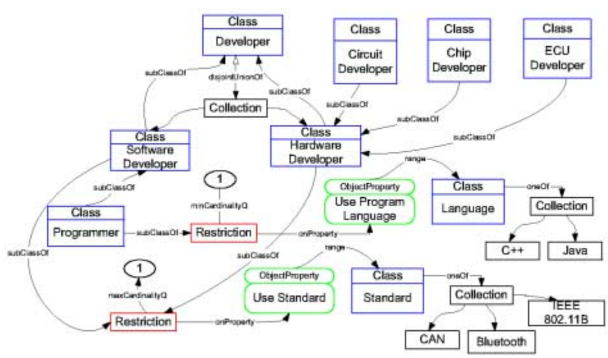

Fig. 1. Ontology of a community for electric device development

where $e_1 \ldots e_N$ are entities (concepts, terminologies, properties) related to the community and $R(\bullet)$ is the relationship between the entities.

An example of an ontology defined using DAML-OIL in the community of electric device development is shown in Fig. 1, where the entities of the society are {Developer, Software Developer, Hardware Developer, Programmer, Language, Standard, C++, Java, CAN, Bluetooth, IEEE 802.11b, ECU developer, Chip Developer, Circuit Developer} and reflect the content of the community.

As a contract provider, an initiator a_{Ii} can then advertise its demand for a subtask by using entities defined in the Ω_C. At the same time, the participators a_{Pj}, j=1…M, in the community expect a contract and advertise their capability for automatically creating alliances. The content of the advertisements of both initiator and participator is a semantic knowledge description or an instance of the local ontology based on the definition of Ω_C in a community C. It can also be written in OIL(DAML-OIL) and is called a local ontology. An example of a hardware developer as a participator can announce himself as "An ECU developer using CAN for a given task" as shown in Fig.2, who is seeking for suitable task from the community of electric device development. In the same way, an initiator can announce a task to seek for a partner who can do it, for instance, who should be a programmer with knowledge and experience of Bluetooth and C++ as shown in Fig.3.

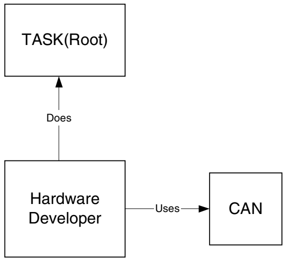

Fig. 2. Advertisement of a participator

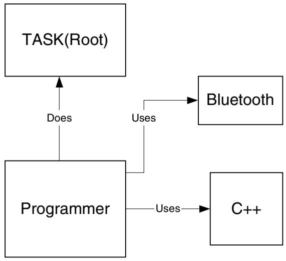

Fig. 3. Advertisement of an initiator

Note that, in the advertisements of both initiator and participator, there is a root object. For an initiator, the root corresponds to a subtask that an initiator can provide and its local ontology describes what kind of subtask it is. For a participator, the root corresponds to a task that an actor can do and the local ontology describes participator's ability to do the task. All the objects inside an actor and the relationship amongst objects should be defined around the root for its demand or capability advertisement.

For each actor, we define a feature vector for any task advertisement t as a numeric representation of the semantics of its local ontology:

$$V(t)=[s_1,s_2, \ldots, s_N]^T \tag{3}$$

The component s_i of $V(t)$ has a one-to-one correspondence to the entity e_i of the ontology. The $s_i \in [0,1]$ is the semantic closeness between e_i and the root(task) in an OIL based advertisement. It is an inverse of semantic distance. One possible form of this inverse can be

$$s_i = \begin{cases} e^{-\alpha DIS(e_i, root)} & \text{if } e_i \text{ is appeared in the local advertisement} \\ 0 & \text{if } e_i \text{ is not appeared in the local advertisement} \end{cases} \tag{4}$$

where the Dis(e_i,*root*) is a semantic distance between entity e_i and the *root* calculated in the local ontology, e.g., in Fig.2 for a participator and in Fig.3 for an initiator. The α is a steepness measure[7], in fuzzy system, which is often selected to be -7/MAX(Dis) because $e^{-7}\approx 0$ when Dis(e_i, *root*) achieves its maximum.

In order to deal with automation of knowledge extraction, semantic distance or similarity between concepts has been researched in recent years, such as semantic web matchmaking[2][8] and conceptual clustering of database schema[9]. In [9], a general semantic distance is defined mathematically as an application of $E\times E$ into R^+, where E is a set of objects in a community ontology Ω_C, with the following properties:

i. $\forall x \in E, \forall y \in E,$ $\quad \mathrm{Dis}(x, y) = 0 \Leftrightarrow x = y$

ii. $\forall x \in E, \forall y \in E,$ $\quad \mathrm{Dis}(x, y) = \mathrm{Dis}(y, x)$

iii. $\forall x \in E, \forall y \in E, \forall z \in E$ $\quad \mathrm{Dis}(x, y) \le \mathrm{Dis}(x, z) + \mathrm{Dis}(z, y)$

The semantic distance can be use to characterize similarities or dissimilarities between two objects. Usually, a distance between two objects is the shortest path between them. The path description greatly depends on the view point of observation. Different types of semantic distances are proposed in [9]: visual distance, hierarchical distance and cohesive distance, etc.. For instance, a visual distance is very close to the graphical representation of a model. It is defined from a view-point that two objects semantically linked by a relationship (or a generation) are very often graphically close. In fact, anyone can define his semantic distance from a viewpoint which is of relevance to his problem. For instance, in a DAML-OIL ontology, two classes that are disjointWith should be given a large semantic distance but two that are sameClassAs should have a zero semantic distance.

Therefore, on the basis of definition of semantic distance, the feature vector can describe the requirements of an initiator and the ability of a participator from a designer's view. It gives the information both about how many entities are related to a task and about what kind of relationship is between the involved entities. Partner seeking becomes a matchmaking process between advertised feature vectors of initiators and participators in a community. An ART network can be constructed for automation of this matchmaking which considers adaptive capability, learning ability, and competitive properties.

3 Self-Organizational Management Networks

In the aforementioned actor-oriented product management system, how to organize the actors to form a partnership becomes a key for product management. In this section, a self-organizational ART network is proposed for adaptive and dynamic partnership seeking. Suppose that an actor a_i is a partnership initiator in a community C_I, who has decomposed a task t_{input} into a series of subtasks $t_{output}(C_I) = \begin{bmatrix} t_{o1} & t_{o2} & \cdots & t_{oK} \end{bmatrix}$ and is seeking for partnership in a community C_I. As a contract provider, it asks the community whether any actor has the desired capability and interest for the given tasks. The desired capability is described by the feature vectors $X_d(t_{oj}),j=1\ldots K$, as defined in equation (3):

$$X_d(t_{oj})=[x_{d1}, x_{d2},\ldots,x_{dN}]^T, \quad x_{di}\in[0,1], j=1\ldots K, \ i=1\ldots N \tag{5}$$

The vector $X_d(t_{oj})$ specifies what kind of abilities and contents are required for doing the subtask t_{oj}, where x_{di} is between 0 and 1; where a bigger x_{di} means the ability of e_i is more important for the subtask. Then, the series of feature vectors to describe the desired subtasks t_{oj} , $j=1\ldots K$, are posted in the community to seek for qualified candidates.

At the same time, the participators a_j, $j=1\ldots M$, in the community C_I are advertising their knowledge and capability to seek for a suitable task from initiators. The representation of the ability of each participator can also be written in OIL(DAML-OIL) and the corresponding feature vector can be calculated based on the semantic distance between entities and the root (the ability of a participator):

$$W(a_j)=[w_{j1}, w_{j2},\ldots,w_{jN}]^T, \quad w_{ji}\in[0,1], j=1\ldots M , i=1\ldots N \tag{6}$$

This feature vector can be explained by the fuzzy cognitive map proposed in [10], which describes the relationship between concepts using connection weights. The w_{ji} in equation (6) describes a connection weight between the concept "ability of the participator a_j" and the concepts of e_i , $i=1\ldots N$, defined in Ω_C. A higher weight indicates a stronger ability in the area related to the concept e_i. Then, the advertisement of each actor, equation (6), can be depicted as a fuzzy network shown in Fig.4. The weight or feature vector of an actor connects the "ability of Actor j" with the concepts of $e_1\ldots e_N$ in the community. For instance, in the community of Fig. 1, a weight vector of {0.3, 0.5, 0.2, 0.6, 0.5, 0.2, 0.9, 0.8, 0.1, 0.4, 0.3, 0.1, 0.1, 0.1} implies that an actor is excellent at the coding using C++ or Java but relatively poor at hardware work.

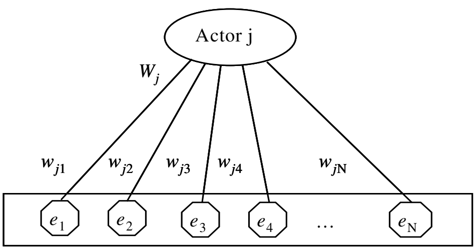

Fig. 4. Advertising ability of an actor

Then, a 3-layers ART network can be constructed for adaptive and dynamic partnership seeking as shown in Fig.5. The layers F0, F1, and F2 are the input layer, comparison layer and recognition layer, respectively. The input layer F0 gets the demand $X_d(t_{oj})$ one by one from an initiator. Each participator advertises its capability in the comparison layer F1 for competence comparison. The nodes in F0 and F1 are composed of the entities of the ontology which is the same as the bottom layer of Fig.4. The corresponding nodes of layer F0 and F1 are connected together via one-to-one, non-modifiable links. The nodes in recognition layer F2 are the actors (participators) of the community C who are candidates for the given tasks. Altogether, there are M nodes corresponding to M participators in the community C, each one represents a characteristic competence of an individual actor and will take part in competition for the posted tasks from the initiator. There are top-down and bottom-up weight connections W_j between the nodes in the layer F1 and the nodes in the layer F2. Initially, the weight W_j is the advertised fuzzy weight vector of a participator a_j in Fig.4. During the

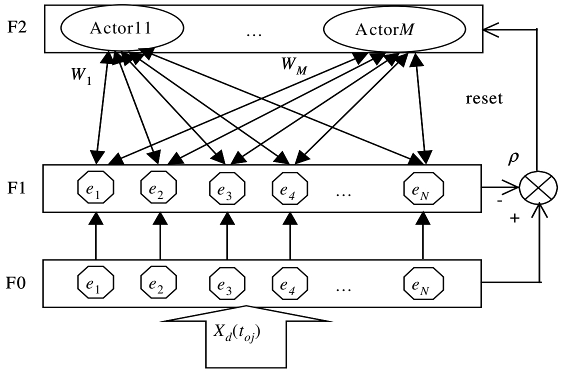

Fig. 5. ART network for self-organizational management

process of that the desired tasks $X_d(t_{oj})$, j=1…K, are posted by initiators and assigned to the participators iteratively, the weights will be adjusted based on a competitive principle such that the most competent candidate for a given task are selected as the winner. This competitive process is then conducted by examining the degree of match between the layer F0 and the Layer F1 of the winning candidate. If the degree of match between the demand of a task and the advertised capability is higher than a vigilance level ρ, a partner which fits is found and the task can be assigned to it for execution. Depending on performances of the execution, its weights can then be adjusted towards or backwards from the requirements for conducting the task. Therefore, it is a learning process based on the experiences of an actor. More experienced actors will be given more chance for similar jobs. After repetitive learning, the self-organizational ART network becomes a view of an initiator about the community i.e. which actor is competent for what kind of task.

The partner seeking process can be shown as below:

1) For each desired subtask t_{oj} from an initiator, the initiator posts the requirement vector $X_d(t_{oj})$ to the input layer of the ART network.
2) The comparison layer F1 attempts to classify it into one of participators based on its similarity to the advertised capability of each participator. A choice input of layer F2 is calculated for each participator a_j by bottom-up weight W_j:

$$T_j = \frac{\left|X_d \wedge W_j\right|}{\alpha + \left|W_j\right|} \tag{7}$$

   where α is a positive real-valued number called the choice parameter, $X_d \wedge W_j$ is a vector that its i^{th} component is equal to the minimum of X_{di} and W_{ji} and $|\bullet|$ is the norm of an vector, which is defined to be the sum of its components.
3) The actor with maximum choice input in the layer F2 will be selected as candidate for the task $X_d(t_{oj})$ (winner-takes-all competition):

$$T_l = \max\{T_i \mid i=1,\ldots,M\} \tag{8}$$

4) The ability of the winner is sent back to the layer F1 by its up-down weights. The examination of similarity between the winner's ability and the required ability is conducted by a vigilance criterion of:

$$\frac{|X_d \wedge W_I|}{|X_d|} \geq \rho \tag{9}$$

where $\rho \in [0,1]$ is the vigilance parameter given by the initiator.

There are the following possibilities after examination:

(1) If the criterion (9) is satisfied, the participator a_I may be competent enough for the task from the judgment of the initiator. Then go to step 5).

(2) Otherwise, the candidate a_I is reset and the next maximum participator is selected as a new candidate for the task. Then, repeat step 4 for vigilance examination.

(3) If no participators in the community C can pass the vigilance threshold, no competent candidate exists and a new actor should be recruited into the community, such as to recruit a new employee, to found a new department for the given task.

5) The initiator sends a contract to the winner a_I by the contract-net protocol[11], receives the bid and evaluates the bid from a_I. If the details can meet the requirement, awards the contract to a_I, otherwise resets a_I and selects the next maximum participator as candidate and go back to step 4).

6) Go back to step 1) for partner seeking of the next subtask $X_d(t_{oj})$, j=1…M, until all subtasks are assigned.

7) When any participator a_i completes its task and sends back results to the initiator, initiator evaluates the quality of the completed task in the interval, $\beta \in [-1,1]$, from "-1" very poor to "+1" perfect. Then, the advertised weight W_i will be modified by

$$W_i^{new} = (1-\eta)W_i^{old} + \eta\beta(X_d \wedge W_i^{old}) \tag{10}$$

where $\eta \in [0,1]$ is the learning rate of fuzzy ART. A bigger η corresponds to a faster learning.

This is a learning process with self-organizational ability for multi-agent systems. It goes through a process from matchmaking to quality feedback. More qualified participators are given more chance to get a contract. Based on the quality of the execution, equation (10) updates the advertised ability of a participator. Better performance (bigger β>0) will make the weight closer to what it has done and give this participator more chance to be a winner for the forthcoming similar jobs. On other hand, poorer performance (smaller β<0) will result from the weight being far away from the given task described by X_d and may lose similar task assignment in the next round of competition. The learning rate is selected based on a tradeoff between new ability learning and previous experience forgetting. With a bigger η, the actor is more likely to do the job that it just did well. Usually, a small company or a person is likely in this case. With a smaller η, the change of the actor's expertise is slower, for instance, the capability centre of a bigger company is usually changed little but the company can be competent in a wider field around the centre, which can be implemented by setting a lower vigilance ρ. If no actor is competent at a task, a new actor called uncommitted node in ART network should be generated in step 4) and becomes the winner automatically. Therefore, the aforementioned organization of a community can begin from inception without a priori knowledge and grows up from its experience step-by-step. In addition, through cooperation between initiator and participator, their descriptions about the demand and actor ability, using the concepts defined in the community, will become closer although their understanding about the concepts of ontology might be different initially. Equation (10) always lets the advertisement of the participator adapt to the requirement of an initiator who can provide tasks to participators. This is a learning process such that the gap between participator and initiator becomes smaller and smaller and each participator expects to win the next round competition.

4 Semantic Relation of a Community

In the last section, a full space of ontology clustering for partnership seeking is proposed. The task assignment depends on the similarity between the demands of an initiator and the advertisement of a participator described in the space of ontology entities. Usually, this is a high-dimensional space. For high-dimensional clustering, most algorithms can not work efficiently because of the sparsity of data[12]. In a community, the subtasks decomposed by initiators have a degree of randomness and cannot be predicted exactly. Therefore, the participators can only advertise their ability by a general means and from their own understanding about terminologies in the ontology. It is impossible to require both initiator and participator to use the same or similar description for their advertisements. Usually, for the advertised vectors, there might be a few dimensions on which the points are far from one another even though the essence is very close. This is because the components of the $(e_1, e_2, ..., e_N)$ space are highly correlated by the semantic relationship but the similarity comparison conducted in the last section thinks them irrelevant. Take an example, someone advertising himself that he can do the work of "software development" but at the same time having not exactly said that he can program using "C++", might fail to get a work of "C++ programming" by the scheme of the last section. In fact, "software" and "C++" have a tight semantic relation. In this section, a fuzzy inference scheme is proposed considering semantic relationship inside the ontology. Then, the self-organizational network proposed in the last section can be used for partnership seeking in the sense of fuzzy matchmaking.

The concepts $(e_1, e_2, ..., e_N)$ used by an initiator to announce their demand is not nonfuzzy; any concept implies some aspects of other concepts, which can be defined by a grade of membership proposed by Zadeh in his fuzzy set theory[13]. The ontology Ω_C of a community C defines the semantic relation between concepts $(e_1, e_2, ..., e_N)$ and can be used to determine the grade of membership. Now, the problem becomes a fuzzy inference from the initiator side to participator side such that a fuzzy matchmaking can be conducted on the layer F1 of Fig.5 and participators with relevant ability can be considered during competition. A fuzzy inference block can be added between layer F0 and layer F1 instead of direct connection in Fig.5. The conclusion of the fuzzy inference will replace X_d for similarity examination in equations (7) and (9).

Suppose the set of $\{e_1, e_2, ..., e_N\}$ forms a *universe of discourse* in the community. Any announcement t_{oj} of an initiator is a linguistic variable, such as "design *driver program* using *Java*". Then, the corresponding feature vector $X_d(t_{oj})=[x_{d1}, x_{d2},\ldots,x_{dN}]^T$ in (5) is a fuzzy variable representing the subtask t_{oj} on the initiator side, where x_{di} , i=1…N, is a grade of membership corresponding to the i^{th} variable in the universe. Then, fuzzy demand $X_d(t_{oj})$ on the initiator side should be transferred to a fuzzy variable $X(t_{oj})$ on the participator side, which considers fuzzy relationship between concepts($e_1\ldots e_N$):

$$X_d(t_{oj}) \wedge (I \Rightarrow P) \Rightarrow X(t_{oj}) \tag{11}$$

where $(I \Rightarrow P) \in R^{N\times N}$ is a fuzzy relation between the initiator side and participator side. It reflects the relationship between entities such that similar concepts can be considered during matchmaking even though they are not explicitly declared in the demands of the initiator.

If the relation $R=(I \Rightarrow P)$ is known, for any given fuzzy variable $X_d(t_{oj})$, it is straightforward to get the fuzzy variable $X(t_{oj})$ based on fuzzy inference. The relation R that reflects correlation between concepts can be obtained based on an OIL description of the global ontology Ω_C , e.g., of Fig. 1, because both fuzzy variables of the initiator and participator are defined by the elements of the global ontology Ω_C or by the set of the universe $\{e_1, e_2, ..., e_N\}$.

In the ontology Ω_C, distances between each pair of concepts (e_i, e_j) can be calculated as stated in the section 2, such as by using *visual distance*[9].Thus, a concept distance matrix can be generated as

$$D = \begin{bmatrix} 0 & d(1,2) & \cdots & d(1,N) \\ d(2,1) & 0 & \cdots & d(2,N) \\ & & \cdots & \\ d(N,1) & d(N,2) & \cdots & 0 \end{bmatrix} \tag{12}$$

where each component of $d(i,j)$ is a semantic distance between concept e_i and concept e_j.

Then, the relation $(I \Rightarrow P)$ can be calculated accordingly based on the distance matrix:

$$(I \Rightarrow P) = \begin{bmatrix} 1 & r(1,2) & \cdots & r(1,N) \\ r(2,1) & 1 & \cdots & r(2,N) \\ & & \cdots & \\ r(N,1) & r(N,2) & \cdots & r(N,N) \end{bmatrix} \tag{13}$$

where $r(i,j)=e^{-\alpha d(i,j)}$ and α is a steepness measure.

Equation (13) reflects the relationship between concepts that are defined in the global ontology of Ω_C.

Then, for any linguistic demand t_{oj}, the fuzzy inference can be conducted based on equation (11):

$$X(t_{oj}) = X_d(t_{oj}) \vee . \wedge (I \Rightarrow P)$$
$$= \begin{bmatrix} x_{d1} & x_{d2} & \cdots & x_{dN} \end{bmatrix} \vee . \wedge \begin{bmatrix} 1 & r(1,2) & \cdots & r(1,N) \\ r(2,1) & 1 & \cdots & r(2,N) \\ & & \cdots & \\ r(N,1) & r(N,2) & \cdots & 1 \end{bmatrix} \quad (14)$$
$$= \begin{bmatrix} x_1 & x_2 & \cdots & x_N \end{bmatrix}$$

where $\vee.\wedge$ is an inner product of fuzzy relation, such as max-min composition in [13]:

$$x_i = Max(\min(x_{d1}, r(1,i)), \min(x_{d2}, r(2,i)), \cdots, \min(x_{dN}, r(N,i))) \quad (15)$$

In the fuzzy inference (14), the input is a fuzzy set representing the demand of an initiator, the output X is also a fuzzy set reflecting which concepts might be required by the task where the semantic correlation between concepts has been considered.

Using the fuzzy set $X(t_{oj})$ instead of $X_d(t_{oj})$ in equation (7), (9), and (10), the ART network proposed in section 3 can realize fuzzy matchmaking during a process of self-organization.

5 Simulation Results

The proposed management network can give each contract initiator a view of actors participating in the contract competition and it can evolve gradually based on the experience of partnership execution. On the basis of this network, the community can be organized in a self-organizational and adaptive way. In order to verify the proposed scheme, the example shown in Fig. 1 is taken as background to the simulation, which is a community for electrical device development. Then, the concept distance matrix (12) of this community can be obtained by the definition of *visual distance*. The corresponding fuzzy relation between concepts can then be obtained based on (13) with α=1, thereafter.

Initially, suppose there are two participators, Actor 1 and Actor 2, in the community, where Actor 1 is a work team for hardware development work and advertises itself by Fig.6, Actor 2 is a work team for software development work with its advertisement of Fig.7.

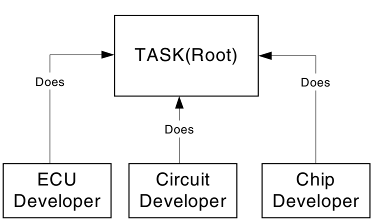

Fig. 6. Advertisement of Actor1

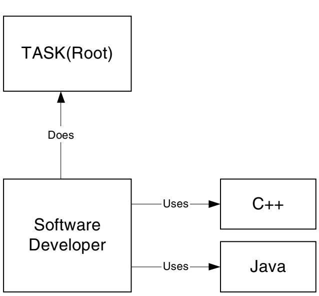

Fig. 7. Advertisement of Actor2

Accordingly, the feature vectors of both participators can be obtained by means of (3) and (4) with α=-4/MAX(Dis):

$$V_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & e^{-\alpha*1} & e^{-\alpha*1} & e^{-\alpha*1} \end{bmatrix}^T$$

$$V_2 = \begin{bmatrix} 0 & e^{-\alpha*1} & 0 & 0 & 0 & 0 & e^{-\alpha*2} & e^{-\alpha*2} & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$$

Suppose that a contract initiator from the automobile industry is seeking partners in the community for a telematic control unit (TCU) development. The TCU is a router which connects internal control area network (CAN) with external WLAN (Wireless Local Area Network, for example IEEE 802.11) and wireless WAN (Wide Area Network), such as the GPRS (General Packet Radio Service) or UMTS (Universal Mobile Telecommunication System), as well as with wireless radio networks such as bluetooth. The initiator has decomposed the development task into 6 subtasks with the feature vectors of:

Task 1: "C++ software work for Bluetooth application"

$$V(T_1) = \begin{bmatrix} 0 & 0 & 0 & e^{-\alpha*1} & 0 & 0 & e^{-\alpha*2} & 0 & 0 & e^{-\alpha*2} & 0 & 0 & 0 & 0 \end{bmatrix}^T$$

Task 2: "TCU programming for integration of Bluetooth, CAN, IEEE.802.11b"

$$V(T_2) = \begin{bmatrix} 0 & 0 & 0 & e^{-\alpha*1} & 0 & e^{-\alpha*2} & 0 & 0 & e^{-\alpha*3} & e^{-\alpha*3} & e^{-\alpha*3} & 0 & 0 & 0 \end{bmatrix}^T$$

Task 3: "Bluetooth communication chip design"

$$V(T_3) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & e^{-\alpha*2} & 0 & 0 & e^{-\alpha*1} & 0 \end{bmatrix}^T$$

Task 4: "Bluetooth board design based on the designed chip"

$$V(T_4) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & e^{-\alpha*2} & 0 & 0 & e^{-\alpha*1} & e^{-\alpha*1} \end{bmatrix}^T$$

Task 5: "TCU board design for connecting CAN, Bluetooth, and IEEE 802.11b"

$$V(T_5) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & e^{-\alpha*2} & e^{-\alpha*2} & e^{-\alpha*2} & 0 & e^{-\alpha*1} & e^{-\alpha*1} \end{bmatrix}^T$$

Task 6: "Integration of hardware work with software work"

$$V(T_6) = \begin{bmatrix} e^{-\alpha*1} & e^{-\alpha*2} & e^{-\alpha*2} & 0 & e^{-\alpha*3} & e^{-\alpha*3} & 0 & 0 & e^{-\alpha*4} & e^{-\alpha*4} & e^{-\alpha*4} & 0 & 0 & 0 \end{bmatrix}^T$$

From the subtask announcements, we can find that T1 and T2 are software related programming work, T3, T4, and T5 are hardware related development work, T6 is integration work and requires both hardware and software knowledge.

Firstly, management networks are established for Actor1 and Actor 2 based on their initial announcement of V_1 and V_2. Both static and dynamic performances will be investigated by the following simulations:

1) Static competition

The subtasks from T_1 to T_6 are advertised in the community and are taken as inputs to the ART network. The subtasks can then be assigned to the best matching actors based on winner-takes-all competition principle. This is a static matchmaking process and the process is called unsupervised in ART. However, the initiator can control this matching process by adjusting vigilance. A higher vigilance means a stricter matching condition, a lower vigilance gives a looser matching condition. The following table shows the competition result for the given tasks with different vigilances:

Table 1. Competition results under different vigilance (where * means a new actor should be recruited into the community for the given task)

	T1	T2	T3	T4	T5	T6
$0<\rho<0.87$	Actor2	Actor2	Actor1	Actor1	Actor1	Actor2
$0.87\leq\rho<0.92$	Actor2	Actor2	Actor1	Actor1	Actor1	*
$0.92\leq\rho<0.95$	Actor2	*	Actor1	Actor1	Actor1	*
$0.95\leq\rho<0.96$	Actor2	*	*	*	*	*
$0.96\leq\rho<1$	*	*	*	*	*	*

It shows that, for a lower vigilance, $0<\rho<0.87$, the proposed network can map the hardware related tasks to the Actor1 and the software related tasks to the Actor2 automatically. Because of lower vigilance, Actor2 also gets the bid of T6 that is an integration work combining software and hardware development. On increasing the vigilance to 0.87, this integration task T6 cannot be assigned to any existed actors and a new actor should be recruited specifically for this kind of job. Further increasing vigilance necessitates that more specialized actors should be included into the community for the given tasks. Increasing vigilance means increasing precision but decreasing intelligence from the principle of IPDI [14]. In fact, each initiator can define a dynamic vigilance for partnership creation. It reflects the confidence of an initiator in the intelligence of specific participators. Participators with higher confidence can be given a lower vigilance or vice versa. This dynamic vigilance can also be adjusted through a learning process to form an adaptive vigilance based on the initiator's experience or other actor's evaluation.

2) Dynamic evolution:

This simulation aims to investigate the evolution ability of the ART network in a virtual enterprise. In a dynamic environment, participators with adaptability have to evolve and change themselves according to the demands from initiators for the purpose of winning the forthcoming contract competition. Therefore, competition can stimulate enterprise development. In the ART network, more competent participators are given more chance to win a contract and the winners are given a chance to learn the demands.

In order to examine the dynamic performance of the proposed network, we need to model the ability of each participator. Suppose the model of each participator can be expressed by a Gaussian function, which acts as a measurement of the actor's expertise and will feedback to the initiator as β in learning law (10) for weight updating:

$$\beta = 2e^{-\frac{\|X-W_i\|^2}{2\sigma_i^2}} - 1 \tag{16}$$

where X is the desired task from an initiator, W_i is the advertised weight of the actor i, σ_i is an accepted field of the participator i, $\beta\in(-1,1]$ reflects the capability of the actor i, from "-1" very poor to "+1" very competent, for a given task X. Consequently, the actor i can perform a task X better when X is closer to W_i, the centre of its expertise. The accepted field σ_i reflects adaptability of an actor for a task with deviation from the actor's expertise centre. A bigger σ_i with wider range of acceptable tasks Now, suppose, in the community, there are intensive demands on actors with both hardware and software knowledge. The demands are expressed by the following feature vector series:

$$V(t_i) = V(T_6) + n(-0.1,0.1) \quad , i=1\ldots N, \tag{17}$$

where $n(-0.1, 0.1) \in R^{14\times1}$ is a vector with elements of uniform distributed noise between [-0.1,0.1].

Let the vigilance ρ=0.8 and suppose the learning rate η=0.2 and each actor has the same accepted field σ_i=1, which implies a poor adaptability in fact. Advertise the new tasks (17) in the community, which require both hardware and software expertise.

Fig. 8 illustrates the evolution process of the ART network, where d-axis is the Euclidian distance between the feature vectors of initiators and the weights of participators, i-axis indicates subtasks sequence. As shown in Fig.8, due to the lower vigilance(0.8), Actor2 got the first 3 tasks. Modeled by (16) with σ=1, each actor is able to complete the tasks satisfactorily only when they are close enough to its expertise centre. Due to the large difference, Actor2 lost the competition with Actor1 in the 4th round. However, the Actor1 has poor adaptability (σ=1) too and it lost the contract in the 7th competition. A new actor, Actor3, had to be recruited into the community for the integration work of hardware and software development. This is an example of equal capability with σ=1 for all actors.

Now, suppose Actor 1 is more competitive with σ=2 but all other actors' accepted fields are kept to be σ=1. As shown in Fig. 9, at the first 2 bids, Actor2 got the contract. Due to its poor performance, it lost the 3rd contract and cannot got back again because Actor1 with σ=2 is more competitive than it. After Actor1 got the bid, the distance between its weights and the desired tasks is decreased when the tasks in (17) come to the community sequentially. It implies that the expertise of Actor 1 was changing from "hardware development" to "hardware and software integration" gradually because of continuous requirements from initiators.

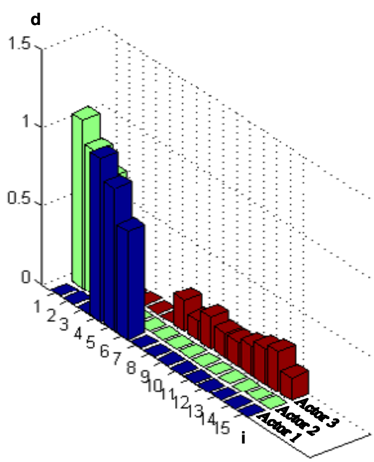

Fig. 8. Euclidian distances between demands and weights (σ=1)

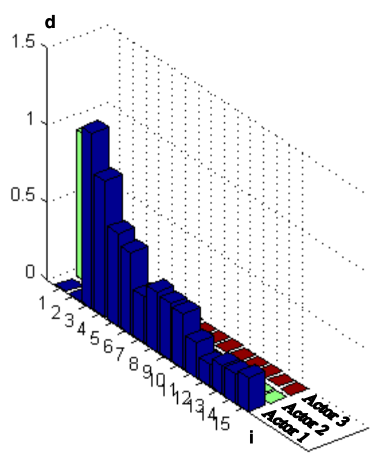

Fig. 9. Euclidian distances between demands and weights (σ_1=2)

6 Conclusions

This paper presented a self-organizational management network for the organization of enterprise partnership. The connections between different actors who are seeking for partnership are adjusted based on fuzzy adaptive resonance theory so that the management network can exhibit unsupervised learning ability, adaptive ability, competitive ability, and self-organizational ability. These abilities usually exist in human society. Semantic difference is used to quantify distance between demand and provision; at the same time fuzzy inference is used to solve ambiguous expression from both sides. Working in this way Virtual Enterprises can evolve dynamically and force to improve product quality of each actor and organizational performance of partnerships.

References

1. Jiang P., Mair Q.: An Actor-oriented Approach to Distributed Product Management Systems. COMPSAC 2002, Oxford, August 2002
2. Sycara K., Klusch M., Widoff S., Lu J.: Dynamic Service Matchmaking among Agents in Open Information Environments. ACM SIGMOD Record 28(1) (1999) pp. 47–53
3. Jeng J.J., Cheng B.H.C.: Specification Matching for Software Reuse. ACM SIGSOFT Software Engineering Notes 20 (1995)
4. Choi Y.S., Yoo S.I.: Multi-Agent Learning Approach to WWW Information Retrieval Using Neural Network. Proceedings of the 4th international conference on Intelligent user interfaces, December, 1998
5. Carpenter G.A., Grossberg S., Rosen D.B.: Fuzzy ART: Fast Stable Learning and Categorization of Analog Pattern by an Adaptive Resonance System. Neural Networks 4(6) (1991) 759–771
6. Horrocks D. F., Broekstra J., Decker S., Erdmann M., Goble C., van Harmelen F., Klein M., Staab S., Studer R., Motta E.: The Ontology Inference Layer OIL. http://www.ontoknowledge.org/oil/TR/oil.long.html
7. Williams J., Steele N.: Difference, Distance and Similarity as a Basis for Fuzzy Decision Support Based on Prototypical Decision Classes. Fuzzy Sets and Systems, in Press.
8. Fankhauser P., Kracker M., Neuhold E.J.: Semantic vs. Structural Resemblance of Classes. Special SIGMOD RECORD Issue on Semantic Issues in Multidatabase Systems. 20(4) (1991) ACM Press
9. Akoka J., Wattiau I. C.: Entity-relationship and Object-oriented Model Automatic Clustering. Data & Knowledge Engineering. 20 (1996) 87–117
10. Miao C.Y., Goh A., Miao Y., Yang Z.H.: Agent That Models, Reasons and Makes Decisions. Knowledge-based Systems. 15 (2002) 302–211
11. Smith R. G.: The Contract-Net Protocol: High-Level Communication and Control in a Distributed Problem Solver. IEEE Transactions on Computers. 29(12) (1980) 1104–1113
12. Aggarwal C. C.: A Human-Computer Cooperative System for Effective High Dimensional Clustering. Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, August 2001
13. Zadeh L.A.: Fuzzy sets. Information and Control. 8 (1965) 338–353
14. Saridis G.N.: Analytical Formulation of The Principle of Increasing Precision with Decreasing Intelligence for Intelligent Machines. Automatica, 25(3) (1989) 461–467