HCI Model with Learning Mechanism for Cooperative Design in Pervasive Computing Environment

Hong Liu[1,2], *Bin Hu*[1,2,3], *Philip Moore*[3]
[1]*School of Information Science and Engineering, Shandong Normal University, China*
[2]*Shandong Provincial Key Laboratory for Novel Distributed Computer Software Technology, China*
[3]*School of Information Science and Engineering, Lanzhou University, China*
lhsdcn@126.com, bh@lzu.edu.cn, ptmbcv@gmail.com

Abstract

This paper presents a human-computer interaction model with a three layers learning mechanism in a pervasive environment. We begin with a discussion around a number of important issues related to human-computer interaction followed by a description of the architecture for a multi-agent cooperative design system for pervasive computing environment. We present our proposed three-layer HCI model and introduce the group formation algorithm, which is predicated on a dynamic sharing niche technology. Finally, we explore the cooperative reinforcement learning and fusion algorithms; the paper closes with concluding observations and a summary of the principal work and contributions of this paper.

Keywords: HCI, Pervasive environment, Niche technology, Cooperative learning.

1 Introduction

Emerging pervasive computing technologies transform the way we live today by embedding computation in our surrounding environments. The evolving pervasive computing paradigm is characterized by an abundance of networked mobile and embedded computing devices that individuals and groups use across a variety of tasks and places; this paradigm is in contrast to more traditional interactions which are typically between an individual and a single device. Important trends in collaborative applications, human-computer interaction, mobile devices, and knowledge management are reflected in this change.

Pervasive computing promises to dramatically increase the number of networked devices in our everyday environment with device-to-device connectivity and connections between devices and individuals. In the evolving pervasive computing paradigm, a broad and diverse range of environments and settings including: homes, workplaces, medical settings, and vehicles will contain sensors, programmable controllers and other hardware that is networked together to provide a wide variety of information and services. The value of such services will be greatly enhanced if users can personalize the parameters and tailor their computing environment by composing services to meet their particular requirements.

Many of these computing devices will have an influence on, and control of, personal and domestic aspects of life. Based on the general usage of such devices which will be mainly characterized by non-technical users [1] such users, while perceiving the state of their world, will not necessarily understand or be aware of the underlying digital or physical causal factors that lead to changes in that state. This perception is typical of a pervasive computing paradigm. Thus, the typical non-technical user will not perceive the sensing, information processing, decisions, or actuation instructions undertaken by pervasive systems, although expert users may have some technical understanding of these processes [2].

The field of Human-Computer Interaction has a strong tradition of highlighting the need for designs to include appropriate visualization, visibility, and feedback to the users; accordingly these new environments may be challenging for designers. The current research project presented in this paper is a multidisciplinary approach which is designed to address this problem by bringing together, and applying, research in the fields of Human-Computer Interaction (HCI) as discussed in [3-4] and [5]. The problem and the challenge raises significant issues related to how the interfaces used in pervasive environment will fulfill the functional, social, cultural and even cross-cultural interaction gaps between users and computers [6-7]. All these questions are intrinsically linked with the challenges in the interfacing arising from the gaps that include: conceptual, modeling, design and application functions.

As users, we play different roles as individuals, members of a group, community, and society; our interactions taking place at the intersections and boundaries of the functional, social and cultural domains of our interac¬tions. In this scenario, we encounter not just the complexity of cross-functional and cross-social interaction but also of cross-cultural communication [8]. Therefore, the issue here is how to find a commonality of cultural experiences while recognizing cultural differences. That is to say how do we pool our shared cultural experiences (share

*Corresponding author: Bin Hu; E-mail: bh@lzu.edu.cn

DOI: 10.6138/JIT.2015.16.2.20131123

our pooled experiences) while recognizing our cultural differences and sustaining our cultural identities.

Recent years have seen a shift in perception of the nature of HCI and interactive systems [9]. The field of HCI has moved from being a relatively minor compo¬nent of software engineering to being the focus of attention for researchers from a variety of disciplines, including psychology and social science. Studies and investigations from these perspectives have led to a gradual evolution in our conception of the interface and of computer-based work in general. As a result, HCI has increasingly come to concern itself not just with the mechanism of the interface, but with a range of related issues concerning the context in which interactive systems are used [10-12].

This paper presents research on human-computer interaction model with learning mechanism for cooperative design in a pervasive environment. It is organized as follows. Section 2 begins with a description of the architecture for a multi-agent cooperative design system in a pervasive computing environment. This is followed in Section 3 by an introduction for the HCI model with the three layers. Section 4 investigates the group formation algorithm based on dynamic sharing niche technology. Section 5 further explores the cooperative reinforcement learning and fusion algorithms. Lastly, Section 6 summarizes the main contributions of this paper and discusses directions for future research.

2 The Architecture of A Multi-Agent System in Pervasive Computing Environment

The goal of pervasive computing is to create human-centered computing environment which provides computation and communication functions anytime and anywhere. The challenge lies in creating a human-centric computational model predicated on a foundation which supports mobile and multi-channel interaction. This section initially analyzes the nature of the interactions between the designers and agents; and then presents the architecture of a design agent for pervasive computing environment.

2.1 Multi-Agent Cooperative Design System

The general architecture of a multi-agent cooperative design system is organized as a population of asynchronous semi-autonomous agents for integrating design and engineering tools and human specialists in an open environment; the architecture is modeled in Figure. Agents in the same design team interact through the local area network while the agents in the different groups interact through the Internet and mobile devices to communicate, exchange of design data and knowledge.

The various agents accomplish the same goal but in a different manner. Because, in design, there is no single or clear-cut answer, different agents working on the same design problem can generate completely different solutions. By having agents with different abilities contributing to design process, the process gains robustness and variety in solving various conceptual design problems [13].

A multi-agent cooperative design system uses a hybrid federal architecture with three layers as shown in Figure 1. Essentially, HCI model is conceived as a centralized structure. The layer structure is as follows: the first layer is management layer, the second layer is tool layer, and the third layer is the design layer which adopts a federal dynamic alliance structure. It is formed dynamically according to design tasks and changed dynamically with the progress of design tasks. Each federal alliance structure adopts a distributed structure in which every design agent is on an equal footing. The task allocation, resource sharing and conflict coordination among design agents is completed by the tool agents on the second layer.

The Multi-agent collaborative design system for pervasive computing environments has: (1) the characteristics of general collaborative design system, and (2) also considers the interaction problems related mobile devices and multimodal information integration. The three layers structure based on the previous study document in [14], where the agents are increased in the second layer as shown in Figure 2.

In our HCI model the code conversion agent contains an HTML-WML transcoder, a graphic converter for GIF/JPG/BMP to WBMP, and a simulation of WAP browser. The HTML-WML transcoder can directly translate HTML pages into accord with WML standard syntax WML pages. Additionally, it can also translate more than one WML pages into its WAP mobile phone to browse and add an intelligent link function. The HTML-WML transcoder can also translate WML language into HTML language, and may be used to validate the WML page or use the Web browser to browse the WAP site. The graphics converter converts the GIF/JPG/BMP image into WBMP image. The transformed WBMP images can be reduced to fit the small screen display. Simulation of WAP browser can browse the WML pages with WML page displayed on the mobile phone interface simulation.

The synchronization service agent provides the content on mobile devices and the server of the corresponding content synchronization capability. Through the process of synchronization, handheld equipment change and changes on the server are transmitted to the other side. The notification service agent automatically notifies the user when a certain event occurs. The database service agent supports the synchronization between the database

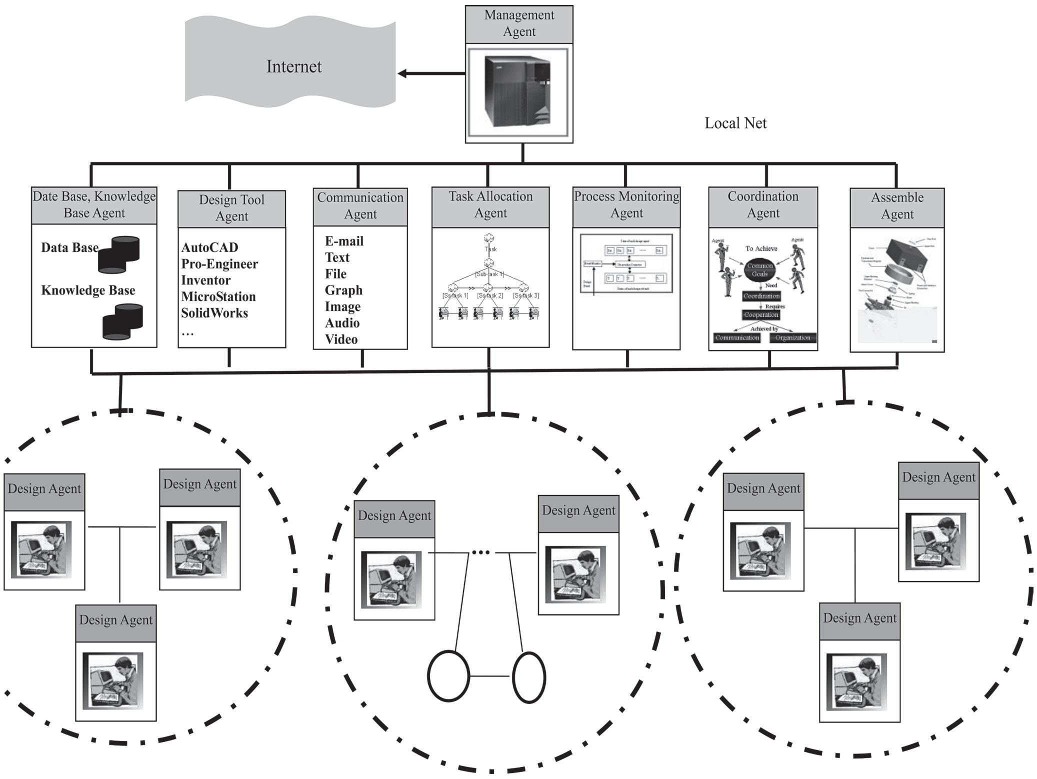

Figure 1 The Hierarchical Federal Architecture of a Multi-Agent Cooperative Design System

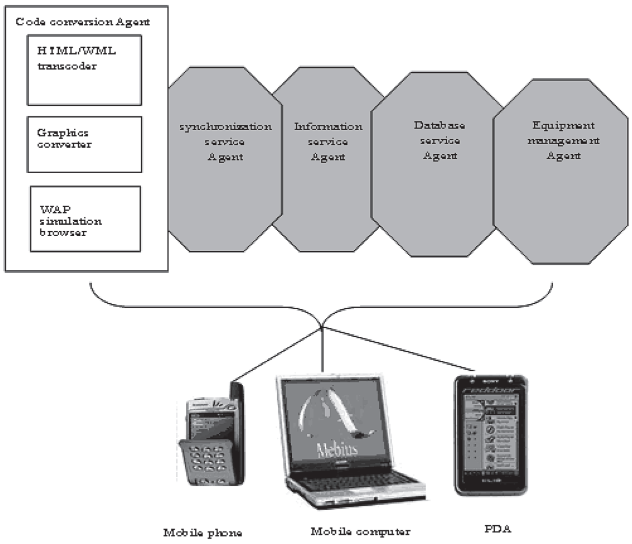

Figure 2 The Agents Are Increased in the Second Layer

and server database on mobile devices. The equipment management agent helps manage the mobile device software; it can be used to distribute new software package, update existing software, or delete redundant software.

3 HCI Model in Multi-Agent Cooperative Design System

This section considers the state-of-the art as it relates to human interaction and learning.

3.1 Related Work on Human Interaction and Learning

Animals or humans interact in various ways [15]:

- Individuals independently obey certain rules such that a whole swarm of individuals exhibits a certain behavior (such as in the case of movement of bird flocks or fish schools).
- Individuals exchange information in often simple or some times complex ways in order to achieve a common goal (examples are pheromone trails of ants or bee dance languages).
- Individuals learn from each other by observing each other (for example: babies learn from their parents by mimicking their parents language or behavior).
- Individuals learn from each other by communicating their knowledge; for example: learning based on rules (such as children that learn from their teachers in school).
- Individuals learn from each other by exchange of experiences using explicit and tacit knowledge, i.e., meta-knowledge about rules (for example, in a team of more or less experienced engineers who develop a new product together).

A self-organized collaboration of individuals that mutually profit from an exchange of knowledge or meta-knowledge (experience) can certainly be seen as a higher-level form of collective intelligence or symbiotic intelligence. The advantages of such a collaboration of individuals may be manifold:

- Individuals maybe enabled more in a more timely way in response to certain critical situations.
- Individuals may even behave proactively, for example: before they are confronted with certain situations, they already possess the knowledge relating to the situation and ways to handle them.
- Teams of individuals maybe enabled to solve problems that they cannot solve by alone.

The social learning theory of Bandura [16] emphasizes the importance of observing and modeling the behaviors, attitudes, and emotional reactions of others. Social learning theory explains human behavior in terms of continuous reciprocal interaction between cognitive, behavioral, an environmental influences. The component processes underlying observational learning are: (1) Attention, including modeled events (distinctiveness, affective valence, complexity, prevalence, functional value) and observer characteristics (sensory capacities, arousal level, perceptual set, past reinforcement), (2) Retention, including symbolic coding, cognitive organization, symbolic rehearsal, motor rehearsal), (3) Motor Reproduction, including physical capabilities, self-observation of reproduction, accuracy of feedback, and (4) Motivation, including external, vicarious and self reinforcement.

The documented research results in the literature have laid a theoretical foundation for this research.

3.2 The HCI Model with the Three Layers

In our HCI model, there are three different learning approaches:

(1) Cooperative learning: which is based on the group learning; interestingly the result of group learning can be exchanged and shared among different group.

(2) Group learning: runs concurrently with the sharing of experiences and knowledge within a group.

(3) Individual learning: is the most important and essential learning approach to learning in nature. For its survival, an individual should continuously accumulate experiences and knowledge by learning and practice; this provides a basis for enhancing the ability of self-adaptation.

In our research, the HCI model with three layers should be created incorporating: a cooperative model, a group model and a user model. The three layers (sequentially from the bottom layer) are introduced in the following section.

3.2.1 User Model

The user model is a collection of users information. The interaction mode among the designers and agents become richer in a pervading environment. In addition to the traditional desktop computers and workstations, a broad and diverse range of fixed, mobile, and wearable devices such as wired devices, wireless devices, handheld devices, sensors and so on may be modeled in our HCI model.

The designers and these devices interact with an agent(s) via multi-channel interface. The interaction approach is shown in Figure 3.

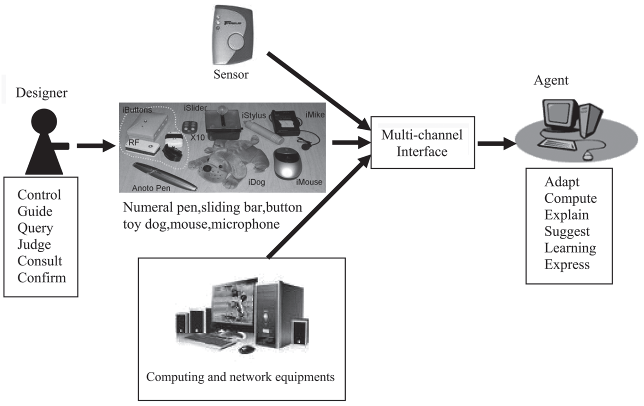

Figure 3 The Interact Manner between a Designer and an Agent in a Pervasive Environment

A pervasive environment encapsulates a broad and diverse range of context information; in actuality, if it can be captures, and digitized any information may be considered to be contextual information.

Contextual information generally falls into four categories. (1) environment context parameters including: location, speed, time and so on, (2) equipment context parameters including: equipment characteristics such as availability, network bandwidth, and screen sizes, (3) personal contextual information including: operational habits, preferences, constraints, and demanded tasks, and (4) social contexts such as: proximate information, group activity, and relationships with other agents and designers in the environment. These contexts are nit independent but are interdependent as they affect each other during design process. The context-awareness is essentially in a learning process [17]. The main base of learning is the accumulation of past contexts; namely context history. The learning process will be described at the following sections.

3.2.2 Group Model

The group model expresses the commonalities between individual users. The groups are founded by formation algorithm which divides the users into several groups according to their background knowledge. The system then adopts different strategies for different groups; this is designed to enable a decrease in the interaction between human and computer.

3.2.3 Cooperative Model

If the agent is the abstraction of the human individual, then multi-agent system is the abstraction of the human society. Analogies can be drawn however each agent member cannot (and does not need) to learn all the facts and parameters in a multi-agent system as multiple agents can share information and knowledge with each other, or learn from other agent.

For the purpose of the problem in this paper, we present a variant dynamic model, the *Dynamic Collaborative Learning Model* (DCLM). The DCLM is predicated on the following idea: it divides design agents with the same design background and similar design task into a class, and chooses a design agent with strong adaptive capacity in each class. By providing individual training to selected agents, new design knowledge obtained through with designers' interaction, such knowledge is then spread in a class of similar design agents. It can enable design knowledge sharing with commensurate reductions in learning time.

Our proposed model combines the advantages resulting from a dynamic niche sharing and cooperative learning strategy. We define the model to be dynamic in the following sense: (1) the population is dynamically divided into subpopulations; this dynamical process is activated by the niche center which is dynamic in itself, (2) the niches are dynamic and are determined by a niche center set consisting of present best solutions of the problem, and (3) the niche center set is itself dynamic being dependent upon the learning process while it determines the niches and learning processes vice versa.

Figure 4 shows four groups on the group learning layer and selected four agents from each group on the cooperative learning layer. This organization is dynamically changed with the development of learning process.

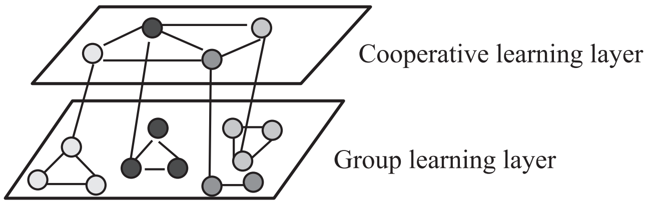

Figure 4 The Relationship between Cooperative Learning Layer and Group Learning Layer

4 The Dynamic Group Formation Algorithm Based on Dynamic Niche Technology

There are at least two fundamental properties shared by many practical MAS applications that require the agents to be adaptable and capable of learning how to improve their group formation strategies [18].

Initially, for the purpose of effectively coordinating in the same or similar kinds of environment, agents need to become effective at completing the same or similar set of repetitive tasks. Clearly, being able to learn from past experience and then improve in future coordination interactions would be very beneficial.

Second, in the most realistic MAS environments [including those where group formation naturally arises as a way of solving a distributed task or resource assignment problem] are characterized by a number of possible sources of uncertainty and noise. Once these sources are taken into account [and assuming agents would need to form group repeatedly] clearly each agent should be able to learn how to better identify which candidate groups with other agents that have a high chance of success, i.e., are most likely to succeed at completing future [unknown] tasks.

In nature, the species are genetically isolated, meaning that individuals only mate with other members of their species. Mating restrictions are enforced simply by evolving the species in separate populations; the species interacting with one another within a shared domain having a cooperative relationship.

The Dynamic Niche Sharing Model has been proposed by Miller and Shaw [19]. It attempts to identify peaks of the forming niches and uses dynamically identified peaks to classify all individuals as either belonging to one of the dynamic niches, or else belonging to the non-peak category as shown in Figure 5.

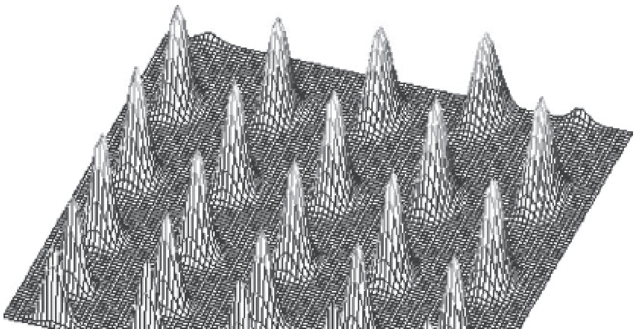

Figure 5 The Peaks Formed by Niche Model

In this model, the dynamic peak is the individual with the highest fitness among the niche, i.e., the center of the niche. Dynamic niche algorithm attempts to recognize the formation of niche peak, and use these peaks for all individual classification.

The shared fitness of an individual i is given by $f_{sh,i} = \frac{f_i}{m_i}$, where f_i is the raw fitness of the individual, and m_i is the niche count, which defines the amount of overlap (sharing) of the individual i with the rest of the population. The niche count is calculated by summing a sharing function over all members of the population: $m_i = \sum_{j=1}^{N} sh(d_{i,j})$. The distance $d_{i,j}$ represents the distance between individual i and individual j in the distance is within a fixed radius σ_{sh}, it greater similarity (decreasing $d_{i,j}$). Otherwise it returns 0. In essence, each member of the population is considered to be the center of a niche, and its shared fitness value is reduced for every other member of a population whose distance is less than a niche radius σ_{sh} from that individual.

$$sh(d) = \begin{cases} 1 - \left(\dfrac{d_{i,j}}{\sigma_{sh}} \right)^{\alpha_{sh}} & if \;\; d_{i,j} < \sigma_{sh} \\ 0 & otherwise \end{cases} \tag{1}$$

The dynamic niche sharing model, which succeeded the fitness sharing method, aims to dynamically classify the individuals as members of one of the niches. The dynamic niche count is:

$$m_i^{dyn} = \begin{cases} n_j & if \;\; individual \; is \; within \; dynamic \; niche \; j \\ m_i & otherwise \end{cases} \tag{2}$$

Where n_i is the size of the jth dynamic niche, and m_i is the standard niche count. The shared fitness is then defined respectively:

$$f_i^{dyn} = \frac{f_i}{m_i^{dyn}} \tag{3}$$

The dynamic collaborative niche model is suitable for locating niches automatically. However, since optimal solutions are not equally distributed in general, this dynamic model can effectively reduce computing time by eliminating those areas where there are no optimal solutions. By this technology, we can find all possible optimal solutions in a relatively small computing complexity.

Populations are dynamically divided into subpopulations via niche center. The dynamic scheme should be determined beforehand using certain measurement; for example, fitness value, distances between niches are possible measurements to determine the dynamic scheme. Each subpopulation will generate its niches during design process which will be added to the niche center set.

For complete collaboration, one natural scheme is migration. When new niches are generated, a portion of the best niches is selected in one subpopulation, and this portion migrates to other subpopulations by a predefined probability. In this mode, subpopulations will exchange design knowledge.

In this algorithm, the number of design agents N and niche radius σ_{sh} should be given beforehand. Initial fitness values are task-oriented and given by management agent according to the historical records of design agents when they performed the similar task.

The dynamic group formation algorithm based on dynamic niche technology:

Step 1: Sort design agents in decrease order according to their fitness;

Step 2: Produce the random integer k in [1, N] (initial niche number);

Step 3: The former k design agents are put into different niches respectively and become niche center. Ensure that all niche in the distance between the centers of greater than σ_{sh}. If can not satisfy this condition, then combined with two niches, k = k-1, k > 0;

Step 4: For other n-k design agents, calculates the distance between design agents with its current niche center. If the design agent and all niche centers' distance are greater than σ_{sh}, then a new niche is generated, and the design agent becomes the center of the new niche; otherwise the design agent is arranged to the nearest niche according to the principle of minimum distance;

Step 5: When all design agents have been assigned, the design agent in niche center that is the most adaptable design agent among the niche;

Step 6: Selected design agents are trained and obtained new design knowledge through with designers' interaction, then spread design knowledge in each niches;

Step 7: Count the fitness values of all design agents according to Equation (3). If it is not terminated by designer, then go to step 1.

The time complexity of algorithm is O(Nk).

5 The Cooperative Reinforcement Learning Algorithm

Because each agent knows has relatively little knowledge of other agents and given the dynamic nature of the environment, as a type of environment independent model of self learning, reinforcement learning is suitable for multiple agent system. The application of reinforcement learning in multi-agent systems has gained significant traction [20-22]

The reinforcement learning process is expressed by the Markov Decision Process (MDP) which is defined as a quadruple <S, A, R, P> where: S is the set of state, A is the action set, R is the reward function, and P is the state transition function.

The essence of the MDP is: the probability and reward value of the current state transferring to the next state is dependent only on the current state and action, which has no relationship with the historical state and historical action. Therefore, in a known state transition, the probability function P and the reward function R under the environment knowledge, dynamic planning techniques can be used to resolve the optimal strategy. While reinforcement learning focuses on the function of P and R under the uncertainty of unknown situation, reinforcement learning identifies how to convert an agent from a free state to the target state by a series of optimization operation.

The reward is obtained by performing an action, therefore, to maximize the expected reward, have to rely on the optimal action strategy to select and perform the action.

If the action strategies are expressed as function π: $S \times A \rightarrow [0,1]$, then $\pi(s, a)$ refers to the probability of selecting action a under the state s.

In the basic Q learning algorithm, using state-action pair put strategy π, expected return function and the state-action pair together. It states that for some strategy and execution state, the quality degree of an action, the standard is good or bad for the expected return function.

In the Q-Learning algorithm described, each agent receives a single reward from the environment at time t+1 after all actions have been processed at time t. This reduces the learning rate per agent to a maximum of one learning event per time unit. Thus, for the group of n Q-Learning agents in the simulation, the learning rate of the group is bounded by the linear function: $Lr = n$.

We can increase the learning rate of the group by allowing each individual agent to share their learning experiences with the rest of the group. If each of the n agents experiences a learning event at time t, this can be passed on to each of the other n-1 agents in the group at time t+1. The maximum learning rate for the group at time t+1 would then occur when each of the n agents experiences another learning event, as well as receiving and processing information from the other n-1 agent's learning events from time t. In such a case, the group learning rate is bounded by the function $Lr = n + n(n - 1) = n^2$.

This maximum learning rate assumes that the communication of the Q-Learning agents is global and not restricted by range. If range restrictions apply, then the above maximum rate is conditional upon the proximity of agents to each other during the simulation, and would be reduced accordingly.

The Q-Learning agents process their learning events through the rewards they receive from the environment and knowledge of the states and actions executed. Thus, communicating a reward to another agent is not enough, as it must be placed in context by the corresponding state-action pair that generated it, as well as the resultant state.

Thus, at each time-step, an agent can potentially increase its learning rate based on the group learning. This technique is applicable given that the group of agents is homogeneous (in the same niche), each having the same set of states and actions available.

The main framework of the algorithm is as shown in Figure 6 [23].

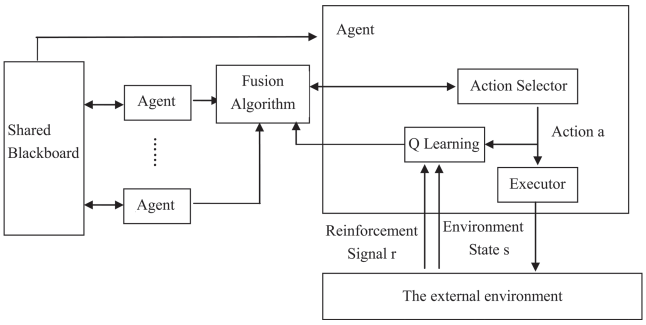

Figure 6 The Framework of Multi-Agent Cooperative Learning [23]

In which:

Action selector: according to the Q value and action selection strategy, select the action.

Q Learning: according to environment state s, action a and reinforcement signal r, by executing the Q-learning algorithm, to learn and adjust the action strategy.

Executor: perform the action given by the action selector and act on environment, make environment state s_t move to next state s_{t+1}.

Shared blackboard: after each F steps learning, each agent will publish their current accumulation of Q value to the blackboard, also get the Q values of other agents from the blackboard Q value, so as to realize information sharing.

Fusion algorithm: fuse strategies from the board and draw lessons from strategy that can get higher reward value. The action selector selects the action based on fused strategy.

Cooperative reinforcement learning algorithm:

Step 1: t = 0, Initialize $Q(s_i, a_i)$ of each agent;

Step 2: To perform standard Q -- learning algorithm for each agent:

(1) Observe the current state of the environment s_i;

(2) According to Equation (4) select an action and implementation;

$$p(a_i, s_t) = \frac{e^{Q(s_t,a_i)/T}}{\sum_{b \in A} e^{Q(s_t,b)/T}} \tag{4}$$

In which, $p(a_i, s_t)$ expresses the probability of selecting a_i in the state s_t, the larger $Q(s_i, a_i)$, the larger the probability of a_i is chosen; b is an action in the set of agent actions; T is the temperature parameter, approximate annealing method is used for T, i.e., to use a higher temperature at beginning, and decreased gradually with the learning process.

Observe the successor state s_{i+1} of the environment, and obtain reinforcement signal r_i from the environment;

According to Equation (5), update the corresponding $Q(s_i, a_i)$ for state-action pair (s, a);

$$Q_{i+1}\left(s_i, a\right) = (1-\alpha_t)Q_i(s_i, a) + \alpha[r_i + \gamma \max_{a^{'}} Q_t(s_{t+1}, \alpha^{'})] \tag{5}$$

Among them, α is learning rate; γ is discount parameter; $0 \leq \alpha \leq 1$, $0 \leq \gamma < 1$.

Step 3: If t can be divided by specified in advance F, then all the agents exchange and fuse strategies.

After each F steps learning, each agent will publish their current accumulation of Q value to the blackboard, also get the Q values of other agents from the blackboard Q value, so as to realize information sharing.

Step 4: t ← t+1;

Step 5: If the successor state meet the end condition, then end; else s ← s’, go to step 2. After the learning convergence, select the action by the greedy strategy in Equation (6):

$$a^*(s) = \arg\max_{b \in A} Q(s, b) \tag{6}$$

Fusion algorithm:

Step 1: Let every F steps is a period of learning. After each learning cycle, calculating the average value for all current Q values of agents.

$$\overline{Q}_{k+1}(s,a) = \frac{1}{N}\sum_{i=1}^{N} Q_{k+1}^{i}(s,a) \tag{7}$$

Step 2: Calculating:

$$\Delta = \overline{Q}_{k+1}(s,a) - \overline{Q}_k(s,a) \tag{8}$$

Step 3: Calculating:

$$Q_{k+1}(s,a) = \begin{cases} \max\{Q_{k+1}^{i}(s,a)\}, & \Delta > 0 \\ \min\begin{cases} \overline{Q}_{k+1}(s,a), & \Delta = 0 \\ Q_{k+1}^{i}(s,a), & \Delta < 0 \end{cases} \end{cases} \tag{9}$$

Step 4: For all agents, calculating:

$$Q_{k+1}^{i}(s,a) = Q_{k+1}(s,a), \text{ i} = 1, 2, ..., \text{N} \tag{10}$$

The fusion algorithm shows: when Q(s, a) increased (average value increasing), i.e., taking the action a in the state s may lead to a higher reward, all the other agents to learn this experience from the agent with the highest Q value; Conversely, when the Q(s, a) decreased (average value decreasing), that is taking the action a in the state s may bring lower rewards, all the other agents to learn this lesson from the agent with the lowest Q value. Exchanging and sharing strategy improves the learning speed and learning effect of agents.

In the algorithm, there are simultaneous learning and cooperation of multi-agents. On the one hand, through learning, an agent can predict the behaviors of other agents, update the shared information, and adjust the action in order to adapt to the changing environment; through cooperation, on the other hand, the agents can learn from each other, sharing their knowledge and experience, so as to accelerate the process of learning.

6 Conclusion

In this paper, we have considered a number of important design issues and challenges related to user-centered human-computer interaction. Based on the discussion we have presented an HCI model predicated on a three layers learning mechanism. The aim of the research is to support harmonious human-computer interaction in pervasive environment.

In summary, this paper makes the following contributions:

(1) It puts forward an HCI model incorporating a three layers learning mechanism for pervasive computing environment.

(2) It investigates the group formation algorithm based on dynamic sharing niche technology.

(3) It explores the cooperative reinforcement learning and fusion algorithms for supporting learning and cooperation of multi-agents.

While the research has resolved a number of important issues we consider that it is important to actually carry out a set of research projects that together follow the model to validate the posited approach; this forms the basis for our ongoing work.

Acknowledgements

This project is founded by the project of Taishan scholars and supported by National Natural Science Foundation of China (No. 61472232, No. 61272094).

References

[1] Tim Owen, Ian Wakeman, Bill Keller, Julic Weeds and David Weir, *Managing the Policies of Non-technical Users in a Dynamic World*, *Proc. IEEE 6th International Workshop on Policies for Distributed Systems and Networks*, Stockholm, Sweden, June, 2005, pp.251-254.

[2] Eleanor O'Neill, Owen Conlan and David Lewis, *Situation-Based Testing for Pervasive Computing Environments*, *Pervasive and Mobile Computing*, Vol.9, No.1, 2013, pp.76-97.

[3] Yueh-Min Huang and Tsung-Ho Liang, *A Technique for Tracking the Reading Rate to Identify the E-Book Reading Behaviors and Comprehension Outcomes of Elementary School Students*, *British Journal of Educational Technology*, 2014, doi:10.1111/bjet.12182.

[4] Mark Weiser, *Creating the Invisible Interface (Invited Talk)*, *Proc. the 7th Annual ACM Symposium on User Interface Software and Technology*, Marina del Rey, CA, November, 1994, doi:10.1145/192426.192428.

[5] Benjamin Weyers, Wolfram Luther and Nelson Baloian, *Interface Creation and Redesign Techniques in Collaborative Learning Scenarios*, *Future Generation Computer System*, Vol.27, No.1, 2011, pp.127-138.

[6] Hansu Gu, Haojie Hang, Qin Lv and Dirk Grunwald, *Fusing Text and Frienships for Location Inference in Online Social Networks*, *Proc. IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, Macau, China, December, 2012, pp.158-165.

[7] Silvia Schiaffino, Marcelo Armentano and Analía Amandi, *Building Respectful Interface Agents*, *International Journal of Human-Computer Studies*, Vol.68, No.4, 2010, pp.209-222.

[8] Karamjit Singh Gill, *Rethinking the Interaction Architecture*, in Satinder P. Gill (Ed.), *Cognition, Communication and Interaction*, Springer, London, 2008, pp.213-234.

[9] Michael A. Goodrich and Alan C. Schultz, *Human-Robot Interaction: A Survey*, *Foundations and Trends in Human-Computer Interaction*, Vol.1, No.3, 2007, pp.203-275.

[10] Bin Hu, Philip Moore and Hsiai-Hwa Chen, *A Semantic Context Model for Location Based Cooperative Mobile Computing*, *Proc. IEEE International Conference on Communications*, Glasgow, UK, June, 2007, pp.326-331.

[11] Philip Moore, Bin Hu and Jizheng Wan, *Smart-Context: A Context Ontology for Pervasive Cooperative Computing*, *The Computer Journal*, Vol.53, No.2, 2010, pp.191-207.

[12] Tsung-Ho Liang and Yueh-Min Huang, *An Investigation of Reading Rate Patterns and Retrieval Outcomes of Elementary School Students with E-Books*, *Educational Technology & Society*, Vol.17, No.1, 2014, pp.218-230.

[13] Hong Liu and Ming-Xi Tang, *Evolutionary Design in a Multi-agent Design Environment*, *Applied Soft Computing Journal*, Vol.6, No.2, 2006, pp.207-220.

[14] Hong Liu, *The Model of Multiagent Cooperative Design System for Pervasive Computing*, *Journal of Computer Aided Design and Computer Graphics*, Vol.19, No.6, 2007, pp.804-810.

[15] Dominik Fisch, Martin Jänicke, Edgar Kalkowski and Bernhard Sick, *Learning from Others: Exchange of Classification Rules in Intelligent Distributed Systems*, *Artificial Intelligence*, Vol.187-188, 2012, pp.90-114.

[16] Albert Bandura, *Social Learning Theory*, General Learning Press, New York, 1977.

[17] Hong Liu, *Context-Aware Agents in Cooperative Design Environment*, *Journal of Computer Applications in Technology*, Vol.39, No.4, 2010, pp.187-198.

[18] Predrag T. Tosic and Ricardo Vilalta, *A Unified Framework for Reinforcement Learning, Co-learning and Meta-learning How to Coordinate in Collaborative Multi-agent Systems*, *Procedia Computer Science*, Vol.1, No.1, 2010, pp.2217-2226.

[19] Brad L. Miller and Michael J. Shaw, *Genetic Algorithms with Dynamic Niche Sharing for Multimodal Function Optimization*, *Proc. the 1996 IEEE International Conference on Evolutionary Computation*, Nagoya, Japan, May, 1996, pp.786-791.

[20] Hansu Gu, Mike Gartrell, Liang Zhang, Qin Lv and Dirk Grunwald, *AnchorMF: Towards Effective Event Context Identification*, *Proc. the 22nd ACM International Conference on Information & Knowledge Management*, San Francisco, CA, October/November, 2013, pp.629-638.

[21] Shoma Tanabe and Naoki Masuda, *Evolution of Cooperation Facilitated by Reinforcement Learning with Adaptive Aspiration Levels*, *Journal of Theoretical Biology*, Vol.293, No.21, 2012, pp.151-160.

[22] Ching-Lung Chang and Siao-Ji Kang, *Using the Reinforcement Learning Scheme to the Priority-Based Routing and Call Admission Control in WDM Networks*, *Journal of Internet Technology*, Vol.13, No.5, 2012, pp.793-802.

[23] Li-Hua Xue, *The Research on the Cooperative Approach of Multi-agents*, Ph.D. Thesis, Changsha Polytechnic University, Changsha, China, 2008.

Biographies

Hong Liu is now a Professor of computer science at the School of Information Science and Engineering, Shandong Normal University. She received PhD degree from the Chinese Academy of Sciences in 1998. Her main research interests include computational intelligence and cooperative design.

Bin Hu is now a Professor at the School of Information Science and Engineering, Lanzhou University. He is also a Taishan Scholar in Shandong Normal University. He received PhD degree from the Chinese Academy of Sciences in 1998. His main research interests include pervasive aware computing and cooperative design.

Philip Moore is now a Professor at the School of Information Science and Engineering, Lanzhou University. He is also a guest professor in Shandong Normal University. He received PhD degree from Fukuoka Institute of Technology, Japan. His main research interests include intelligent context aware systems and data fusion with intelligent context processing.