

Delayed Consistency Model for Distributed Interactive Systems with Real-time Continuous Media

Xiao Qin

*School of Computing and Information Technology
Griffith University
Brisbane, Queensland 4111, Australia
xqin74@yahoo.com*

Abstract

The advanced multimedia and high-speed networks make distributed interactive systems more promising and practical. These systems are distributed systems, which allow many clients who are located in different locations to concurrently explore and interact with each other. This system can be built in both the local area network (LAN), and the wide area network (WAN), such as the Internet. Operations issued at one site are immediately executed at the local sites for a good response time, and are propagated to other sites. One of the challenging issues in this system is the consistency maintenance. Such issue in the discrete interactive media has been studied in many literatures. Unfortunately, the consistency maintenance scheme for discrete interactive media is not suitable for continuous media domain. This paper illustrates a consistency problem in continuous interactive media by a simple example. The absolute consistency model, a strong requirement, is suitable for LAN, but result in a bad responsiveness in WAN. To make the model more practical for WAN, we propose a new consistency model named delayed consistency model. In this model, if there is an operation for an object x is issued at site i , every site should be able to execute this operation at a specified time. With this approach, other sites are forced to update the state at a certain amount of time later than site i does. Thus, other sites will finally view the same state of x as that of site i . Our model is flexible, because it is unnecessary for all sites to have the identical delayed time. If the system is based on the real-time network, then another advantage of the model is providing a useful parameter for real-time network scheduling.

Keywords

Consistency maintenance, interactive object, continuous media, distributed systems

1. Introduction

With the increasing development of distributed

systems, multimedia and high speed networks [1], building distributed interactive systems becomes promising and practical. Using such systems, groups of geographically distributed users and services can share information that is created and updated dynamically. Examples of distributed interactive applications include distributed games, distributed simulation [5][6][7], distributed interactive learning [3] and distributed virtual environment [2][4], as well as CSCW applications such as collaborative editing systems [8][9][10][11]. In the distributed interactive systems, clients can extract the relevant objects by updating their states, or communicate real-time with other clients who also work in the same system.

Due to the fast development of the Internet computing, there are increasing number of distributed interactive systems developed based on the Internet. Distributed interactive system provides an infrastructure to build a large scaled environment for interactive activities by interconnecting many clients, in which people may cooperate with each other. This system has brought a new set of open issues and challenging problems to solve.

A well-designed distributed interactive system has four basic features. (1) The proper interpretation of the time, (2) Consideration of the transmission delays of the operations, (3) Execute the interactive operations in a correct causal order. (4) Real-time response, the users make their judgement and reaction based on the situation presented to them by human-computer interfaces. These features can be guaranteed by the consistency maintenance scheme, which is one of the most important issues in distributed systems.

In an interactive system with continuous media, the objects are moving around. Even without operations issue on them, the objects can move by themselves according to the rules, which have been specified by the design stage. It is a basic requirement that all sites should have the same view upon the object. Hence, when one site issues an operation to an object, this operation together with its time-stamp is propagated to other sites, other sites can update the state of the object according to the received operation.

In this paper, we focus on the consistency issue for the interactive continuous objects. The distributed interactive system studied in this paper has no centralized management, and each object in the system has the replicated copies at all sites. The network delay is unpredictable bound, therefore, this model can be applied for the Internet based interactive systems.

The rest of this paper is organized as follows. In Section 2, an overview of related work is presented. Section 3 gives a simple example to illustrate an inconsistency problem, which is named as delayed inconsistency. To keep the system delayed consistent, section 4 presents a consistency model for distributed continuous objects. Section 5 described the consistency maintenance for distributed interactive objects. Finally, we summarize the contributions of this paper and suggest the future research work.

2. Related Work

Consistency maintenance is one of the most significant challenges in designing and implementing the distributed interactive systems. Such issue in the discrete interactive media have been explored and discussed in many literatures. Real-time collaborative editing system is a good example of interactive system with discrete media [8][10][11][14][12].

In paper [8], a consistency model, with properties of convergence, causality preservation, and intention preservation, are proposed as a framework for consistency maintenance in real-time collaborative editing systems. The integrated set of schemes and algorithms, which support the proposed model, are presented in [8]. To study the consistency maintenance in real-time collaborative editing system, Li et al. studied how the user intentions might be impacted when the finite duration of drawing operations is considered [10]. The problem of maintaining consistency in the fact of high-latency communication network is also studied in [10]. Sun and Chen proposed a novel distributed multi-version approach to conflict resolution for real-time collaborative graphics editor [14][15].

Storm et al. proposed a model, in which several objects may be atomically updated, and these objects automatically maintain consistency with their replicas using an optimistic algorithm [16]. The algorithms are based on optimistic guess propagation principles, adapted for fast commit by using primary copy replication approach.

Unfortunately, the consistency maintenance schemes devised in these papers only concern about the discrete interactive systems. Unlike the object with discrete media, a continuous media object not only changes its state according to the operations issued on it, but also

continuously updates the state due to the passing of time as well. Thus, the object's state not only depends on what operation issued on it, but also depends on the time at which this operation is executed.

Recently, the continuous media has been employed in various kinds of distributed interactive systems, such as multi-user computer games [22], distributed virtual reality [17][18][20][24] and simulations [7][19]. Distributed virtual environment is a distributed system, which allows many clients who are located in different locations to concurrently explore and interact with each other in a high resolution, 3dimensional, graphical virtual environment [24]. Greenhalgh et al. studied a QoS architecture for collaborative virtual environments [18], and their work focus on the management of streamed video within shared virtual worlds. Choukair and Retaillieu explored a model of the Distributed Virtual Environment Collaboration Model (DVECOM), which aims to provide an end-user QoS support for distributed virtual reality applications [20]. Bieber and Siron proposed a security architecture, which aims to protect the intellectual property of firms participating to a distributed interactive simulation [7]. As the number of simulated entities grows, the number of messages that need to be sent per unit of time can grow to unmanageable numbers. To reducing the number of messages, Messina et. al presented a approach to keep track of what entities need to know about which other entities and only send information to the entities that need to know [19]. However, these works did not study the consistent control in the systems.

The consistency maintenance in the continuous media remains many open issues. Zhou et al. introduced a concept of time-space inconsistency for continuous distributed interactive applications [13], and three kinds of inconsistency problems were addressed in [13]. Our paper, on the other hand, addresses a problem that is not covered in [13]. Diot and Gautier described the design and implementation of a distributed multiplayer game on the Internet, in which the buckets synchronization mechanism is devised to guarantee the consistency of the game [22]. Mauve studied the important tradeoff relationship between the responsiveness of the medium and the appearance of short-term inconsistencies [21]. Paper [21] shows that the fidelity of the application can be significantly raised by voluntarily decreasing the responsiveness of the medium. This approach is named local lag, which means instead of immediately executing an operation issued by a local user, the operation is delayed for a certain amount of time before it is executed. This scheme is very similar as the buckets synchronization addressed by Diot and Gautier [22].

Our consistency model, unlike the model proposed in [21][22], shows that it is unnecessary for all the sites have the same state at the same time. Different sites could share the same view of the objects at the different time, which is

specified at the designed stage and predictable for user.

3. An Example of Inconsistency in Continuous Interactive Objects

Since the continuous interactive object may move by itself even without any operation issued on it, its state may continuously update not only according to the operations issued on it, but also because of the passage of time as well. In our model, it is assumed that the local clocks of all the sites are synchronized periodically. The clock synchronization can be guaranteed by Network time Protocol [23] or GPS clocks, and the clock deviation is less than 50ms. The work reported in [21] has the similar assumption.

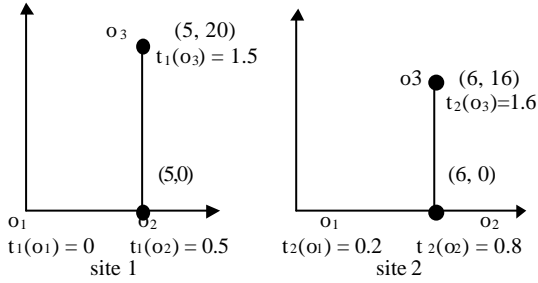


Figure 1 Example of an inconsistency problem in object with continuous media. Site 1 issues three operations at time 0, 0.5 and 1.5, respectively. The communication delay for operations are 0.2, 0.3 and 0.1. The final position at site 1 is (5, 20) and the position after executing o_3 at site 2 is (6, 16).

As shown in Figure 1, we illustrate our example in a 2-dimension environment. It is straightforward to extend this example into 3-D environment. Suppose the initial position of object x at site 1 and site 2 is the same, and set to be $(0, 0)$, and the initial velocity of x is $(0, 0)$. Site 1 issue three operations: operation 1 make x move, operation 2 changes the direction and accelerates x , and operation 3 stops x again. We calculate the final position of x at two sites as follow.

At time $t_1(o_1) = 0$, the operation o_1 from site 1 make x move constantly at the velocity $v = (10, 0)$. At time $t_1(o_2) = 0.5$, operation o_2 changes the direction of the object x , and also increases speed to $v' = (0, 20)$. Hence, object x changes the direct at the coordinate $(5, 0)$. At time $t_1(o_3) = 1.5$, x is stopped by o_3 , at the position of $(5, 20)$.

After site 1 executes each operation on object, it immediately sends the operation to site 2. After receiving the operation from site 1, site 2 executes it without any delay. Assume that the communication delays for o_1, o_2

and o_3 are $d_1 = 0.2, d_2 = 0.3$, and $d_3 = 0.1$, respectively. The execution time of three operations at site 2 is, $t_2(o_i) = t_1(o_i) + d_i$, where $i = 1, 2$ and 3. At site 2, x begins moving at time $t_2(o_1) = 0.2$, and execute the operation o_2 at $t_2(o_2) = 0.8$. Therefore, x changes the direct at position $(6, 0)$. Object x stops in response to o_3 at time $t_2(o_3) = 1.6$ with the final position $(6, 16)$. It is obvious that the position of x at site 1 is not consistent with that at site 2.

The reason for the inconsistency problem shown in this example is that the communication delays associated with three operations are different from each other. The unbounded network delay gives rise to this kind of inconsistency.

4. Consistency Model

In this section, we describe the system model. The purpose of this work is to find out a scheme to solve the inconsistency problem illustrated in above section. A distributed interactive system can be developed in a local area network, private network or even in the Internet, to present the system generally, we use a graph to describe the underlying network setting. The system is denoted as a graph $G = (V, E)$, where $V = \{v_1, v_2, \dots, v_n\}$ represents the set of sites connected by the network, and the set of weighted and directed edges E represents communication delay between two sites. $(v_i, v_j) \in E$ indicates a network channel between two sites. d_i^j represents the upper bound communication delay for a message transmitted via this channel. In the following definitions, we assume time t to be the wall-clock time.

4.1 Absolute Consistency

Definition 1. Given an object x , the state of x is defined by its current coordinate and its current property, such as the directional velocity, size, colour, etc. At the wall-clock time t , let $s_x^i(t)$ denotes the state of x at site I at time t , where $1 \leq i \leq n$.

Definition 2. Given an operation o , $v(o)$ denotes the site at which o is generated, $g_t(o)$ represents the time when o is generated, and $e(o)$ denotes the object on which o is operated. $a_i(o)$ and $c_i(o)$ represent the arrival time and the execution time of o at v_i . It is obvious that $g_t(o) \leq v(o) = i$, and $a_i(o) \leq v(o) = i$.

Definition 3. Given two operations o_i and o_j , o_i is causal order preceding o_j , i.e. $o_i \rightarrow o_j$, iff: (1) $v(o_i) = v(o_j) = k$, and $g_k(o_i) < g_k(o_j)$; (2) $v(o_i) \neq v(o_j)$, $a_k(o_i) < g_k(o_j)$, where $k = v(o_i)$; (3) There exists an operation o_k , such that $o_i \rightarrow o_k, o_k \rightarrow o_j$.

Definition 4. Given an object x , the object is *absolute consistent* at time any t , iff $\forall i, j \in N, i \neq j: s_x^i(t) = s_x^j(t)$. The system is absolute consistent, iff $\forall x, t > 0: x$ is absolute consistent.

In most cases, especially for the system that is built on the Internet, absolute consistent is not possible to satisfy, due to the transmission delay. This fact is described in theorem 1 and 2 as follows, these theorems all based on the assumption that the system clocks are synchronized.

Theorem 1. Assume that system is initialized to be absolute consistent. The system is absolute consistent, if and only if any operation at all sites must execute at the same time.

Proof. (1) Firstly, it is straightforward to prove that any operation at all sites must execute at the same time \rightarrow the system is absolute consistent. Since the initial state of the system is consistent and each operation is executed at all the sites at the same time, the state at each sites update the state at the same time. Therefore, system is absolute consistent is proved.

(2) Secondly, we prove the system is absolute consistent \rightarrow any operation at all sites must execute at the same time. This can be proved by contradiction. Given an operation o issued on the object x , where $v(o) = i$. Let t be $g_i(o)$. We suppose $\forall t' < t: \exists j \in N, i \neq j: s_x^i(t') = s_x^j(t')$, thus, the system is absolute consistent at any time before t . If site i executes o at t_1 whereas site j execute o at $t_2 > t_1$, then it is clear that, $s_x^i(t_1) \neq s_x^j(t_2)$, thus, the system is not absolute consistent at time t_1 . Therefore, the operations must be executed at all sites at the same time.

Theorem 2 Given an operation o issued on the object x , where $v(o) = i$, if o executes at site i immediately after it is issued, then system is not absolute consistent at time $g_i(o)$.

Proof. Let t be $g_i(o)$. We suppose $\exists e > 0: \forall j \in N, i \neq j: s_x^i(t-e) = s_x^j(t-e)$, thus, the system is absolute consistent at time $t - e$. (1) If at time t , one site j ($j \neq i$) updates the state of x with operation $o' \neq o$, then $\exists j \in N, i \neq j: s_x^i(t) \neq s_x^j(t)$. We prove that system is not absolute consistent at time $g_i(o)$. (2) Secondly, we prove the case that at time t , all other site j ($j \neq i$) does not generate operation on x . Since o update the state of x at time t , we have $s_x^i(t) \neq s_x^j(t-e)$. Since it takes time $d > 0$ to propagate operation from i to other site, the state of x at site j ($j \neq i$) remains unchanged at time t , in other words, $\forall j \in N, i \neq j: s_x^j(t) = s_x^j(t-e)$. We get $\forall j \in N, i \neq j: s_x^i(t) = s_x^j(t) = s_x^j(t-e)$, thus the system is not absolute consistent at time $g_i(o)$ is proved. Based on (1) and (2), theorem 2 is proved.

Theorem 2 indicates that, to keep the system absolute consistent, the operation generated at the local site could not execute immediately after it is issued. On the contrary, the operation must delay for a certain of time before it is

executed. To solve the absolute consistency problem, we have the following theorem 3.

Theorem 3. Given an operation o issued on the object x , to keep the system absolute consistent, the execution time of o at site i is equal or greater than $g_i(o) + \text{MAX}_{j=1, j \neq i}^n(d_i^j)$, where $i = v(o)$, and $d_i^j(o)$ is the transmission delay for the operation o sent from site i to site j .

Proof. We prove this theorem by contradiction. Let $d_i^k = \text{MAX}_{j=1, j \neq i}^n(d_i^j)$ and suppose site i does not delay for a certain of time d_i^k , thus, site i executes operation at time $g_i(o) + d$, where $d < d_i^k$. Since site k receives and executes the operation at $g_i(o) + d_i^k$, site i and site k execute the o at two different time. According to theorem 1, the system is not absolute consistent. Hence, the execution time of o at site i is equal or greater than $g_i(o) + \text{MAX}_{j=1, j \neq i}^n(d_i^j)$ is proved.

4.2 Delayed Consistency of Objects

From theorem 3, we observe that if the delay time for each local operation is long enough, the number of absolute inconsistencies can be sharply decreased. This scheme can also be found in the work of Mauave [21] and Cristian [25]. Unfortunately, employing a long delay for local operation results in a long response time, this may not be accepted by the users. Hence, it is very important to determine the tradeoff between consistency and a good response time.

For the system built on the Internet, it is very difficult to guarantee the absolute consistent. The reason is that Internet gives rise to the unpredictable bounds and large variations on message transmission delay, and the delay is normally significant noticeable.

Under such circumstance, a good compromise is hard to decide, in other words, we either have a bad responsiveness with few inconsistencies or have a great number of inconsistencies with a short response time. To solve this problem, we propose the concept of the late consistency. In our model, we assume that only one site may own a given object at a given period of time, and a site can only issue the operation on the object it owns. For example, in the distributed car racing, player A at site 1 only can steer his car, and he can observe but not steer other players car. If we regard A's car as a object, then the owner of this car is player A.

Definition 5. Given an object x , the ownership of x reflects its owner at a give time t . It is denoted as $w_x(t)$. For the simplicity, we assume that the ownerships for the objects are not transferred, therefore, we have $\forall t \in \mathcal{T}: w_x(t) = w_x$.

From the definition of w_x , we get:

$$v(o) = i \rightarrow e(o) = x \wedge w_x = i. \quad (1)$$

Definition 6 Given an object x , the object is *delayed consistent* at time $t \geq 0$, iff $\forall i, j \in \mathcal{N}, i \neq j: w_x = i \otimes$

$s_x^i(t) = s_x^j(t + \mathbf{d}_i^j)$. \mathbf{d}_i^j is the phase difference between two sites.

From the above definition, we know that if an object is owned by site i with the state s at time t , such object at other sites will have the same states. It is noted that state s appears at other sites at a certain amount of time later than t . For example, from time step 1 to 20, the state of the object x at site 1 is $\{s_x^1(1), s_x^1(2), \dots, s_x^1(20)\}$, and x has the same sequence of the state at site 2 starting from time 5 to 24. In other words, we have $\{s_x^2(5) = s_x^1(1), s_x^2(6) = s_x^1(2), \dots, s_x^2(24) = s_x^1(20)\}$. The phase difference between site 1 and 2 is 4, thus, $\mathbf{d}_1^2 = 4$.

To keep the delayed consistency for an object, the local operation o on x is executed immediately after it is issued by site $v(o)$, and o is propagated to other sites. As the other sites received the operation o , they execute the operation after delaying it for a certain amount of time. Hence, at site $j \neq v(o) = i$, o is executed at time $g_i(o) + \mathbf{d}_i^j$. That is

$$c_j(o) = g_i(o) + \mathbf{d}_i^j \quad (2)$$

This approach ensures that the object x is delayed consistent. This is described in the property as below.

Property 1. Give an object x , $\forall o, \exists i, j \in \mathcal{N}, i \neq j: e(o) = x \wedge v(o) = i \wedge c_j(o) = g_i(o) + \mathbf{d}_i^j \wedge (\forall t < g_i(o): x \text{ is delay consistent}) \rightarrow x$ is delayed consistent.

Assume all objects in site i are delayed consistent, then given two operations o_1 and o_2 issued at site i , we get $c_j(o_2) - c_j(o_1) = g_i(o_2) - g_i(o_1)$, where $j \neq i$. This feature is formally described in theorem 4.

Theorem 4. Given two operations issued on site i , we have $v(o_1) = v(o_2) = i \otimes \forall j \in \mathcal{N}, j \neq i: c_j(o_2) - c_j(o_1) = g_i(o_2) - g_i(o_1)$.

Proof. This theorem can be proved straightforward from the approach describe above. From equation (2), we have

(a) $\forall j \in \mathcal{N}, j \neq i: c_j(o_1) = g_i(o_1) + \mathbf{d}_i^j$ and (b) $\forall j \in \mathcal{N},$

$j \neq i: c_j(o_2) = g_i(o_2) + \mathbf{d}_i^j$. Equation (b) is subtracted by equation (a) we have $\forall j \in \mathcal{N}, j \neq i: c_j(o_2) - c_j(o_1) = g_i(o_2) - g_i(o_1)$.

Unlike the model proposed by Mauave [21] and Cristian [25], this approach does not force all other sites to execute o at the same time, thus, different sites have difference parameter. The parameter \mathbf{d} is specified according to users requirement, this makes the system design more flexible. As shown in Fig. 2, delay time from site 2 to site 1 is 9, whereas delay time from site 2 to site 3 is 5.

Let d_i^j be the upper bound on the communication delay

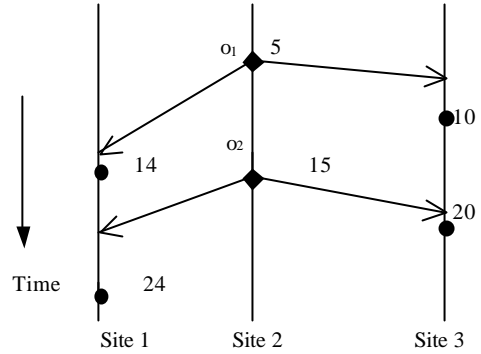


Figure 2. Delay time from site 2 to site 1 is 9, whereas the delay time from site 2 to site 3 is only 5.

from site i to site j , suppose the parameter d_i^j can be obtained during the design stage, then it ensures that all operations arrive before d_i^j . Hence, the specified delay time \mathbf{d}_i^j must be equal or greater than d_i^j as formally described in theorem 5.

Theorem 5. Give an object x , and $i = v(o)$, if the object is *delayed consistent*, then $\forall j \in \mathcal{N}, i \neq j: \mathbf{d}_i^j \geq d_i^j$.

Proof. Assume at $t = g_i(o)$, site i update the state of x by executing the operation o . We suppose the transmission delay equals to the upper bound d_i^j , and site j changes the state in response to o at time $t + d_i^j$. Thus, $s_x^j(t) = s_x^j(t + \mathbf{d}_i^j)$. In this case, we have $\mathbf{d}_i^j = d_i^j$. If site j delays for a certain amount of time τ after receiving o at time $t + d_i^j$, then we get $s_x^j(t) = s_x^j(t + \mathbf{d}_i^j + \tau)$. Therefore, $\mathbf{d}_i^j = d_i^j + \tau \geq d_i^j$. Theorem 5 is proved.

4.3 Delayed Consistency of Sites

Given two site i and j ($i \neq j$), delayed consistency model and scheme for objects make site i view the same states of the objects belongs to site j at a certain amount of time later. Given another site k ($k \neq i$ and $k \neq j$), site i also views the same states of the objects belongs to site k at a later time. We are interested in the views of objects belongs to sites j and k from site i . Even though all the objects belongs to sites j and k are delayed consistent, some interesting problems still may occur. We will show the problem by another simple scenario that is depicted in figure 3. For simplicity, we only show the example in 1-dimension and it is easy to extend this example to a more complex one into 2-D and 3-D environment.

In an Internet based racing game, player A and C at site

1 and 3 drives car A and C, which are denoted as object x, and y, respectively. Player B at site 2 is an observer who is interested in this racing game. We assume that $\delta_1^2 = 5, \delta_3^2 = 10$. Operation o_1 and o_3 are issued on object x at site 1, whereas o_2 and o_4 are issued on object y at site 2. From fig. 3, we find that both object x and y are delayed consistency.

The initial positions of x and y are 0. At time step 1, operations o_2 make object x move at the speed of 10, and at time step 2, object y also starts moving with a speed of 10 by operation o_2 . Therefore, the distance between x and y is 20, thus x is 20 ahead of y when y begins moving. However, from the observer's point of view, when both x and y are moving, the distance between x and y is 70 instead of 20. Because at site 2, x starts moving at 5, whereas y begins moving at 12. Hence, when y begins moving, x is 70 ahead of y.

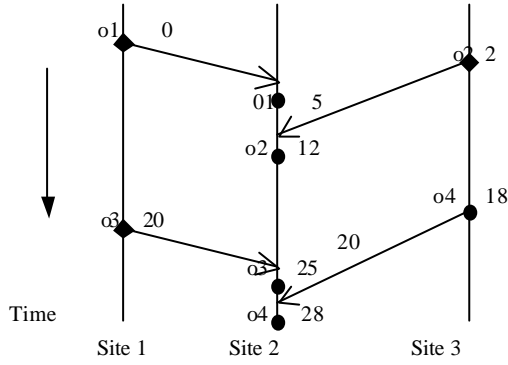


Figure 3. $d_1^2 = 5, d_3^2 = 10$, operation o_1 and o_3 are issued on object x at site 1, whereas o_2 and o_4 are issued on y at site 2.

At time step 18, operation o_4 stops object y, so the final position of y is 160. While y stops at 18, object x continues moving for 2 time steps, and x is stopped by o_3 at time 20 with the final position 200. Again, from site 2's view, it is quite different. Player B at site 2 finds that x stops 3 time steps earlier than y, thus, when x stops, object y is still keep on moving.

Because in this scenario, the specified delays δ_1^2 and δ_3^2 are different. The states of the objects in site 1 and site 3 have been delayed for different amount of time before they are viewed by site 2. The approach to solve this problem is simple, let δ_1^2 and δ_3^2 be the same value.

To study this inconsistent problem in a general way, we introduce the definition of the delay consistency for a site. Let X be the set of object, it has n subsets, denotes as $X = X_1 \dot{\cup} X_2 \dot{\cup} \dots \dot{\cup} X_n$, where $X_i = \{x \in X | w_x = i\}$. It is clear that, $\forall i, j \in \{1, 2, \dots, n\}, i \neq j: X_i \cap X_j = \emptyset$

Definition 7. The state of the site i at time t is represented as, $S^i(t) = S_I^i(t) \cup S_E^i(t)$, where $S_I^i(t)$ is composed by the states of the objects owned by site i , whereas $S_E^i(t)$ contains the states of the objects that belong to other site. Thus, $S_I^i(t) = \{s_x^i(t), \text{ where } x \in X_i\}$ and $S_E^i(t) = \{s_x^i(t), \text{ where } x \notin X_i\}$

Definition 8 Given a site i , it is delayed consistent, iff $S_E^i(t) = S_I^1(t - d^i) \cup \dots \cup S_I^{i-1}(t - d^i) \cup S_I^{i+1}(t - d^i) \dots \cup S_I^n(t - d^i) = \bigcup_{j=1, j \neq i}^n S_I^j(t - d^i)$. The system is consistent, iff $\forall i \in \{1, 2, \dots, n\}$: all site i are delayed consistent.

As defined in def. 8, if the system is consistent, it must have the following property.

Property 2. For any site i in the system, it satisfies: $\forall i \in \{1, 2, \dots, n\}, \forall j \in \{1, 2, \dots, n\}, i \neq j: d_j^i = d^i$.

Parameter d^i is derived from $d_j^i (i \in \{1, 2, \dots, n\}, j \in \{1, 2, \dots, n\}, i \neq j)$. d^i is given and proven in the theorem 6 as below.

Theorem 6. If the system is consistent, then, $\forall i \in \{1, 2, \dots, n\}: d^i \geq \text{MAX}_{1 \leq j \leq n, j \neq i} d_j^i$.

Proof. Given a site $i (i \in \{1, 2, \dots, n\})$, we assume $d_k^i = \text{MAX}_{1 \leq j \leq n, j \neq i} d_j^i$. According to the theorem 5, we have $d_k^i \geq d_k^i$. Since the system is consistent and based on the property 2, we get $d_k^i = d^i$. Therefore, we obtain $d^i \geq \text{MAX}_{1 \leq j \leq n, j \neq i} d_j^i$. Thus, we get $\forall i \in \{1, 2, \dots, n\}: d^i \geq \text{MAX}_{1 \leq j \leq n, j \neq i} d_j^i$.

To keep the delayed consistency for a given site i , our approach is very simple: The operation o is delayed before it is executed by site j . This approach ensures that the execution times of o at site i is d^i later than that at site j . Thus,

$$c_i(o) = g_j(o) + d^i \quad (3)$$

Assume a site i is delayed consistent, and given two operations o_1 and o_2 issued at site j and k , respectively. We have $c_i(o_2) - c_i(o_1) = g_k(o_2) - g_j(o_1)$, where $j \neq i$ and $k \neq i$. Theorem 7 explains this feature formally.

Theorem 7. If a site i is delayed consistent, we have $\forall i \in \{1, 2, \dots, n\}, \forall k \in \{1, 2, \dots, n\}, k \neq i: v(o_1) = j \cdot \dot{\cup} v(o_2) = k \cdot \text{@} c_i(o_2) - c_i(o_1) = g_k(o_2) - g_j(o_1)$.

Proof. From equation (3), we have (a) $\forall i \in \{1, 2, \dots, n\}, \forall j \in \{1, 2, \dots, n\}, j \neq i: v(o_1) = j \cdot \text{@} c_i(o_1) = g_j(o_1) + d^i$ and (b) $\forall i \in \{1, 2, \dots, n\}, \forall k \in \{1, 2, \dots, n\}, k \neq i: v(o_2) =$

$k \oplus c_i(o_2) = g_k(o_2) + d^i$. Equation (b) is subtracted by equation (a) we have $v(o_2) = k \oplus c_i(o_2) - c_i(o_1) = g_k(o_2) - g_j(o_1)$.

6. Conclusion

We address the consistency issue in the distributed interactive system with continuous media. We illustrate a consistency problem by a simple example. The main reason for this consistency problem is that the objects can move and update the automatically even without manipulation of the users. Hence, the state of an object not only depends on the operation, but also depends on when the operation is executed on the object as well. It is noted that the objects with discrete media do not have such consistent problem. To solve this problem, we propose the delayed consistency model for the interactive object. We found another inconsistent problem, which is not covered by the delayed consistency model of object. After employing the delayed consistency model and the new approach, this problem is also solved properly. Our model is better than the models discussed in the literatures [21]. The advantage of our model is multiple described as follows.

(1) In this model, it is unnecessary for all the sites to have the same state at the same time. Different sites could share the same view of the objects at the different time, which is specified at the designed stage.

(2) The system can tolerant long communication delay time by having a good predicability. The local site views the state of the other sites, and has the knowledge about what is the time associated with the states of other sites.

(3) In our model, there is a specified delay time for each object. If the system is built on the real-time communication, the specified delay time is a good parameter for the scheduling mechanism of the real-time network schedules real-time messages by using this parameter. During the real-time scheduling, the message with shorter delay time will be assigned a higher priority [26].

This work represents our first and preliminary attempt to address the consistency issue in a complicated problem. There are a lot of open issues remained to be studied. Our future work in this field could include, (1) Our model is assumed that the clocks in the system are synchronized by Network time Protocol or GPS clocks, and there is no clock drift. Based on the model proposed in this paper, we study a more sophisticated model that takes the clock synchronization and the clock drift into account. (2) Another assumption of our work is that the ownership of each object is not changeable. In our future model, we will relax this assumption by letting the ownership be transferred from one site to another site, this will give rise

to new consistency problem, and we will devise the new approach to solve the problem. (3) The model devised in this paper assumes that there is an upper bound on the communication delay for each network channel, and operations will be guaranteed to arrive at the destination within the specified delay time. In our future model, we will assume that there is no such upper bound for message communication. In case that the operations arrive at the destination later than the specified delay time, the consistency can not be kept. Hence, a scheme must be designed to recover the state of the objects in this case.

References

- [1] D.L. Mills. Internet Time Synchronization: the network Time Protocol. IEEE Transactions on Communications, Vol.39, No.10, pp.1482-1493, Oct. 1991.
- [2] J.C. Oliveira, S. Shirmohammadi and N.D. Georganas. Collaborative Virtual Environment Standards: A Performance Evaluation. In *Proceedings of the 3rd International Workshop on Distributed Interactive Simulation and Real-Time Applications*. College Park, Maryland, 23 - 24 October, 1999.
- [3] J. Chen, C. Dede, X. Fu, and Y. Yang. Distributed Interactive Learning Environment. In *Proceedings of the 3rd International Workshop on Distributed Interactive Simulation and Real-Time Applications*. College Park, Maryland, 23 - 24 October, 1999.
- [4] X. Shen, R. Hage, and N. Georganas. Agent-Aided Collaborative Virtual Environments over HLA/RTI. In *Proceedings of the 3rd International Workshop on Distributed Interactive Simulation and Real-Time Applications*. College Park, Maryland, 23 - 24 October, 1999.
- [5] G. Fortino and L. Nigro. Simulation of Multimedia Systems Based on Actors and QoSsynchronizers. In *Proceedings of the 3rd International Workshop on Distributed Interactive Simulation and Real-Time Applications*. College Park, Maryland, 23 - 24 October, 1999.
- [6] J. Pullen. Reliable Multicast Network Transport for Distributed Virtual Simulation. In *Proceedings of the 3rd International Workshop on Distributed Interactive Simulation and Real-Time Applications*. College Park, Maryland, 23 - 24 October, 1999.
- [7] P. Bieber and P. Siron. Design and Implementation of a Distributed Interactive Simulation Security Architecture. In *Proceedings of the 3rd International Workshop on Distributed Interactive Simulation and Real-Time Applications*. College Park, Maryland, 23 - 24 October, 1999.
- [8] C. Sun, X. Jia, Y. Zhang, Y. Yang, and D. Chen. Achieving convergence, causality-preservation, and intention preservation in real-time cooperative editing

- systems. *ACM Transactions on Computer-Human Interaction*, 5(1):63-108, March 1998.
- [9] C. Sun. Undo any operation at any time in group editors. *In Proceedings of ACM 2000 Conference on Computer Supported Cooperative Work*, December, Philadelphia, Pennsylvania, USA, 2000.
- [10] Du Li, Limin Zhou, and Richard R. Muntz. A new paradigm of user intention preservation in realtime collaborative editing systems. *In Proceedings of the Seventh International Conference on Parallel and Distributed Systems*, Iwate, Japan, July, 2000.
- [11] C. Sun and C.A. Ellis. Operational transformation in real-time group editors: Issues, algorithms, and achievements. *In Proceedings of ACM Conference on Computer Supported Cooperative Work*, Seattle, USA, November 1998.
- [12] C.A. Ellis and S.J. Gibbs. Concurrency control in groupware systems. *In Proceedings of the ACM SIGMOD conference on Managements of Data*. Portland, OR, USA, pp. 399-407, 1989.
- [13] S. Zhou, W. Cai, F.B. Lee and S.J. Turner. Consistency in Distributed Interactive Application. *In Proceedings of the 2001 European Simulation Interoperability Workshop*, London UK, June 2001.
- [14] D. Chen and C. Sun. A distributed algorithm for graphic objects replication in real-time group editors. *In Proceedings of the international ACM SIGGROUP conference on Supporting group work*, Phoenix, AZ USA, pp.121-130, 1999.
- [15] C. Sun and D. Chen. A multi-version approach to conflict resolution in distributed groupware systems. *In Proceedings of the 20th IEEE International conference on Distributed Computing Systems*, pp.316-325, April, 2000.
- [16] R. Strom, G. Banavar, K. Miller, A. Prakash, and M. Ward. Concurrency Control and View Notification Algorithms for Collaborative Replicated Objects. *IEEE Transactions on Computers*, Vol. 47, No. 4, April 1998.
- [17] E. Frécon and M. Stenius. Dive: A scalable network architecture for distributed virtual environments. *Distributed Systems Engineering Journal*, Vol.5 No.3, pp.91-100, 1998.
- [18] C. Greenhalgh, S. Benford, and G. Reynard. A QoS architecture for collaborative virtual environments. *In Proceedings of the ACM Multimedia*, Orlando, FL, USA, pp.121-130, 1999.
- [19] P. Messina, S. Brunett, D. Davis, T. Gottschalk, D. Curkendall, L. Ekroot, and H. Siegel. Distributed Interactive Simulation for Synthetic Forces. *In proceedings of the 6th Heterogeneous Computing Workshop*, Geneva, Switzerland, April 1, 1997.
- [20] Z. Choukair and D. Retailleau. Integrating QoS to Collaborative Distributed Virtual Reality Applications. *In proceedings of the third IEEE International Symposium on Object-Oriented Real-Time Distributed Computing*, March, 2000.
- [21] M. Mauave. Consistency in replicated continuous interactive media. *In Proceeding of ACM Conference on Computer Supported Cooperative Work*, Philadelphia, PA, USA, pp. 181-190, December 1998.
- [22] C. Diot, L. Gautier. A distributed architecture for multiplayer interactive applications on the Internet. *IEEE Network*, pp.6-15, July/August 1999.
- [23] D.L. Mills. Network Time Protocol (Version 3) specification, implementation and analysis. *DARPA Network Working Group Report RFC-1305*, University of Delaware, 1992.
- [24] J. Lui, O. So, and T. Tam. Deriving Communication Sub-Graph and Optimal Synchronizing Interval for a Distributed Virtual Environment System. *In Proceedings of IEEE International Conference on Multimedia Computing and Systems*, Vol II, Florence, Italy, June, 1999.
- [25] F. Cristian, H. Aghili, R. Strong, and D. Dolev. Atomic broadcast: From simple message diffusion to Byzantine agreement. *Information and Computation*, 118(1):158-179, April 1995.
- [26] S. Zhang and E. Lee. Efficient Global Allocation of Synchronous Bandwidths for Hard Real-Time Communication with the Timed Token MAC Protocol. *In Proceedings of Sixth International Conference on Real-Time Computing Systems and Applications*. December, 1999, Hong Kong.

Xiao Qin was born in February 25, 1974. He received a B.S. and M.S. degree both in computer science from Huazhong University of Science and Technology (HUST) in 1992 and 1999, respectively. From July 1999 to June 2000, he worked as a research assistant at HUST. In July 2000, he became a Ph.D. student at the school of computing and information technology, Griffith University, Australia. Currently, he is a Ph.D. student at Department of Computer Science and Engineering, University of Nebraska-Lincoln, USA. He has served as an editor for IEEE Distributed System Online for one year since October 2000. He is a student member of IEEE and Computer Society. His research interests are parallel processing, real-time computing, collaborative computing and distributed systems. The work reported in this paper had been done by the author when he was a Ph.D. student at Griffith University.