International Journal of Computer Science 3;4 © www.waset.org Fall 2008

Coordination for Synchronous Cooperative Systems Based on Fuzzy Causal Relations

Luis A. Morales Rosales, Saul E. Pomares Hernandez, and Gustavo Rodriguez Gomez

Abstract**—**Synchronous cooperative systems (SCS) bring together users that are geographically distributed and connected through a network to carry out a task. Examples of SCS include Tele-Immersion and Tele-Conferences. In SCS, the coordination is the core of the system, and it has been defined as the act of managing interdependencies between activities performed to achieve a goal. Some of the main problems that SCS present deal with the management of constraints between simultaneous activities and the execution ordering of these activities. In order to resolve these problems, orderings based on Lamport's *happened-before relation* have been used, namely, causal, Δ-causal, and causal-total orderings. They mainly differ in the degree of asynchronous execution allowed. One of the most important orderings is the causal order, which establishes that the events must be seen in the *cause-effect* order as they occur in the system. In this paper we show that for certain SCS (e.g. videoconferences, tele-immersion) where some degradation of the system is allowed, ensuring the causal order is still rigid, which can render negative affects to the system. In this paper, we illustrate how a more relaxed ordering, which we call Fuzzy Causal Order (FCO), is useful for such kind of systems by allowing a more asynchronous execution than the causal order. The benefit of the FCO is illustrated by applying it to a particular scenario of intermedia synchronization of an audio-conference system.

Keywords—Event ordering, fuzzy causal ordering, happened-before relation and cooperative systems.

I. INTRODUCTION

SYNCHRONOUS cooperative systems (SCS) have been an important research topic of Computer-Supported Cooperative Work (CSCW). The SCS bring together users which are geographically distributed and connected through a network to carry out a task. Examples of SCS include Tele-Immersion and Videoconferences. In SCS the coordination is the core of the system, and it has been defined as the act of managing interdependencies between activities performed to achieve a goal [1]. According to [2], some of their main problems are:

- Simultaneity constraints between activities: activities are dependent because they need to occur at the same time; otherwise, they simply cannot occur. A well known example of this kind of constraint is a shared resource (e.g. only one activity can write in a database in order to keep it consistent).
- Execution ordering between activities: activities are dependent because they need to appear in a certain order (e.g. a file must be opened before write operations can be done).

An important topic for the SCS, which is of interest to this work, and that resolves the problems mentioned above is *event ordering*. The event ordering in cooperative systems consists in establishing a certain order among the events that occur according to some particular criteria. Respecting the ordering of events ensures the coherency of the system. According to the chosen criteria, the resulting event ordering allows a greater or smaller degree of asynchronous execution. In cooperative systems that communicate only by message passing, there are three kinds of events: *internal*, *send* and *receive* events. The *internal* events occur inside a process, and they are never known by the rest of the participants. The *send* and *receive* events, on the other hand, are those through which the participants communicate and cooperate. In this paper, only the *send* and *receive* events are considered since they modify the global state of a system.

There are two broad categories for event ordering used in cooperative systems: total ordering and partial ordering. We refer in this paper only to those that are based on Lamport's relation. For total ordering we have the total-causal order, which is the strictest ordering in cooperative systems is used for total ordering; it establishes only one linearization that is consistent with the causal ordering among all the events that occur in the system, even those that occur concurrently. For that reason, the execution of the system is considered as synchronous. The partial ordering presents two variants: the causal order [3] and the Δ-causal order [4]. The main difference among these orders is that the Δ-causal considers that the events have an associated lifetime. The causal order establishes that for each participant in the system the events must be seen in the *cause-effect* order as they have occurred, whereas the Δ-causal order establishes that the events must be seen in the *cause-effect* order only if the cause has been seen before its lifetime expires. Otherwise, the *cause-effect* is considered to be broken, and therefore inexistent.

It is important to note that no type of the event ordering is better than another. Each event ordering is meant to be used in a particular type of problem, where it ensures the necessary ordering so as to satisfy its consistency constraints.

In this paper it is shown that for certain cooperative

Luis A. Morales Rosales is a PhD student, Saul E. Pomares Hernandez and Gustavo Rodriguez are researchers at the Computer Science Department at the National Institute of Astrophysics, Optics and Electronics (INAOE), email: {lamorales, spomares, grodrig}@inaoep.mx

domains (e.g. scheduling, planning, and intermedia synchronization) where some degradation of the system is allowed, ensuring the causal order based on Lamport's relation is still rigid, which can render negative effects to the system (e.g. the halt of the system, discarded data and delivery delay of the event). The allowed degradation differs in each domain according to the problem to solve. For example, in the scheduling domain for complex problems, optimal schedulers are computationally heavy, and in some cases, it is practically impossible to construct them. In these cases, it is preferable to use a near-optimal scheduling, which ensures a minimum of application requirements, such as bandwidth, access time, and lost rate. In the planning domain, sometimes it is not possible to carry out the entire set of tasks since they have some conflict among them. Therefore, a planner can identify what tasks must be executed in order to satisfy the maximum number of constraints, and therefore, maximize the performance of the system. In the domain of intermedia synchronization, the degradation can refer to the synchronization error allowed among the multimedia data. For example, the synchronization error for a dialogue among participants (audio-audio streams in real time) is acceptable if it is within ±120*ms*.

In this paper, we propose a Fuzzy Causal Order (FCO) for synchronous cooperative systems which allows a more asynchronous and relaxed execution than the causal order. With the FCO ordering, we can give more importance to the coherency of the system than to the logical dependency that exists between the events. The FCO is based on the fuzzy causal relation (FCR) and the fuzzy causal consistency (FCC) that we will show in Section IV. In a broad sense, the fuzzy causal relation can establish a *cause-effect* degree between the events indicating "*how long ago*" an event *a* happened before an event *b*. In order to know "*how good*" the coherency of the system is in a certain time, the fuzzy causal consistency is used. Besides, we will show the benefits of the FCO by applying it to the concrete problem of intermedia synchronization of an audio-conference system.

The outline of this paper is as follows. First, the state of the art is presented. Next, some basic concepts are introduced in the preliminaries section. After that, the fuzzy causal relation and the fuzzy causal consistency for synchronous cooperative systems are described. Subsequently, we show how the fuzzy causal relation and the fuzzy causal consistency are applied to the intermedia synchronization problem. Finally, some conclusions are presented.

II. STATE OF THE ART

The state of the art is presented in two sections. In the first section, the main works that in general include the concept of fuzzy relation are explained. The second section includes the works that have used some concepts of fuzzy logic applied to the problem of intermedia synchronization in SCSs.

A. *Fuzzy relation*

The fuzzy relation is widely used in the fuzzy logic area. This relation indicates in a broad sense the degree of compatibility among two concepts. The first work that introduces the concept of fuzzy causal relation in order to establish a fuzzy causal relation (degree of affectation) among events or concepts of the system is the work of Fuzzy Cognitive Maps. Hence, it is important to remark the main differences of this work with the fuzzy causal relation presented in this paper.

Fuzzy cognitive maps (FCM) are fuzzy weighted directed graphs with feedback that create models that emulate the behavior of a complex process using fuzzy causal relations [6]. Unlike what is proposed in this paper, the concept of fuzzy causal relation used for the FCM does not apply for the event ordering in cooperative systems because in order to construct the fuzzy weighted directed graph for a system, it is necessary to know the degree of affectation (behavior) of all events in the system before launching it. Here, we would like to remark that the FCMs are constructed off-line.

Baldoni and Giacomin in [7] integrate ideas of flexibility and uncertainty into Allen's interval-based temporal logic and define an interval fuzzy algebra IA^{fuzz}. This work focuses on dealing with the qualitative aspect of temporal knowledge for the solution of planning problems and prioritized constraints to express the degree of satisfaction needed. They only label the different relations among intervals with a degree of satisfaction that the search of the solution must satisfy. Besides, they must also have prior knowledge of the behavior and the relations of the system, so the interval fuzzy algebra cannot be applied for the event ordering in distributed systems.

B. *fuzzy logic in cooperative systems*

Some of the main works that have included concepts of fuzzy logic in cooperative systems are focused on trying to solve the multimedia synchronization problem on demand, which consists in assuring the temporal appearance order of the data at the reception of every participant as they were sent. It is important to remark that none of these works have used the concepts of fuzzy causal relation, neither the fuzzy causal consistency for cooperative systems, nor a solution that can be applied for the synchronization in real time using fuzzy concepts in a DMS as it is presented in this work.

Zhou and Murata in [8] presented a temporal petri-net model called Extended Fuzzy Timing Net for distributed multimedia synchronization. Among their main characteristics, they contemplate temporal uncertain requirements, making a measurement of the quality of service parameters required by the application in order to check if they are satisfied. They use a trapezoidal membership function to calculate and to know if the data are synchronized (e.g. audio and video). The model is based on the concept of master-slave to carry out the synchronization. Extended Fuzzy Timing Net model needs a set of forward relations between multimedia objects, which are specified by the designer of the application.

Ramaprabhu et. al in [9] have presented algorithms for the broadcasting of video on demand. In this work the *fuzzycast* concept is introduced, which consists in determining the delivery order of the data based on the technique of the nearest neighbor with regard to the time in which they were generated, considering parameters such as available bandwidth, delay and buffer space. They use a server for the data transmission to all the participants of the group.

Coelho et. al in [10] presented a methodology for a high-level specification and decentralized coordination of temporal interdependences among objects of multimedia documents. In this work, they introduced the use of causality to establish fuzzy rules to carry out the multimedia synchronization. Among their main characteristics are: the specification is performed by the user indicating how the events will be synchronized using fuzzy scripts; they classify the entities that compose the scenes to verify the consistency of their temporal relations and the generation of the synchronization mechanism that will be associated to each multimedia entity. The fuzzy parameters for the synchronization are explicitly defined by the designer of the application. They use a global reference to determine the synchronization time, as well as a scheme producer-consumer to establish synchronization points. The specification is made offline, so it has to be defined before the desirable reproduction for the objects.

III. PRELIMINARIES

In this section, some basic definitions are described to understand the fuzzy causal order for cooperative systems. In addition, these definitions are used to clarify the main differences between the causal relation, the Δ-causal relation and the fuzzy causal relation.

A. The System Model

Processes. The application under consideration is composed of a set of processes $P=\{i, j\ldots\}$ organized into a group which communicate by broadcast asynchronous messages passing. In our case, the members of the group g are defined as $Memb(g)=P$.

Messages. We consider a finite set of messages M, where each message $m\in M$ is identified by a tuple (*participant*, *integer*), $m=(p, x)$ where $p\in P$ is the sender of m, denoted by $Src(m)$, x is the local logical clock for messages of p, when m is broadcasted. The set of destinations $Dest(m)$ of message m is composed of the participants connected to the $Group(Dest(m)=Memb(g))$. We denote the messages sent by the process p by $M_p=\{\, m\in M : Src(m) = p \,\}$.

Events. Let m be a message, it is denoted by $send(m)$ the emission event of m by $Src(m)$, and by $delivery(p,m)$ the delivery event of m to participant p connected to $Group(m)$. The set of events associated to M is then the set $E = \{send(m) : m\in M\} \cup \{delivery(p,m) : m \in M \wedge p \in Dest(m)\}$. An emission event $send(m)$ where $m=(p,x)$ may also be denoted by $send(p,m)$ or $send(m)$ without ambiguity. The subset $E_p\subseteq E$ of events involving p is $E_p=\{send(m) : k=Src(m)\} \cup \{delivery(p,m) : p\in Dest(m)\}$.

Intervals. We consider a finite set I of intervals, where each interval $A\in I$ is a set of messages $A\subseteq M$ sent by a participant $p=Part(A)$, defined by the mapping $Part{:}I\to P$. Formally, we have $m\in A \Rightarrow Src(m)=Part(A)$. Due to the sequential order of $Part(A)$, we have for all $m, m' \in A$, $m \to m'$ or $m' \to m$. We denote by a^- and a^+ the endpoint messages of A, such that for all $m \in A$: $a^-\neq m$ and $a^+\neq m$ implies that $a^- \to m\to a^+$.

B. Background and Definitions

Happened-before relation proposed by Lamport

The happened-before relation, also known as causal relation, was proposed in [5] and was defined by Lamport as follows:

Definition 1. The causal relation "→" is the least partial order relation on the set E that satisfies the three following conditions:

- If a and b are events belonging to the same process, and a was originated before b, then $a\to b$.
- If a is the send message of a process, and b is the reception of the same message in another process, then $a\to b$.
- If $a\to b$ and $b\to c$, then $a\to c$.

By using "→", Lamport defines that two events are concurrent as follows:

$a \parallel b$ if $\neg\,(a\to b \vee b\to a)$

Causal order delivery proposed by Birman

Birman, based on Lamport's relation, defined that for group communication [3], a behavior or set of behaviors satisfies the causal order delivery if the diffusion of a message m causally precedes the diffusion of a message m', and the delivery of m causally precedes the delivery of m' for all participants that belong to the destinations of m and m'. Formally, this is defined as:

Definition 2. The causal order delivery must satisfy the following condition:

If $send(m) \to send(m') \Rightarrow \forall p \in dests(m) \cap dests(m')$:
$delivery(m) \to delivery(m')$

Δ-Causal order delivery proposed by Baldoni

The Δ- causal relation was introduced in [4] as an extension to Birman's work. The Δ- causal relation assigns a lifetime to the events, which support lost messages by preserving the order of precedence established by Lamport. The Δ-causal delivery is formally defined as:

Definition 3. A distributed computation $\hat{E}$ respects a Δ-causal order if:

- All the messages $M(\hat{E})$ that arrive in Δ are delivered in Δ; all others are never delivered (they are considered to be lost or discarded);
- All the events of delivery respect a causal order.

Where $\hat{E}=(E,\to)$, is a set of events partially ordered (*send* and *delivery*) and $M(\hat{E})$ is the set of all the messages exchanged in $\hat{E}$.

IV. FUZZY CAUSAL RELATION AND FUZZY CAUSAL CONSISTENCY

A. Fuzzy causal relation (FCR)

The fuzzy causal relation (FCR) is denoted by "$a \xrightarrow{\lambda} b$". The FCR is based on a notion of "*distance*" among the events. The *distance*, according to the addressed problem, can be established considering three main domains: *spatial, temporal* and/or *logical*. The reference for the logical domain is the event ordering based on Lamport's relation. Using the notion of distance, the FCR establishes a *cause-effect degree* that indicates "*how long ago*" an event *a* happened before an event *b*.

The distance between events is determined by the fuzzy relation *DR*: $E \times E \rightarrow [0, 1]$, which is established from the union of sets of membership functions, R_S(spatial), R_T(temporal), and R_L(logical), one set for each domain. It is formally defined as follows:

$$DR(a,b) = R_S(R_1 \cup R_2 \cup ... R_o) \cup \mathrm{R_T}(R_1 \cup R_2 \cup ... R_r) \cup R_L(R_1 \cup R_2 \cup \ldots \cup R_s)$$

The number of membership functions, R_i, by each domain is determined according to the problem to resolve. The fuzzy union operator chosen for intra and inter domains is the *max* operator $max(R_1,..., R_k)$.

In this paper, one hypothesis considered for the FCR is that "*closer*" events have a stronger *cause-effect* relation, according to the addressed problem. For this reason, it is established that the *DR* grows monotonically, and it is directly proportional to the *spatial*, *temporal* and/or *logical distances* between a pair of events. This means, for example, that a *DR(a,b)* with a value tending to zero indicates that the events a and b are "*closer*".

It is important to remark that the *DR* cannot determine precedence dependencies among events, it only indicates certain *distance* among them. Hence, in order to establish a *cause-effect* degree (precedence) among events, we formally define the Fuzzy Causal Relation using the values of the *DR* as follows:

Definition 4. The fuzzy causal relation "   " on a set of events *E* satisfies the two following conditions:

$a \xrightarrow{\lambda} b$ If $a \rightarrow b \;\wedge\; 0 \leq DR(a,b) < 1$

$a \xrightarrow{\lambda} c$ If $\exists b \mid a \rightarrow b \rightarrow c \;\wedge\; DR(a,b) \leq DR(\overset{\xrightarrow{\lambda}}{a,c}) : DR(a,b), DR(a,c) < 1$

The first condition establishes that two events (a,b) are fuzzy causal related if *a* happened before *b,* and the value of the $DR(a,b)$ is smaller than one. The second condition is the transitive property. This condition establishes that two events (a,c) are fuzzy causal related if there exists an event *b* such that *a* happened before *b* and *b* happened before *c*; in addition, the values for $DR(a,b)$ and $DR(a, c)$ monotonically grow, and they are smaller than one. If any of these conditions are satisfied, the value of the $DR(a,b)$ determines the *cause-effect* degree between the present pair, and we refer to it as $FCR(a,b)$. In any case, when the value of the $DR(a,b)$ is equal to one, this means that the events no longer have a *cause-effect* relation.

By using Lamport's relation, a pair of events are concurrent if $\neg (a \rightarrow b \vee b \rightarrow a)$, expressed as "$a \parallel b$". In our case, based on the value of the *DR,* we introduce the concept of Fuzzy Concurrent Relation (FCNR), which is formally defined as:

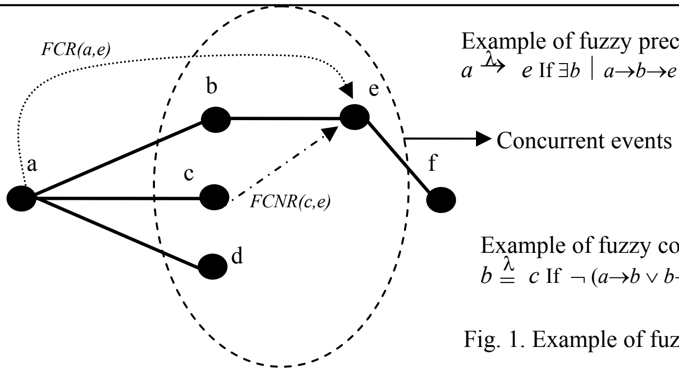
Definition 5. Two events are fuzzy concurrent "$a \overset{\lambda}{=} b$", if the following condition is satisfied:

$a \overset{\lambda}{=} b$ If $\neg (a \rightarrow b \vee b \rightarrow a) \;\wedge\; ((DR(a, b) = DR(b, a)) < 1)$

A fuzzy concurrent relation among two events exists if the events are concurrent and the values of their *DR* are equal and less than the unit, which we refer to as $FCNR(a, b)$.

This means that it can established that *spatial* and/or *temporal* relation(s) among the events, even when a *logical* precedence relation cannot be determined. It is observed that when the *DR* for a pair of concurrent events (a, b) is equal to or less than one, this means that the event *a* has some effect on the event *b* and viceversa. Hence, for the fuzzy concurrent events *a* and *b*, the order (a,b) or (b,a) is indistinct for the system.

In order to illustrate the use of the FCR and the FCNR, Fig. 1 shows a scenario to determine the fuzzy precedence and the fuzzy concurrency between events.



Example of fuzzy precedence among causal messages $FCR(a,e)$

$a \xrightarrow{\lambda} e$ If $\exists b \mid a \rightarrow b \rightarrow e \;\wedge\; DR(a,b) \leq DR(a,e) : DR(a,b), DR(a,e)) < 1$

Example of fuzzy concurrent messages $FCNR(c,e)$

$b \overset{\lambda}{=} c$ If $\neg (a \rightarrow b \vee b \rightarrow a) \;\wedge\; \mathrm{D}R(b,c) = \mathrm{D}R(c,b) : \mathrm{D}R(b,c), \mathrm{D}R(c,b) < 1$

Fig. 1. Example of fuzzy precedence in a distributed system

For example, for the case of the relation among the events *a* and *e*, the *FCR*(*a*, *e*) determines if a *cause-effect* relation exists that must be taken into account for the event ordering . For the fuzzy concurrent events *e* and *b*, the *FCNR*(*c*, *e*) identifies that there is a certain *spatial* and/or *temporal* relation among them.

B. Fuzzy Causal Consistency (FCC)

The Fuzzy Causal Consistency (FCC) is based on the FCR. The goal of the FCC is to indicate "*how good*" the coherency of the system is in a certain time. The meaning of the coherency can be indicated according to the problem to resolve. By calculating the value of the FCC, it can be determined if the coherency of the system is good enough to continue.

The FCC is calculated by the weighting average of the fuzzy causal relations for every event contained in the causal history *H*(*a*) of the event *a* from which the performance of the system wants to be known. The values of the fuzzy causal consistency in our case are normalized in the interval [0,1].

Fig. 2 shows the strategy to obtain the fuzzy causal consistency for an event *a* at a process *p*. The set *H*(*a*) contains the events that are causally related to the event a, which is the event from the FCC is going to be calculated.

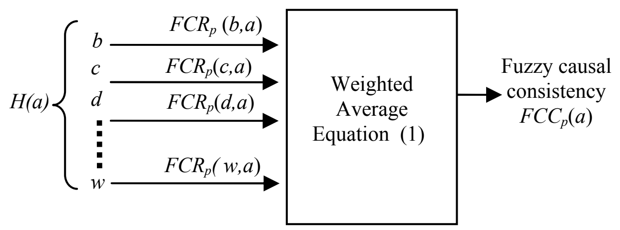


Fig. 2. Fuzzy causal consistency

$$FCC_p(a) = \frac{\sum GP(b) FCR_p(a,b)\ \forall b \in H(a)}{\sum GP(b)\ \forall b \in H(a)} \qquad (1)$$

where:

***GP*(*b*)** is a weighing degree used to determine priorities or weight to every fuzzy causal relation when it is needed.

***FCR*$_p$(a, *b*)** is the fuzzy causal relation of a pair of events (*a*,*b*) at a process *p*.

C. Fuzzy causal delivery for event ordering

The Fuzzy Causal Delivery Order (FCO) is based on the concepts of FCR and FCC. The goal of the FCO is to allow a more asynchronous delivery of events compared with the causal delivery order. The FCO establishes that if for a pair of messages (*m*, *m*') the *send* of *m* fuzzy causally precedes the *send* of *m*', then for all destinations of *m* and *m*' the *delivery* of *m* precedes *m*' or viceversa if and only if the coherency of the system determined by the fuzzy causal consistency of *m* is inside the maximum FCC allowed by the system (FCC_{max}). Formally, the FCO is defined as follows:

Definition 6. The fuzzy causal delivery order must satisfy the following condition:

If *send*(*m*) $\xrightarrow{\lambda}$ *send*(*m*') **then**

$\forall p \in dests(m) \cap dests(m'),\ FCC_p(m) \leq FCC_{max}$

1. $delivery_p(m) \rightarrow delivery_p(m')$ **or**
2. $delivery_p(m') \rightarrow delivery_p(m)$

where:

FCC_p(*m*) is the fuzzy causal consistency for the event *m* at its reception by the participant *p*.

FCC_{max} is the maximum FCC allowed according to the performance required by the system.

The FCO establishes that if the value of the fuzzy causal consistency (coherency of the system) for the event *m*, $FCC_p(m)$ is equal to or lower than the maximum fuzzy casual consistency tolerated by the system, FCC_{max}, then the delivery of a pair of events can be carried out in the form, (*m*, *m*') or (*m*', *m*), allowing the interchange of events. As a direct consequence of this property, it can be observed that the FCO can perform a more asynchronous and relaxed events delivery.

V. FUZZY CAUSAL ORDER VERSUS CAUSAL ORDER

In this section, we will show the usefulness of the FRC and the FCC and the way to use them in cooperative systems. First, the main differences and advantages of the *fuzzy causal relation* and the *happened before relation* proposed by Lamport will be presented. Next, we will describe how the FCR and the FCC can be applied for the concrete problem of intermedia synchronization.

Let's take the distributed multimedia scenario depicted in Fig. 3. In this case, the participant *Part*(X) sends video and the participant *Part*(Y) sends audio; these continuous data must to be synchronized at their delivery at the participant *k*. The continuous media are widely represented by intervals. The intermedia synchronization problem is commonly solved by synchronizing the interval endpoints, which are causal dependency messages: For more details, refer to [11].

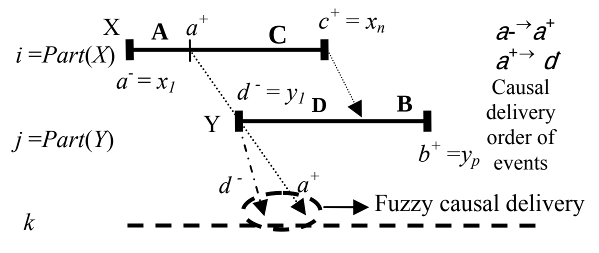


Fig. 3. Example of the fuzzy causal

For the strict causal algorithms based on Lamport's relation, the delivery of the event d^- implies that the event a^+ has been delivered. Due to delays in the network or loss of the event a^+, the delivery time of d^- can be infinite.

For the Δ-causal algorithms, the Δ-causal order ensures that the delivery of the event d^- is carried out if it fulfills the following conditions:

−the event d^- has been received in its lifetime (Δ), and

−the events that precede d^- have been delivered in a causal order or have been discarded because their lifetime has expired.

For this case, the delivery of d^- will be carried out only if a^+ has been delivered or discarded.

These algorithms maintain the strict causal order proposed by Lamport; their main advantage is that the maximum delivery waiting time for the events is determined according to the lifetime established.

In this work, which is based on the fuzzy causal order, the event d^- can be delivered immediately before the event a^+, if and only if the fuzzy causal consistency is within the parameters established to maintain the coherency of the system. For the case of the intermedia synchronization problem, the performance is linked to the maximum synchronization error allowed.

VI. APPLYING THE FCR AND THE FCC TO AN AUDIO-CONFERENCE SYSTEM

In order to show the situation of media data out of phase (coodination error), a scenario that consist of a group of four hosts is presented; three of them transmitting live media data (W, X, and Y hosts), while the other functions only as Client (Host Z). Refer to Fig. 4.

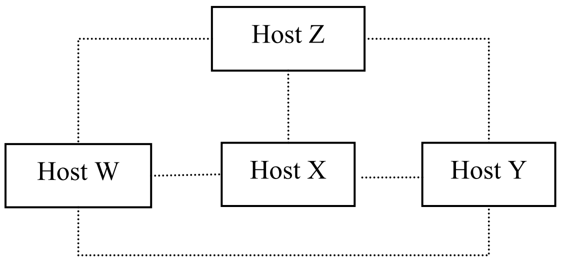


Fig. 4. Example of a distributed multimedia scenario

This scenario represents a dialogue among some participants. Each host has two inputs and one output communication channels. The sending host only transmits one media (audio), which is codified as a plane object. See Fig. 5.

Even when the audio is considered to be continuous, its transmission is in fact non-continuous since compression techniques, such as silence compression, are used. In this case, a *begin* message is sent each time that the voice activity is initiated, and an *end* message is sent each time that there is a low or null voice activity. The remaining audio frames are sent as *fifo* messages.

(omitted paragraph since it was repeated)

In this scenario, each host has the synchronization mechanism running. The maximum waiting time for every pair of media data is established at $\Delta=120ms$, which is the maximum synchronization error established for the reproduction of an audio-audio communication in real time [12].

Based on the system model previously established, a scenario is presented in Fig. 5. It will illustrate how the FCR, the FCC, and the FCO for the message m_i are calculated.

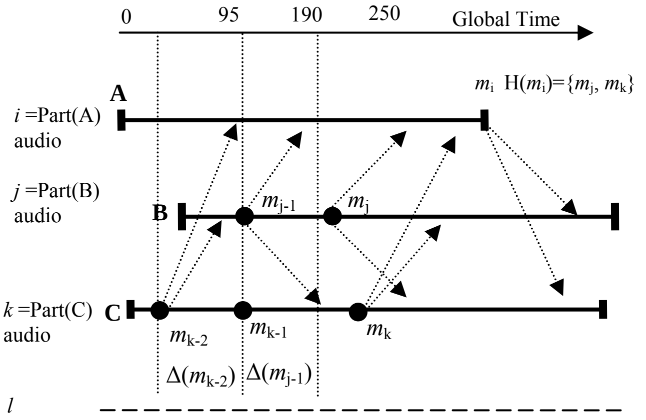


Fig. 5. Example of a distributed multimedia scenario

The FCO condition is evaluated at the reception of the synchronization message m_i by the participant l. In order to know if m_i can be delivered it has to calculate the value of $FCC_l(m_i)$. For the value of the $FCC_l(m_i)$ it is going to determine the FCRs for each participant contained in the causal history of m_i. The FCR is calculated between the messages contained in the causal history of m_i, H(m_i)=(m_j, m_k), and the last message received by the participant l from each participant contained in H(m_i); in this case, FCR(m_{j-1}, m_j) for the participant j and the FCR(m_{k-2}, m_k) for the participant k.

As a recall, for the FCR the domains that the relation is going to include must be chosen. In this paper, in order to resolve the event ordering problem, only the logical and temporal domains are considered. For each domain, one membership function, R_D and R_N, respectively is defined. The spatial domain is not included since the audio data does not consider it. On the other hand, the logical domain is considered because, as we previously showed, the synchronization is based on the causal interval endpoints of the media involved. The temporal domain also is included because the synchronization error among the media involved is measured in physical time units (milliseconds). These domains give us useful information to determine the data delivery order in the synchronization problem according to the performance desired. The separation among the events can be used as distances for the logical domain according to the event ordering chosen: local causal distance (*fifo*), causal distance introduced in [13] (*causal order*), total distance (*total order*) and the total-causal distance (*total causal order*). In this paper, the local causal distance was chosen because the interest of the synchronization problem is to focus on measuring the separation among each media data. For the temporal domain, we use as distance the physical time.

VII. CONCLUSIONS

This paper shows how the Fuzzy Causal Order can be useful for synchronous cooperative systems to resolve problems of coordination. It is shown that the FCO allows a more asynchronous and relaxed ordering, which is based on the definitions of Fuzzy Causal Relation and Fuzzy Causal Consistency. With the FCR, we establish the *cause-effect* degree between events by considering *spatial*, *temporal*, and/or *logical* domains. The FCR indicates "*how long ago*" an event *a* happened before an event *b*. The FCC indicates "*how good*" the coherency/consistency of the system is in a certain time, which will help us determine if the system is good enough to continue or if one needs to perform a corrective action to keep it running. Finally, we have shown the benefits of the FCO by applying it to the intermedia synchronization problem on an audio-conference scenario. Future aims of this work include research on how to apply the FCR, the FCC and the FCO in order to resolve problems in other domains, such as planning and scheduling.

REFERENCES

[1] T.W. Malone and K. Crowston, "The interdisciplinary study of coordination", ACM Computing Surveys, Vol. 26, No. 1, March 1994.

[2] A Coordination Component Framework for Open Systems Juan Carlos Cruz, Sander Tichelaar, Oscar Nierstrasz SCG Report, IAM, University of Bern,1996.

[3] Birman, A. Schiper, and P. Stephenson, "Lightweight Causal and Atomic Group Multicast," ACM Trans. Compt. Syst. Vol. 9, No. 3, ISSN:0734-2071, Aug. 1991, pp 272-314.

[4] Roberto Baldoni and Ravi Prakash and Michel Raynal and Mukesh Singhal. "Efficient Δ-causal broadcasting". International Journal of Computer Systems Science and Engineering, Volume 13, Number 5, 1998, pp. 263-269.

[5] Leslie Lamport. "Time, Clocks, and the Ordering of Events in a Cooperative System". Communications ACM, Volume 21, Number 7, ISSN:0001-0782, July 1978 pp. 558-565.

[6] José Aguilar, "Dynamic Random Fuzzy Cognitive Maps". Computación y Sistemas, Abril-Junio 2004, Volumen 7, Numero 004, Instituto Politécnico Nacional, Distrito Federal, México, ISSN 1405-5546pp. 260-271.

[7] S. Badaloni, M. Giacomin, "The algebra IAfuz: a framework for qualitative fuzzy temporal reasoning". Artificial Intelligence Volume 170, Number 10, Elsevier, July 2006, pp. 872-908.

[8] Yi Zhou, Tadeo Murata. "Modeling and Analysis of Distributed Multimedia Synchronization by Extended Fuzzy-Timing Petri Nets". Transactions of the Society for Desing and Process Science, Volume 5, Number 4, ISSN:1092-0617, December 2001, pp. 23-37.

[9] Ramaprabhu Janakiraman, Marcel Waldvogel and Lihao Xu. "Fuzzycast: Efficient Video-on-demand over Multicast". Proceedings INFOCOM 2002, New York, NY, USA, June 2002.

[10] André L. V. Coelho, Alberto B. Raposo, Ivan L. M. Ricarte. "Bringing Flexibility to the Specification and Coordination of Temporal Dependencias among Multimedia Components". VII Simposio Brasileiro de Sistemas Multimídia e Hipermídia, Florianópolis, Brazil, SBC. 2001, pp. 37-52.

[11] Luis Morales Rosales, Saul Pomares Hernandez, "A Temporal Synchronization Mechanism for Real-Time Distributed Continuous Media". International Conference on Signal Processing and Multimedia Applications (SIGMAP'06), Setubal, Portugal, Aug. 2006, INSTICC Press, ISBN 972-8865-64-3, pp. 302-309

[12] Anna Haj Hać and Cindy X. Xue, "Synchronization in multimedia data retrieval". International Journal of Network Management, John Wiley & Sons, Inc. New York, NY, USA Volume 7, Issue 1 (January–February 1997), ISSN:1099-1190, pp. 33–62.

[13] Eduardo Lopez, Jorge Estudillo, Jean Fanchon and Saul E. Pomares, "A fault-tolerant causal broadcast algorithm to be applied to unreliable networks". The 17th IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS 2005), November 14-16.

Luis Alberto Morales Rosales is currently a PhD student in the Computer Science Department at the National Institute of Astrophysics, Optics and Electronics, in Puebla, Mexico. He received his M.S. from the same institute. His research efforts focus on partial order algorithms, multimedia synchronization, fuzzy control systems, and distributed systems. His Postgraduate Studies have been and continue to be supported by the National Council of Science and Technology of Mexico (CONACYT).

Saul Eduardo Pomares Hernandez is an Associate Researcher in the Computer Science Department at the National Institute of Astrophysics, Optics and Electronics, in Puebla, Mexico. He completed his PhD degree at the laboratory for Analysis and Architecture of Systems of CNRS, France in 2002, and his M.S. at the Centre of Research and Advanced Studies (CINVESTAVIPN) in Mexico. Since 1998, he has been researching in the field of distributed systems, partial order algorithms, multimedia synchronization, and security.

Gustavo Rodriguez Gomez is an Associate Researcher in the Computer Science Department at the National Institute of Astrophysics, Optics and Electronics, in Puebla, Mexico. He completed his PhD from the same institute in 2002, and his M.S. in mathematics at the Science Faculty of the National Autonomous University of Mexico (UNAM). He has been researching in the field of scientific computing, simulation of continuous processes, analysis and design of scientific software, and numerical solution of partial and ordinary differential equations.