# Journal of Applied Sciences

ISSN 1812-5654

science
alert

ANSInet
an open access publisher
http://ansinet.com

Journal of Applied Sciences 9 (8): 1441-1449, 2009
ISSN 1812-5654

# Fuzzy Causal Ordering of Events in Distributed Systems

Luis A. Morales Rosales, Saul E. Pomares Hernandez and Gustavo Rodriguez Gomez
Department of Computer Science, National Institute of Astrophysics, Optics and Electronics (INAOE),
Luis Enrique Erro No. 1, 72840, Tonantzintla, Puebla, Mexico

**Abstract:** Event ordering is an important research subject in Distributed Systems (DS). Event ordering addresses the problem of establishing a certain order among the events that occur in DS according to some particular criteria. The types of event orderings used for DS are no order, FIFO, causal, Δ-causal, total and causal-total. They mainly differ in the degree of asynchronous execution allowed. One of the most important orderings is the Causal Order (CO), which is based on Lamport's happened-before relation. It establishes that the events must be seen in the cause-effect order as they occur in the system. However, for certain applications, for example multimedia synchronization, where some degradation of the system is allowed, ensuring the CO based on Lamport's relation is rigid and negative affect the performance of the system. In this study, it is introduced a new ordering for DS in order to achieve a more asynchronous execution than the CO, this new ordering is called Fuzzy Causal Order (FCO). Besides, it is defined the Fuzzy Causal Relation (FCR) and Fuzzy Causal Consistency (FCC), the FCR establishes logical dependencies based on the precedence of events and by considering some kind of distance between their occurrences. With the notion of distance was possible to establish a cause-effect measure between two events a and b that indicates how long ago an event a happened before an event b. Through the FCC it was possible to determine how good the performance of the system is at a given moment. The usefulness of the FCO, FCR and FCC is showed by applying them to the concrete problem of intermedia synchronization in DS.

**Key words:** Event ordering, fuzzy causal ordering, intermedia synchronization, distributed systems

## INTRODUCTION

Event ordering in Distributed Systems (DS) consists in establishing a certain order among the events that occur according to some particular criteria. According to the chosen criteria, the resulting event ordering allows a greater or smaller degree of asynchronous execution. In a distributed system, 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. On the other hand, the send and receive events are those through which the participants communicate and cooperate. In this study, it is only considered the send and receive events since they modify the global state of a system. There are two broad categories for event ordering used in distributed systems: total ordering and partial ordering. For total ordering, there are two variants: total-causal order and total order. The total-causal order is the strictest ordering in distributed systems; it establishes only one linearization, 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. On the other hand, the total order establishes a sequential order for all the events that occur in the system without ensuring the causal order. The partial ordering presents two variants: the causal order (Birman *et al*., 1991) and the Δ-causal order (Baldoni *et al*., 1998). Both of them are exclusively based on the happened-before relation defined by Lamport (1978); the main difference 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 in existent.

Partial ordering is important since it allows that the ordering view concerning the set of events E of a system to differ among the participants; however, it does ensures that for a subset of events $E' \in E$, all participants will have the same consistent view according to the chosen criteria.

**Corresponding Author:** Luis Alberto Morales Rosales, Department of Computer Science,
National Institute of Astrophysics, Optics and Electronics (INAOE),
Luis Enrique Erro #1, 72840, Tonantzintla, Puebla, Mexico
Tel: [+52] (222) 266-3100/8207 Fax: [+52] (222) 266-3152

The smaller is E' and the fewer ordering constraints are required, the more asynchronous is the system since there are less events to order and less constraints between the events to accomplish. 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 study, it is claimed as hypothesis that for certain domains, for example scheduling, planning and intermedia synchronization, where some degradation of the system is allowed, ensuring the causal order strictly based on Lamport's relation is still rigid, which can render negative affects to the performance of 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 msec.

In this study, it is introduced a new event ordering for distributed systems that allows a more asynchronous execution than the causal order; this new ordering is called Fuzzy Causal Order (FCO). The FCO is based on the Fuzzy Causal Relation (FCR) and the Fuzzy Causal Consistency (FCC). The fuzzy causal relation establishes cause-effect dependencies among events based not only on their precedence dependencies but also by considering some kind of distance between the occurrences of the events. By using the notion of distance, it aims to establish a cause-effect degree that indicates how long ago an event a happened before an event b. Besides, the fuzzy causal consistency is based on the FCR, by considering some attributes of the addressed problem, it gives information about how good the performance of the system is at a given moment. There are two hypotheses behind this: first, according to the addressed problem, it is established that closer events have a stronger cause-effect relation and secondly, events with a stronger cause-effect relation have a greater impact (negative or positive) on the performance of the system. While the FCR is directly concerned with the first hypothesis, the FCC deals with the second one.

The usefulness of the fuzzy causal order is showed by applying it to the concrete problem of intermedia synchronization in a distributed multimedia system (DMS), where a certain synchronization error in the system is allowed according to the type of media involved for example continuous and/or discrete and the transmission mode (on-demand, or real-time).

## STATE OF THE ART

Here, in the first section, it is explained the main studies that have included the concept of fuzzy relation. The second section includes the studies that have used some concepts of fuzzy logic in order to solve the problem of intermedia synchronization.

**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 study in introduce the concept of fuzzy causal relation is the Fuzzy Cognitive Maps in order to establish a fuzzy causal relation, degree of affectation, among events or concepts of the system. Fuzzy Cognitive Maps (FCM) are fuzzy weighted directed graphs with feedback that create models that emulate the behavior of complex process using fuzzy causal relations (Aguilar, 2004). However, the concept of fuzzy causal relation used for the FCM cannot apply for the event ordering in distributed systems because to construct the fuzzy weighted directed graph for a system, it is needed to know the degree of affectation of all events in the system. It should be observed that the FCMs are off-line constructed.

Silvana and Giacomin (2006) integrated ideas of flexibility and uncertainty into Allen's interval-based temporal logic and define an interval fuzzy algebra $IA^{fuzz}$. This study is focus on to deal with the qualitative aspect of temporal knowledge for the solution of planning problems and prioritized constraints to express the degree of satisfaction needed. They just label the different relations among intervals with a degree of satisfaction that the search of the solution must to satisfy. Besides, they also must to know in advance the behavior and the relations of the system, so the interval fuzzy algebra cannot apply for the event ordering in distributed systems.

**Intermedia synchronization using fuzz logic concepts:** Some of the main studies that have included concepts of fuzzy logic in distributed systems are focus on try to solve the multimedia synchronization problem on demand, which consists in to assure the temporal appearance order of the data at the reception of every participant as they were sent. This problem is in essence an event ordering problem. It is important to remark that none of these studies have developed the concepts of fuzzy causal relation neither the fuzzy causal consistency for distributed 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 study.

Zhou and Murata (2001) 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 services 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.

Janakiraman *et al.* (2002) give algorithms for the broadcasting of video on demand. In this study, the fuzzycast concept is introduced and consists in to determine the delivery order of data based on the technique of the nearest neighbor taken into account the generation time of data. They use parameters such as, available bandwidth, delay and buffer space and a server for the data transmission to all the participants of the group.

## PRELIMINARIES

Here, some basic definitions are described to understand the fuzzy causal relation. Besides, these definitions are used to clarify the main differences between the strict causal relation, the Δ-causal relation and the fuzzy causal relation.

### The system model

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

**Messages:** The system considers a finite set of messages M, where each message $m \in M$ is identified by a 2-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)). The messages sent by the process *p* are denote by $M_p = \{m \in M : \mathrm{Src}(m) = p\}$.

**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 $E = \{\mathrm{send}(m) : m \in M\} \cup \{\mathrm{delivery}(p, m) : m \in M \wedge p \in \mathrm{Dest}(m)\}$. 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 $E_p = \{\mathrm{send}(m) : k = \mathrm{Src}(m)\}\ \{\mathrm{delivery}(p, m) : p \in \mathrm{Dest}(m)\}$.

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

### Background and definitions

**Happened-before relation proposed by Lamport:** The happened-before relation also known as causal relation was defined by Lamport (1978) 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 $a \rightarrow b$
- If a is the send message of a process and b is the reception of the same message in another process, then $a \rightarrow b$
- If $a \rightarrow b$ and $b \rightarrow c$ then $a \rightarrow c$

By using →, Lamport (1978) defines that two events are concurrent as follows:

$$a \parallel b \text{ if } \neg (a \rightarrow b \vee b \rightarrow a)$$

**Causal order delivery proposed by Birman:** Birman *et al.* (1991), based on the Lamport's relation, defined for group communication that 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 belongs to the destinations of m and m'. Formally defined as follow:

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

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

**Δ-causal order delivery proposed by Baldoni:** The Δ-causal relation was introduced by Baldoni *et al*. (1998) as an extension to Birman's study. The Δ-causal relation assigns a lifetime to the events, which allows support the messages lost, by preserving the order of precedence established by Lamport (1978). The Δ-causal delivery is formally defined as:

**Definition 3:** A distributed computation Ê respects a Δ-causal order if:

- All the messages M(Ê) that arrive in Δ, they 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, $\hat{E}=(E, \rightarrow)$, is a set of events partially ordered (send and delivery) and M(Ê) is the set of all the messages exchanged in Ê.

## FUZZY CAUSAL RELATION AND FUZZY CAUSAL CONSISTENCY

**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 (1978) 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:

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

The number of membership functions, R, by each domain is determined in accordance of the problem to resolve. The fuzzy union operator chosen for intra and inter domains is the max operator max $(R_1,\ldots, R_k)$.

In this study, 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, only indicates certain distance among them. Hence, in order to establish a cause-effect degree (precedence) among the fuzzy causal relation is formally defined by using the values of the DR as follows:

**Definition 4:** The fuzzy causal relation $\xrightarrow{\lambda}$ 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 $\exists b \mid a \rightarrow b \rightarrow c \wedge DR(a,b) \leq DR(a,c)$: $DR(a,b)$, $DR(a,c) < 1$

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 establish 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. Besides, the values for DR(a,b), DR(a, c) monotonically grows and they are smaller than one. If any of these conditions is satisfied, the value of the DR(a,b) determines the cause-effect degree between the present pair and it is represented by FCR(a,b). In any case when the value of the DR(a, b) is equal one, means that the events do not have anymore a cause-effect relation.

By using Lamport's (1978) relation a pair of events are concurrent if $\neg (a \rightarrow b \vee b \rightarrow a)$, expressed as $a \parallel b$. In this study, based on the value of the DR, the concept of Fuzzy Concurrent Relation (FCNR) is formally defined as:

**Definition 5:** Two events are fuzzy concurrent $a \underline{\underline{\lambda}} b$, if the following condition is satisfied:

$a \underline{\underline{\lambda}} b$ 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

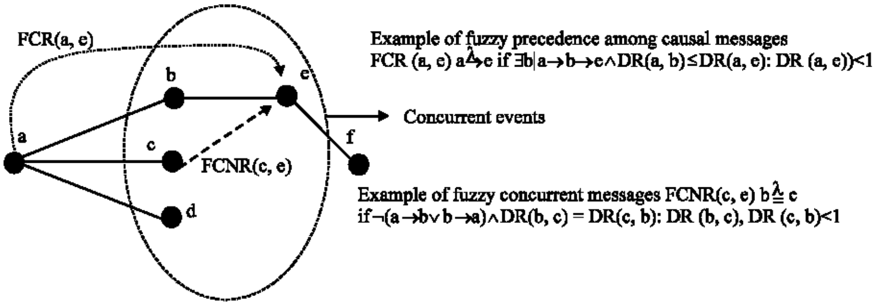


Fig. 1: Example of fuzzy precedence in a distributed system

equal and less than the unit, which is represented as FCNR(a, b). This means, that it can establish 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) are equal and less than one, it means that the event a have 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.

To show the use of the FCR and the FCNR, consider the example given in Fig. 1, which shows a scenario to determine the fuzzy precedence and the fuzzy concurrency between events. For example, for the case of the relation among the events a and e, the FCR(a, e) determine if there exist a cause-effect relation that must be taken into account for the event ordering. For the fuzzy concurrent events e and b, the FCNR(c, e) identify that there is certain spatial and/or temporal relation among them.

**Fuzzy causal consistency:** The Fuzzy Causal Consistency (FCC) is based on the FCR. The goal of the FCC is to indicate how good the performance of the system is in a certain time. The meaning of the performance can be indicated according to the problem to resolve. It is by calculating the value of the FCC that it can be determined if the performance 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 this case are normalized in the interval [0,1].

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

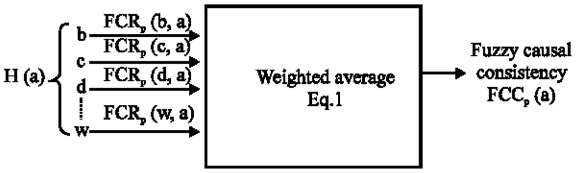


Fig. 2: Fuzzy causal consistency

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

where, GP(b) is a weighting degree used to determine priorities or weight to every fuzzy causal relation when is needed. FCR$_p$(a, b) is the fuzzy causal relation of a pair of events (a, b) at a process p.

## 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 destination of m and m' the delivery of m precedes m' or viceversa, if and only if the performance 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 to satisfy the following condition:

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

$$\forall p \in dests(m) \cap dests(m'),\ FCC_p(m) \leq FCC_{max} \begin{cases} 1.\ delivery_p(m) \rightarrow \\ delivery_p(m') \text{ or} \\ 2.\ delivery_p(m') \rightarrow \\ delivery_p(m) \end{cases}$$

where, $FCC_p(m)$ is the fuzzy causal consistency for the event m at its reception by the participant p and $FCC_{max}$ 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 (performance of the system), for the event m, $FCC_p(m)$, is equal 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 realize a more asynchronous events delivery.

**Fuzzy causal order versus causal order:** Here, it will show the useful of the FRC and the FCC and how to use them in distributed systems. First, it is presented the main differences and advantages among of the fuzzy causal relation versus the happened before relation proposed by Lamport (1978). Next, it is described how the FCR and the FCC can be applied for the concrete problem of intermedia synchronization.

Let to consider 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; see for more details Morales and Pomares (2006).

For the strict causal algorithms based on Lamport's (1978) relation, the delivery of the event $d^-$ implies that the event $a^+$ has been delivered. Owing to delays in the network or lost 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 precedes $d^-$ have been delivered in a causal order or have been discarded because their lifetime have expired

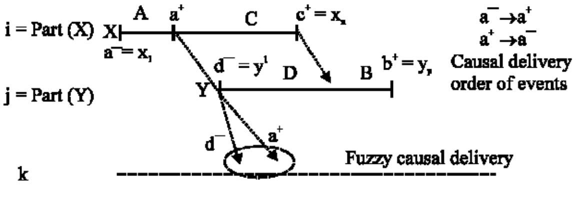


Fig. 3: Example of a distributed multimedia scenario

In 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 (1978) and their main advantage is that the maximum delivery waiting time for the events is determined according to the lifetime established.

In this study, with base 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 by the performance of the system. In the case of the intermedia synchronization problem the performance is linked to the maximum synchronization error allowed.

## APPLYING THE FCR AND THE FCC TO THE INTERMEDIA SYNCHRONIZATION PROBLEM

In order to show the situation of media data out of phase (synchronization error) an 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 function only as Client (Host Z); (Fig. 4). 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 (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 is

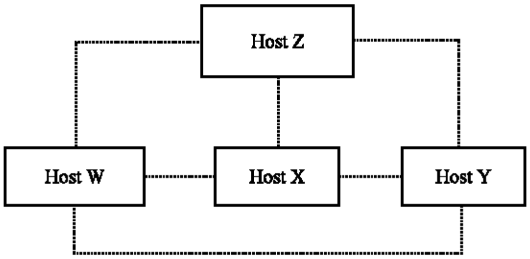


Fig. 4: Example of a distributed multimedia scenario

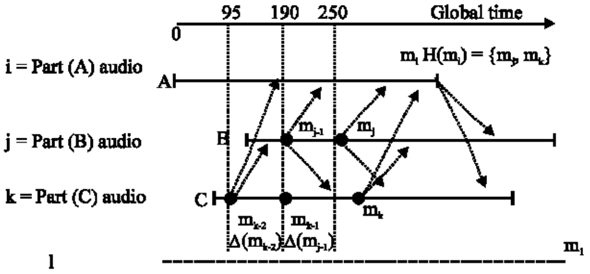


Fig. 5: Example of the fuzzy causal delivery

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. For the remaining audio frames they are sent as fifo messages.

In this scenario, each host has the synchronization mechanism running. The maximum waiting time for every pair of media data is established at $\Delta = 120$ msec, which is the maximum synchronization error established for the reproduction of an audio-audio communication in real time according to Haj and Xue (1997).

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

The FCO condition is evaluated at the reception of the synchronization message $m_i$ by the participant 1. 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 1 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 it has to choose the domains that the relation is going to include. In this study, in order to resolve the event ordering problem it is considered only the logical and temporal domains. For each domain, it is defined one membership function, $R_D$ and $R_N$, respectively. The spatial domain is not included since the audio data do not consider it. On the other hand, the logical domain is considered because, as it was 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. It can be used as distances for the logical domain the separation among the events according to the event ordering chosen, local causal distance (fifo), causal distance introduced by Lopez *et al*. (2005) (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 for the synchronization problem is focus in measure the separation among each media data. For the temporal domain is used as distance the physical time. Next, it will illustrate how the input data for the temporal and logical membership functions are calculated.

**Calculation of the local causal distance for $R_D$:** The input value (local causal distance) of the membership function $R_D$ is calculated at the reception of each synchronization point (begin or end of interval). The $R_d$ (m) measure the local causal distance between each causal message contained in the causal history of m and last message received from each participant contained in its causal history ($last\_message_p$). These values are the input for calculation of the membership function $R_D$, which will be described later. The calculation of $R_d$ is carried out using the following formula:

$$R_d(m') = (m' - last\_message_p)\ \forall m' \in H(m)\ \wedge\ \forall p \in H(m) \quad (2)$$

Here. the values of $R_d(m)$ for the pair of messages $(m_{j-1}, m_j)$ and $(m_{k-2}, m_k)$are:

$$R_d(m_j) = m_j - m_{j-1} = 1 \text{ and } R_d(m_k) = m_k - m_{k-2} = 2$$

**Calculation of the temporal distance for $R_N$**: The input value (temporal distance) of the membership function $R_N$ is calculated also at the reception of each synchronization point. In this work, it is assumed the existence of a global clock to simplify the calculation of the temporal distance; nevertheless, this measure can be realized in a distributed way. The $\Delta(m)$ measure the difference in physical time between the sending time of a message *m* and its arrival to every participant (temporal distance). These values are the input for the calculation of the membership function $R_N$, which will be described subsequently. The calculation of $R_N$ is carried out for the last message received from each participant contained in the causal history of the synchronization point using the following formula:

$$\Delta(m) = time\_arrival\,(m) - time\_sent\,(m) \qquad \forall p \in H(m) \wedge m = last\_message_p \quad (3)$$

Here, the input values for $R_N$ of the messages $m_{j-1}$ and $m_{k-2}$ are:

$$\Delta(m_{j-1}) = 190 - 95 \text{ msec} = 95 \text{ msec and } \Delta(m_{k-2}) = 250 - 190 \text{ msec} = 60 \text{ msec}$$

**Calculation of the fuzzy causal relation:** For the intermedia synchronization problem the value of the FCR for a pair of messages FCR(a, b) is determined from the union of two membership functions, $R_D \cup R_N$, logical and temporal domain, respectively. The fuzzy operator chosen as union operator is the maximum. The FCR is formally defined as:

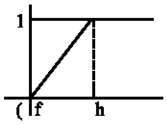
$$FCR_{R_D \cup R_N}(x) = \max\{x_{R_D}(x), x_{R_N}(x)\} \quad (4)$$

The membership functions are normalized in the interval [0,1]. The meaning of the values obtained by the FCR(a, b) is the following one:

- When FCR(a, b) tends to zero indicates that the messages have been delivered with acceptable network conditions. In other words, the delay between the messages is not big, neither its causal distance, which reflects that no considerable changes with regard to the ideal network conditions at the messages delivery time (e.g., there is not data lost)
- When FCR(a, b) tends to the unit indicates that the network conditions are deplorable. This means that the messages lost and/or big delays are present in the network, by showing that the network conditions are far from the ideal, which is reflected by a growing in the causal and temporal distances among the messages

The normalization in the interval [0,1] for the values of the membership functions, $R_N$ and $R_D$ can be calculated by a triangular function defined as:

$$R(x)\begin{cases}0 & \text{if}\quad x<f\\ \dfrac{x-f}{h-h} & \text{if}\quad f\le x<h\\ 1 & \text{if}\quad x\ge h\end{cases} \tag{5}$$



In Fig. 6 and 7 are showed examples of the triangular membership functions $R_N$ and $R_D$, where the cause-effect relation between the temporal distance and the causal distance is show, respectively. The temporal distance among the events is determined in accordance with the maximum delay allowed by the application, also know as synchronization error, this values are related to the type of media involve in the synchronization process (e.g., audio-audio, video-audio, image-audio). For the scenario presented in Fig. 5, a maximum delay of ±120 msec is used for the temporal distance; because according to Haj and Xue (1997) this is the maximum delay for a dialogue among some participants. The logical distance is bounded to four events; this is because it has been shown that in the RTP protocol that the probability of the loss of four events of the same process or participant is minimal (Perkins, 2003).

In order to calculate the FCR(a,b) previously have been obtained the input values of the membership functions $R_N$ and $R_D$. Now it has to normalize their value in the interval [0,1] to determine the input values of the FCR.

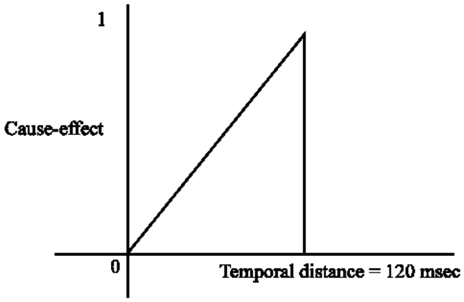


Fig. 6: Temporal cause-effect degree

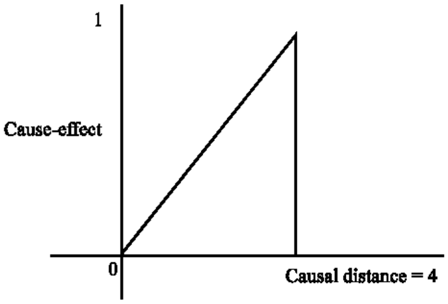


Fig. 7: Logical cause-effect degree

The values for the membership functions $R_N$ and $R_D$ among $m_j$ and $m_{j-1}$ according to the Eq. 5 are:

$R_N = \Delta(m_{j-1})/120$ msec $= 95/120$ msec $= 0.79$ and $R_D = R_d(m_{j-1})/4 = 1/4 = 0.25$

Hence, the value of $FCR(m_{j-1}, m_j) = \max\{R_N, R_D\} = \max\{0.79, 0.25\} = 0.79$.

The values for the membership function $R_N$ and $R_D$ among $m_{k-2}$ and $m_k$ according to the Eq. 5 are:

$R_N = \Delta(m_{k-2})/120 = 60/120$ msec $= 0.5$ and $R_D = R_d/4 = 2/4 = 0.5$

Hence, the value of $FCR(m_{k-2}, m_k) = \max\{R_N, R_D\} = \max\{0.79, 0.25\} = 0.79$.

**Determining the FCC for the FCO:** Once calculated the FCR for each participant contained in the causal history of $m_i$, it is able to calculate the $FCC_1(m_i)$. For simplicity of the calculation of the $FCC_1(m_i)$ in the example it is assumed a GP of one in the Eq. 1, which means that all the messages have the same priority for system.

The value of the $FCC_1(m_i)$ according to the Eq. 1 can be expressed as:

$$FCC_1(m_i) = \frac{FCR(m_{k-2}, m_k) + FCR(m_{j-1}, m_j)}{2} = \frac{0.79 + 0.5}{2} = 0.645$$

**Checking the FCO condition for $m_i$:** Once calculated the value of the $FCC_l(m_i)$ it can check if the $FCO_l(m_i)$ condition is satisfied. In order to evaluate if the FCO is satisfied, the value of $FCC_{max}$ is established as 0.9. This is done by using the ideal conditions of the system. The value was calculated according with the maximum delay tolerated (±120 msec for the case of audio-audio) and the maximum causal distance allowed between a pair of messages (four events according to the RTP protocol).

Returning to the example given, in order to know if $m_i$ can be delivery according to the FCO condition the value of $FCC_l(m_i)$ must to be equal or lower than the $FCC_{max}$ and hence, it is observed that in the example the value of $FCC_l(m_i)$ was 0.645, so the delivery condition is satisfied and the event $m_i$ is delivered; otherwise, if the value of $FCC_l(m_i)$ is larger than $FCC_{max}$ the event cannot be delivered.

## CONCLUSIONS

In this study, it has been presented a new fuzzy causal order for events that occur in a distributed system. The FCO allows to establishing a more asynchronous ordering, which is based in the definitions of fuzzy causal relation and fuzzy causal consistency. The FCR establishes a cause-effect degree between events, by considering spatial, temporal and/or logical domain. The FCR indicates how long ago an event a happened before and event b. On the other hand the FCC indicates how good the performance of the system is in a certain time. Finally, it has been presented an approach of how to apply the FCR, the FCC and the FCO to the intermedia synchronization problem in distributed multimedia systems. Future directions of this study are to address the research to apply the FCR, the FCC and the FCO in order to resolve problems in other areas such as planning and scheduling.

## REFERENCES

Aguilar, J., 2004. Dynamic random fuzzy cognitive maps. J. Comput. Sci. Syst., 7: 260-271.

Baldoni, R., R. Prakash, M. Raynal and M. Singhal, 1998. Efficient ?-causal broadcasting. Int. J. Comput. Syst. Sci. Eng., 13: 263-269.

Birman, K., A. Schiper and P. Stephenson, 1991. Lightweight causal and atomic group multicast. ACM Trans. Compt. Syst., 9: 272-314.

Haj, A. and C. Xue, 1997. Synchronization in multimedia data retrieval. Int. J. Network Manage., 7: 33-62.

Janakiraman, R., M. Waldvogel and L. Xu, 2002. Fuzzycast: Efficient video-on-demand over multicast. IEEE Comput. Commun. Societies, 2: 920-929.

Lamport, L., 1978. Time, clocks and the ordering of events in a distributed system. Commun. ACM., 21: 558-565.

Lopez, E., J. Estudillo, J. Fanchon and S. Pomares, 2005. A fault-tolerant causal broadcast algorithm to be applied to unreliable networks. Proceeding of 17th International Conference on Parallel and Distributed Computing and Systems, Nov. 14-16, Phoenix, AZ. USA., pp: 465-470.

Morales, R.L. and S. Pomares, 2006. A temporal synchronization mechanism for real-time distributed continuous media. Proceedings of the International Conference on Signal Processing and Multimedia Applications, August 2006, Setubal, Portugal, INSTICC Press, pp: 302-309.

Perkins, C., 2003. RTP Audio and Video for Internet. 1st Edn., Addison Wesley, USA., ISBN: 0-672-32249-8.

Silvana, B. and M. Giacomin, 2006. The algebra IAfuz: A framework for qualitative fuzzy temporal reasoning. Artif. Intell., 170: 872-908.

Zhou, Y. and T. Murata, 2001. Modeling and analysis of distributed multimedia synchronization by extended fuzzy-timing petri nets. Trans. Soc. Des. Process. Sci., 5: 23-37.