# Abstract Channels as Connectors for Software Components in Group Communication Services.

Jean Fanchon, Khalil Drira
Laboratory for Analisys and Architecture
of Systems of CNRS (LAAS-CNRS)
7 Av. Colonel Roche,
31077 Toulouse Cedex, France
{fanchon,khalil}@laas.fr

Saul Pomares Hernandez
National Institute of Astrofisics Optics
and Electronics (INAOE)
Luis Enrique Erro #1,
Tonantzintla, Puebla, Mexico.
spomares@inaoep.mx

## Abstract

*Building new services by assembling software components, when adopted at the communication level, would allow developers to build powerful group communication services by assembling proven and efficient algorithms for ordered group communication. The integration of communication protocols with different delivery policies has not been addressed in past research on group communication. This integration cannot be considered as concatenation of protocols and needs delivery policies to be redefined in the context of multi-channel communication. Our paper deals with this issue. We define communication channels as software connectors of the communication level for coordinated group communication. We cover the three standard delivery policies (FIFO order, Causal order, and Total order).*

*Key words. Group Communication, Software Components, Ordering Protocols.*

## 1. Introduction.

In [13], Plazil and Visnovsky predict that: "in the near future, the majority of software applications will be composed from reusable, potentially off-the-shelf software components," and this should allow software engineering to evolve from ad-hoc artisanally implemented modules to industrially designed reliable systems. Defining integration protocols and adaptors for coherent assembling of software components into applications is a key research issue in component-based software engineering [22] and is being investigated by academic and industrial researchers. Serving this new approach of component-based software engineering, recent research effort is being conducted to define and manage the coherence. Most existing works in this area focus on the coherence at the application level, which deals with describing the accepted sequences of service request. These are referred as the protocols of components in [5] and the Object´s protocol in [13]. The coherence is checked by comparing sequences to the set of accepted sequences.

Investigating the design and the associated coherence problems at the communication level is becoming an active research area. This involves problems such as: encapsulating communication protocols in software components [20], synchronization management when integrating multimedia protocols [18], and consistency management when integrating security mechanisms in collaborative applications [2]. Managing the coherence of message ordering is a critical issue in group communication and has not been addressed by past research. It allows unexpected and unsafe non-consistent situations to be avoided, by maintaining a coherent view within the distributed set of components for 1-to-N and N-to-M communication. This means, for example, that receiving "actions" (e.g. requests or questions) are guaranteed to be received before their corresponding "reactions" (e.g. results or responses) by using causally ordered communication. This also means coordinating the distributed decision making processes at the system or at the user levels by totally ordered communication. At the communication level, the coherence management, which focuses on message ordering, is different from its equivalent problem at the application level; it not only complements it, but is also more complex. It includes integrating and simultaneously handling different ordering policies within distributed groups of components. In this paper, we try to explore this new area of interest by defining and managing the coherent integration at the communication level, and by considering three standard group communication message ordering policies. We introduce the "abstract communication channels" as component connectors that coordinate the different ordering policies, thus allowing the coherence to be satisfied. Similarly to the work of [2] and [18], our results should help to design (UDP-based) reliable collaborative applications that support internet-based group collaborative activities.

*The coherence at the communication level.* Each of the delivery policies has an important role in multimedia and distributed systems [1,3,4,6,7,14]. FIFO order ensures that the messages between two participants are delivered in the same sequential order in which they were sent. It is widely spread (TCP-IP) and used to transmit various types of data from text files to continuous multimedia streams, such as audio and video. Causal order [1,11,15,16,17,21] ensures the coordination of group discussion by preventing inconsistencies that may occur when responses are received prior to their questions. Causally ordered communication is used to maintain a consistent view between a group of participants [9]. Total order ensures that all participants in a group receive the messages in the same order. It is often used to ensure a common view between all participants of a cooperative system [3,4,19]. Some works already propose protocols to handle different policies in communication channels, but they restrict all channels to only use one ordering policy per channel [11,15,19]. We claim that the integration of different delivery modes in a single protocol constitutes a basis for a coordination service capable of ensuring an optimal and flexible use of the communication medium, which is adapted to the specific needs of the applications.

*Our contribution.* Our approach consists in the use of channels as abstractions for a subset of messages exchanged by a subset of entity members of this channel. Each basic channel may have a single specific delivery policy, e.g. FIFO, causal or total; and inter-channel delivery properties can be specified on subsets of channels with similar delivery policies. In a first integration of multiple delivery modes (partitioned multimodal communication), each message may be sent into a single channel, with a single delivery policy. This allows components to exchange messages following different policies, but the corresponding sets of messages are disjoint: no delivery relation can be defined between messages sent on two different channels. In order to overcome this restriction, we define coordinated multimodal communication, where messages may be sent into a subset of channels. A message can be sent to any subset of channels to which the sender is connected. It is received by the members of any of these channels, and its delivery to a receiver must be compatible with the delivery policy of each channel shared by the message and the receiver. A single message may have different delivery properties with respect to other messages. In this approach, existing algorithms can be used to ensure the correct delivery for each policy, with their intra and inter-channel coordination; different modules of existing protocols can be merged to construct a multimodal protocol. Our approach is indeed a compatible extension of the current ones, with regard to the delivery policies as well as the channel structure. We show in particular how this approach solves in a simple way the application example 1 below.

In this approach, existing algorithms can be used to ensure the correct delivery for each policy, fifo, causal or total, with either intra and inter-channel coordination; different modules of existing protocols can be merged to construct a multimodal protocol. Our approach is indeed a compatible extention of the current ones from the point of view of the delivery policies as well as the channel structure.

*Example 1:* As an application example, we consider the components involved in an audio and video conferencing system: they exchange video and audio data as well as control information by broadcasting messages form one sender to n receivers. For the whole set of messages denoted $M$, the delivery guarantees the fifo policy between each pair of participants. A subset of messages $M_c \subseteq M$ is furthermore delivered in a causal mode, and another subset of messages $M_t \subseteq M$ (with $M_t \cap M_c = \emptyset$) is delivered in total order. The messages in $M_c$ can be used as periodic global synchronisers, and maintain a relative consistency between the sets of audio/video messages of $M$ delivered to each entity, in a more relax and effective way than if all messages where to be delivered causally. The messages in $M_t$ are received in the same order by each entity, and they may contribute to warranty an identical common view on all the entities by maintaining a replicated state.

*Example 2:* The figures below shows examples of multimodal exchanges between four components. In figure 1, four messages are delivered in a Fifo mode, the messages 1 , 2 and 4 are delivered in a causal mode, while message 3 fails to be delivered in a causal mode w.r.t. messages 1 and 2. In figure 2, all four messages are delivered in a causal mode, concurrent messages 3 and 4 are delivered in a total mode while messages 2 and 3 are not.

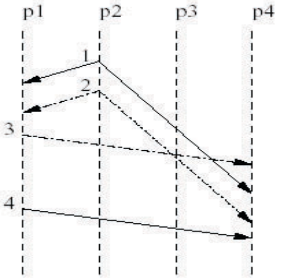

Figure 1

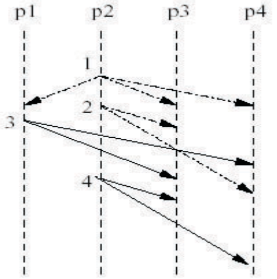

Figure 2

The present paper is structured as follows. In section 2, we define the coordinated behaviours of a set of communicating components, a newly proposed "partial" causal precedence relation induced on the events of a behaviour by a subset of messages and the related delivery policies. In section 3, we present multimodal communication systems, constituted by communicating components and a set of "monomodal" channels. In the section 4, we define interchannel coordination and interchannel delivery policies. In section 5 we show how a multimodal protocol can be specified from a set of existing protocols, one for each of "elementary" delivery policies.

## 2. Background and definitions.

In the sequel we consider a set of sequential components interfaced with a communication protocol through which they can exchange messages. The interactions of the components with the protocol are called events, and they are occurrences of sendings and deliveries of messages. A coordinated behaviour describes a partial ordering between these events during a run of the application. This partial order is often called causal precedence or happened-before relation [8,9]. In these behaviours, unlike in Message Sequence Charts, messages have a single source but may have any number of destinations and this is the main difference (besides minor technicalities) with MSCs. A delivery policy between a set of components is specified by a property of their behaviours behaviour: the components communicate following a particular policy if and only if all of their coordinated behaviours satisfy the corresponding property.

### 2.1. Components, messages and behaviours.

We denote $I = \{i, j...\}$ the set of integers used as components identifiers. A *finite coordinated behaviour* involving the components $I$ is a tuple $u = (M, Src, Dest, E, \leq)$ where:

- $M = \{m, m'..\}$ is a finite set of exchanged messages.
- $Src$ and $Dest$ are two mappings $Src : M \longrightarrow I$ and $Dest : M \longrightarrow 2^I$ defining the source and the set of destinations of each message. The set of messages sent by the component $i$ is denoted $M_i = \{m \in M, Src(m) = i\}$, and we have $M = \cup_{i \in I} M_i$.
- $E$ is a finite set of communication events defined by $E = \{send(m),\ m \in M\}$ $\cup \{deliver(j, m),\ m \in M,\ j \in Dest(m)\}$. The event $send(m)$ denotes the sending of the message $m$ by the component $i = Src(m)$ and is also written $send(i, m, Dest(m))$ or $send(i, m)$. The event $deliver(j, m)$ denotes the delivery of the message $m$ to the component $j \in Dest(m)$.
- $\leq$ is a partial order relation (reflexive, transitive, acyclic) on the set of events, $\leq \subseteq E \times E$. The relation $\leq$ is minimal among the partial orders satisfying the following properties:

1. For any identifier $i \in I$, the set $E_i \subseteq E$ of events involving a component $i \in I$, defined by $E_i = \{send(i, m), m \in M_i\} \cup \{deliver(i, m), i \in Dest(m)\}$, is totally ordered by $\leq$. In particular for any pair of messages $m, m' \in M_i$, we have either $send(i, m) \leq send(i, m')$ or $send(i, m') \leq send(i, m)$.

2. For any message $m \in M$ and any component $j \in Dest(m)$, we have $send(m) \leq deliver(j, m)$.

### 2.2. Partial causal precedence.

We consider a subset $M' \subseteq M$ of the messages of a behaviour $u = (M, Src, Dest, E, \leq)$, and we define the *partial causal precedence induced by $M'$* , denoted $\leq_{M'}$. This relation is defined on the set of events $E' \subseteq E$ denoting sendings or deliveries of the messages belonging to $M'$, i.e. $E' = \{send(m) : m \in M'\} \cup \{deliver(i, m) : m \in M' \bigwedge i \in Dest(m)\}$. For any identifier $i \in I$, we let $E'_i = E' \cap E_i$ be the subset of events involving $i$ and some message belonging to $M'$. The partial precedence $\leq_{M'} \subseteq E' \times E'$ induced by $M'$ is the least partial order relation on $E'$ satisfying the following properties:

1. for any component $i \in I$, the "local" restrictions of $\leq_{M'}$ and $\leq$ to the events of $E'_i$ coincide: $\forall e, e' \in E'_i : e \leq e' \iff e \leq_{M'} e'$, which can be written $\leq_{M'} /_{E'_i \times E'_i} = \leq /_{E'_i \times E'_i}$.
2. for any message $m \in M'$ and $j \in Dest(m)$, we have $send(m) \leq_{M'} deliver(j, m)$.

It is a direct consequence of the definitions that the partial causal precedence is included in the "global" one, for any pair of events $e, e' \in E'$ we have $e \leq_{M'} e' \Longrightarrow e \leq e'$, and we have $\leq_{M'} \subseteq \leq \ /_{E' \times E'}$. However this inclusion is strict in general, and the partial causal precedence is not the restriction of the "global" causal precedence $\leq$ to $E'$ as shown in example 3 and figure 3.

*Notations:* $m \prec m'$ and $m \prec_{M'} m'$ are sometimes used as shortcuts for $send(m) \leq send(m')$ and $send(m) \leq_{M'} send(m')$.

### 2.3. Delivery modes.

Usually delivery modes are defined for all the messages of a behaviour. In multimodal communication, a delivery mode may be defined for any subset of messages. Let $M' \subseteq M$ a subset of the messages of the behaviour $u = (M, Src, Dest, E, \leq)$, then

*Fifo* The messages of $M'$ are delivered in a fifo mode iff for all $i \in I$, $m, m' \in M' \cap M_i$:
$send(i, m) \leq send(i, m') \implies \forall j \in Dest(m) \cap Dest(m') : deliver(j, m) \leq deliver(j, m')$

*Causal* Using the partial causal precedence related to $M'$, we define the causal delivery property as follows.

The messages of $M'$ are delivered in a causal mode iff for all $i, j \in I$, $m \in M' \cap M_i$, $m$' $\in M' \cap M_j$:
$send(i, m) \leq_{M'} send(j, m') \implies \forall k \in Dest(m) \cap Dest(m') : deliver(k, m) \leq deliver(k, m')$

*Total* The messages of $M'$ are delivered in a total mode iff for all $m, m' \in M'$, $i, j \in Dest(m) \cap Dest(m')$:
$deliver(i, m) \leq deliver(i, m') \implies deliver(j, m) \leq deliver(j, m')$

The use of the partial precedence relation $\leq_{M'}$ in the definition of the causal delivery is commented in detail in example 3 at the end of section 3.2.

## 3. Channels and Multimodal Communication Systems.

In order to define different relations on the messages of a behaviour, we first define the notion of channel. An elementary channel is an abstraction that represents a subset of the messages exchanged by a subset of components connected to this channel and called the members of the channel. Furthermore these messages are delivered following a specific mode of the channel to all the members of the channel. Each channel is characterised by its members and its delivery mode.

A *multimodal communication system* ( MCS) is a tuple $S = (I, C, Memb, Mode)$ where

- $I$ is a finite set of components
- $C$ is a finite set of channels
- $Memb$ is a mapping $Memb : C \longrightarrow 2^I$ defining for each channel the set of connected components and
- $Mode$ is a mapping $Mode : C \longrightarrow \{free, fifo, causal, total\}$ defining for each channel a delivery policy

### 3.1. Partitionned Multimodal Communication.

In a partitionned multimodal communication, each message may be sent into a single channel, with a single delivery policy. This allows components to exchange messages following different policies, but the corresponding sets of messages are disjoint: no delivery relation can be defined between messages sent on two different channels.

A coordinated behaviour of the system $S = (I, C, Memb, Mode)$ is a tuple $u = (M, Src, Dest, Chan, E, \leq)$ where

- $(M, Src, Dest, E, \leq)$ is a behaviour as defined in section 2
- $Chan$ is a mapping $Chan : M \longrightarrow C$ defining for each message the channel on which it is emitted. The source of a message $m$ is a member of $Chan(m)$, $\forall m \in M : Src(m) \in Memb(Chan(m))$. Let $c = Chan(m)$ the set of destinations $Dest(m)$ and the set of connected components $Memb(c)$ coincide: $\forall m \in M : Dest(m) = Memb(Chan(m)) - \{Src(m)\}$. The set of messages emitted on a channel $c$ is denoted $M(c)$, $M(c) = \{m \in M | c = Chan(m)\}$.
- The sending event $send(m)$ of a message $m$ by the component $i = Src(m)$ on the channel $c = Chan(m)$ is also denoted $send(i, m, c)$ or $send(m, c)$. The set of events on a channel $c$ is $E(c) = \cup\{send(m),\ deliver(j, m) : m \in M(c),\ j \in Memb(c)\}$, and we have $E = \cup_{c \in C} E(c)$.

The coordinated behaviour $u = (M, Src, Dest, Chan, E, \leq)$ must furthermore satisfy the following properties:

*Fifo* For any channel $c$ such that $Mode(c) = fifo$, the messages of $M(c)$ are delivered in fifo mode:
for all $i \in Memb(c)$, $m, m' \in M(c)$,
$send(i, m, c) \leq send(i, m', c) \implies \forall j \in Mem(c) : deliver(j, m) \leq deliver(j, m')$

*Causal* For any channel $c$ such that $Mode(c) = causal$, the messages of $M(c)$ are delivered in causal mode:
for all $i$ , $j \in Memb(c)$, $m, m' \in M(c)$,
$send(i, m, c) \leq_{M(c)} send(j, m', c) \implies \forall k \in Mem(c) : deliver(k, m) \leq deliver(k, m')$
note that we use the partial causal precedence $\leq_{M(c)}$ induced by the messages of the channel $c$ (see example 3 below)

*Total* For any channel $c$ such that $Mode(c) = total$, , the messages of $M(c)$ are delivered in total mode:
for all $m, m' \in M(c)$, $i, j \in Memb(c)$,
$deliver(i, m) \leq deliver(i, m') \implies deliver(j, m) \leq deliver(j, m')$

*Example 3:* The behaviour depicted by figure 3 below shows why the partial dependency relation $\leq_c$ ( resp $\leq_{M'}$ in the previous section) is used instead of the global one $\leq$ in the causal channel definition. In this scenario let $c$ be a channel with four messages, $M(c) = \{1, 3, 4, 5\}$. We have $1 \prec 3$, $1 \prec 4$ and $1 \prec 5$, but neither $1 \prec_{M(c)} 3$, nor $1 \prec_{M(c)} 4$ nor $1 \prec_{M(c)} 5$. If the causal delivery property on the channel $c$ was defined using $\prec$, only the messages $4$ and $5$ would satisfy this property w.r.t. message $1$, and this could only be achieved by letting the message $2$, which does not belong to $M(c)$, carry the necessary causal information about the channel $c$. If the message $2$ also belonged to $M(c)$, then the delivery of message $4$ would fail to be causal. The causal delivery of $5$ w.r.t. $1$ could be realised as a side effect if another causal channel $c'$ was defined with $M(c') = \{2, 5\}$.

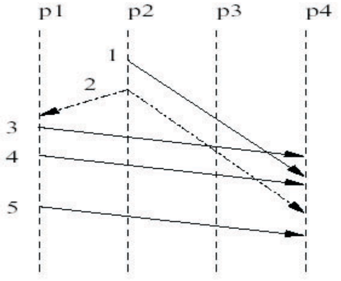

Figure 3

## 3.2. Coordinated Multimodal Communication.

We extend now the communication capabilities: In the previous section, no delivery relation could be defined between messages sent on two different channels. In order to overcome this restriction, we allow messages to be sent into a subset of channels. A message can be sent to any subset of the channels to which the sender is connected, it is received by the members of any of these channels, and its delivery to a receiver must be compatible with the delivery policy of each channel shared by the message and the receiver. Let us stress that a message is delivered only once to a single receiver, even if it has been sent on many of the channels the receiver is connected to. We consider a CCS $S = (I, C, Memb, Mode)$.

A multimodal coordinated behaviour of the system $S$ is a tuple $u = (M, Src, Dest, Chan, E, \leq)$ where

- $(M, Src, Dest, E, \leq)$ is defined as in section 2
- $Chan$ is a mapping $Chan : M \longrightarrow 2^C$ defining for each message the set of channels on which it is emitted. The source of a message $m$ must belong to all the channels of $Chan(m)$, i.e. $Src(m) \in \cap_{c \in Chan(m)} Memb(c)$. The set of destinations $Dest(m)$ and the set of connected components to any channel of $Chan(m)$ coincide: $Dest(m) = \cup_{c \in Chan(m)} Memb(c) - \{Src(m)\}$ . The set of messages emitted on a channel $c$ is denoted $M(c)$, $M(c) = \{m \in M | c \in Chan(m)\}$.
- The sending event $send(m)$ of a message $m$ by the component $i = Src(m)$ on the set of channels $C' = Chan(m)$ can be denoted $send(i, m, C')$ or $send(m, C')$. The set of events on a channel $c$ is $E(c) = \{send(m),\ m \in M(c)\} \cup \{deliver(j, m) : m \in M(c),\ j \in Dest(m)\}$, and we have $E = \cup_{c \in C} E(c)$.

The coordinated behaviour $u = (M, Src, Dest, Chan, E, \leq$ ) must furthermore satisfy the following properties:

for any channels $c, c_1, c_2$ and any sets of channels $C_1, C_2 \subseteq C$

*Fifo* If $c \in C_1 \cap C_2$ with $Mode(c) = fifo$, and $i \in Mem(c)$, then:

$send(i, m, C_1) \leq send(i, m', C_2) \Longrightarrow \forall j \in Mem(c) : deliver(j, m) \leq deliver(j, m')$

*Causal* If $c \in C_1 \cap C_2$ with $Mode(c) = causal$, and $i, j \in Mem(c)$, then:

$send(i, m, C_1) \leq_{M(c)} send(j, m', C_2) \Longrightarrow \forall j \in Mem(c) : deliver(j, m) \leq deliver(j, m')$

note that as in previous section we use the partial causal precedence $\leq_{M(c)}$ induced by the messages of the channel $c$.

*Total* If $Mode(c) = total$, and $m, m' \in M(c)$, then for all $i, j \in Mem(c)$:

$deliver(i, m) \leq deliver(i, m') \Longrightarrow deliver(j, m) \leq deliver(j, m')$

*Example 4.* The example depicted by figure 1 in the introduction describes a scenario which can be realised with two channels, one has fifo delivery and contains all the four messages, the second has causal delivery and only contains the messages 1 and 4. The scenario depicted by figure 2 can be realised with two channels, one has causal delivery and contains all the four messages, the second has total delivery and only contains the messages 3 and $4$

## 4. Multimodal Interchannel Coordination.

We extend now the capabilities of a multimodal coordinated communication by allowing interchannel coordination [11,14,15,19]: interchannel coordination rules the delivery of messages which belong to two or more elementary channels.

An extended multimodal communication system is a tuple $S = (I, C, Memb, Mode, \Omega, Ichan, Imode)$ where

- $(I, C, Memb, Mode)$ is defined as in section 2.
- $\Omega$ is a finite set of multichannel names.
- $Ichan$ and $Imode$ are two mappings $Ichan : \Omega \longrightarrow 2^C$, and $Imode : \Omega \longrightarrow \{fifo, causal, total\}$, which define for each multichannel the corresponding set of channels and their interchannel coordination mode.
- The mappings $Mode$ and $Imode$ satisfy the following consistency property: for any channel $c \in C$ and multichannel $\gamma \in \Omega$, $c \in Ichan(\gamma) \implies Mode(c) = Imode(\gamma)$

For a multichannel $\gamma \in \Omega$ we let $M(\gamma) = \cup\{M(c), c \in Ichan(\gamma)\}$ be the set of messages of $\gamma$.

A coordinated behaviour of the system $S = (I, C, Memb, Mode, \Omega, Ichan, Imode)$ is a tuple $u = (M, Src, Dest, Chan, E, \leq)$ defined as in section 2.3 but which satisfies furthermore the following interchannel delivery properties : for any channels $c, c_1, c_2$ any sets of channels $C_1, C_2 \subseteq C$ and any multichannel $\gamma \in \Omega$

*Fifo* If $Imode(\gamma) = fifo$, $c_1 \in C_1, c_2 \in C_2$, $\{c_1, c_2\} \subseteq Ichan(\gamma)$ and $i \in Mem(c_1) \cap Mem(c_2)$, then:

$send(i, m, C_1) \leq send(i, m', C_2) \implies \forall j \in Mem(c_1) \cap Mem(c_2) : deliver(j, m) \leq deliver(j, m')$

*Causal* If $Imode(\gamma) = causal$, $c_1 \in C_1, c_2 \in C_2$, $\{c_1, c_2\} \subseteq Ichan(\gamma)$ and $i, j \in Mem(c_1) \cap Mem(c_2)$, then:

$send(i, m, C_1) \leq_{M(\gamma)} send(j, m', C_2) \implies \forall k \in Mem(c_1) \cap Mem(c_2) : deliver(k, m) \leq deliver(k, m')$

where we note $\leq_{M(\gamma)}$ the partial causal precedence induced by the set of messages $M(\gamma) = \cup_{c \in Ichan(\gamma)} M(c)$.

*Total* If $Imode(\gamma) = total$, $c_1 \in C_1, c_2 \in C_2$, $\{c_1, c_2\} \subseteq Ichan(\gamma)$ $m \in M(c_1)$, $m' \in M(c_2)$ and $i, j \in Mem(c_1) \cup Mem(c_2)$, then :

$deliver(i, m) \leq deliver(i, m') \implies deliver(j, m) \leq deliver(j, m')$

In each definition above, taking $c_1 = c_2$, we see that $c$ is a fifo (resp. causal, resp. total) channel whenever it belongs to a multichannel $\gamma$ with inter-channel fifo (resp. causal, resp. total) delivery, and this justifies the consistency property required by the definition.

*The solution of the Audio-video conferencing example.* We consider again the example 1 presented in the introduction, where a group of participants of a video-conference exchange in fifo mode a set of messages $M$, where a subset of messages $M_c \subseteq M$ must be delivered in a causal mode, and another subset $M_t \subseteq M$ with $M_t \cap M_c = \emptyset$ must be delivered conforming to a total order policy. We also suppose that only a subset of participants with identifiers $J \subseteq I$ exchange the messages of $M_t$. Clearly a causal order channel $c_c$ with $I = Memb(c_c)$ and $M(c_c) = M_c$ and a total order channel $c_t$ with $J = Memb(c_t)$ and $M(c_t) = M_t$ must be defined ($Mode(c_c) = causal$ and $Mode(c_t) = total$). Furthermore the messages belonging to each of these channels must be delivered in fifo mode, which is implicit for the messages of $M(c_c) = M_c$, in particular all the messages $M_t \cup M_c$ sent by the components in $J$. This can only be done by defining two fifo channels $c_1$, $c_2$ such that $I = Memb(c_1)$ and $J = Memb(c_2)$ and such that $c_1$ and $c_2$ have interchannel fifo delivery, i.e. for some $\gamma \in \Omega$, we have $\{c_1, c_2\} = Ichan(\gamma)$ and $Imode(\gamma) = fifo$. The required properties will be satisfied if the messages of each channel are the following: $M(c_1) = M - M_t$, $M(c_2) = M_t$, $M(c_c) = M_c$, and $M(c_t) = M_t$. This can be achieved if in the application codes, the messages $M - (M_c \cup M_t)$ are sent on $c_1$, the messages of $M_c$ are sent on $\{c_1, c_c\}$, and the messages $M_t$ are sent on $\{c_2, c_t\}$.

## 5. Multimodal Protocols.

### 5.1. Communication protocols.

A standart protocol component maintains a local control information and is interfaced with the underlying network by the method calls $send\_to\_net$ and $receive\_from\_net$. Upon a $send$ call by the application, the protocol component builds the control information to be sent with the message, effectively multicasts the message and the attached control information to the recipients by a $send\_to\_net$ call, and updates its local control information. Upon reception of a message, an occurrence of $receive\_from\_net$, the receiving protocol component tests the control information attached to the message w.r.t. its local control information: if the required properties for the delivery are satisfied (e.g. fifo, causal or total), the message is delivered to the application by the $deliver$ call and the local control information of the protocol is updated. Otherwise the message is buffered and its delivery to the application postponed to the delivery of another message which enables it by the induced modification of the local control information.

### 5.2. Merging protocols to ensure multimodal communication.

A multimodal protocol can be realised by merging monomodal ones in the following way: we suppose a communication system $(I, C, Memb, Mode)$ communicating through a set of channels and a set of multichannels, each of them with a specific delivery policy ensured by a specific protocol. We let each local component $i$ of the multimodal protocol maintain a local protocol control information in an array $LocalPI(i)$, where, for each channel $c$ to which $i$ is connected, $LocalPI(i)(c)$ hold the current values for the monomodal protocol in charge $c$ . When $i$ sends a message $m$ on a set of channels $C' = Chan(m)$ by a call $send(i, m, C')$ (which means that $i$ is connected to each $c \in C'$), the local protocol component builds a control information array $MessagePI(m)$ sent with $m$, with for each $c \in C'$ a field $MessagePI(m)(c)$. Furthermore the new value of $LocalPI(i)(c)$ is computed when the call to $send\_to\_net(m, MessagePI(m), C')$ is done. Upon the reception of the message $m$ by a component $j$ belonging to $Dest(m) = \cup_{\mathsf{c}\in\mathsf{C}'} Mem(c)$, for each channel $c \in C'$ such that $j \in Memb(c)$, the guard of the associated monomodal protocol is evaluated, which depends on $LocalPI(j)(c)$ and $MessagePI(m)(c)$. If all these guards are evaluated to true, the message is delivered through the event $deliver(i, m)$, and this delivery satisfies the policies of all the channels and multichannels $m$ belongs to. New values of $LocalPI(j)(c)$ are computed. Otherwise the delivery of $m$ is delayed until all the guards are satisfied. The amount of control information sent with the messages is at most the sum of the amounts necessary to each protocols.

```
On $send(m, Chan(m))$ by a component $i = Src(m)$
such that $i \in \cap_{c\in Chan(m)} Memb(c)$:
    For all $c \in Chan(m)$ do
        $MessagePI(m)(c) = MakeInfo(LocalPI(i)(c))$
    Execute $send\_to\_net(m, MessagePI(m), Chan(m))$
    For each $c \in Chan(m)$ do
        $LocalPI(i)(c) = OutUpdatePI(i)(c)$
On $receive\_from\_net(m, MessagePI(m), Chan(m))$ by a component $j$
such that $j \in \cup_{c\in Chan(m)} Memb(c)$:
    $InChan(j)(m) = \{c \in Chan(m) | j \in Memb(c)\}$
    For all $c \in InChan(j)(m)$ do
        $DeliverBool(j)(c)(m) = EvalGuard(c)(LocalPI(j)(c), MessagePI(m)(c))$
    If $\bigwedge_{c\in InChan(j)(m)} DeliverBool(c)(m) = True$ then
        $deliver(j, m)$,
        $LocalPI(j)(c) = InUpdatePI(c)(LocalPI(j)(c), MessagePI(m)(c))$.
```

## 6. Conclusion

We have presented a framework which extends in a significant way the coordination capabilities of a set of distributed entities. It extends the various delivery policies in a very flexible and modular way. It covers a wide range of concrete situations in cooperative computing, and should lead to useful theoretical and software developments.

## References

[1] F.Adelstein, M. Singhal. *Real-Time Causal Message Ordering in Multimedia Systems*, Journal of Telecommunication Systems, Modeling, Analysis, Design, and Management, No. 7, , pp. 59-74, July 1997

[2] Y. Amir, G. Ateniese, D. Hasse, K. Yongdae, C. Nita-Rotaru, T. Schlossnagle, J. Schultz, J. Stanton, G. Tsudik. *Secure group communication in asynchronous networks with failures: integration and experiments* , Proceedings of the 20th International Conference on Distributed Computing Systems, 10-13 April 2000 Page(s): 330 -343

[3] Y. Amir, L. Moser, P. Melliar-Smith, D Agarwal, and P Ciarfella. *The totem single-ring ordering and membership protocol*, ACM Trans. Compt. Syst., vol 13, No 4, pp. 311-342, 1995

[4] X. Chen, L.E. Moser, P.M. Melliar-Smith. *Totally Ordered Gigabit Multicasting*, Journal Distributed System Engineering vol. 4 pp 229-243, 1997.

[5] A. Farias, Y. Gueheneuc. *On the coherence of component protocols*, Electronic Notes in Theoretical Computer Science 82 No. 5 (2003) URL: http://www.elsevier.nl/locate/entcs/volume82.html \

[6] T. Gayraud, P. Berthou, M. Diaz. *M3POC: A Multimedia Multicast Transport Protocol for Cooperative Applications*, IEEE International Conference on Multimedia (ICME 2000), New York (USA), Juillet , 2000

[7] Kaashoek, M.F., Tanenbaum, A.S., Verstoep, K. *"Group Communication in Amoeba and its Applications,"* Distributed Systems Engineering Journal, vol 1, pp. 48-58, July 1993.

[8] L.Lamport. *Time,Clocks and the Ordering of events in Distributed Systems*. Communications ACM 21 (7) 1978.

[9] F.Mattern. *Virtual Time and Global States of Distributed systems*. Parallel and Distributed Algorithms, North-Holland 1989.

[10] M. Molteni, M Villari. *Using SCTP with Partial Reliability for MPEG-4 Multimedia Streaming*, in Proc of 2nd European BSD Conference, Amsterdan, Netherlands, 2002.

[11] A.Mostefaoui, M.Raynal. *Causal multicast in overlapping groups: towards a low cost approach.* IEEE ACM Trans. on Networking, Vol 6, N 5 October 1985.

[12] M. Nakamura, Y. Kakuda, T. Kikuno. *An integration-oriented approach for designing communication protocols from component-based service specifications*, INFOCOM '96. Fifteenth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation. Proceedings IEEE , Volume: 3 , 24-28 March 1996 Page(s): 1157 -1164 vol.3

[13] F. Plasil, S. Visnovsky. *Behavior protocols for software components*, in: IEEE Transactions on Software Engineering VOL. 28, NO. 11, NOVEMBER 2002.

[14] S.Pomares Hernandez. *Services de coordination et protocoles de diffusion causale pour les applications cooperatives distribuees*. These LAAS-CNRS novembre 2002. RR LAAS n 02332.

[15] S.Pomares Hernandez, K.Drira, J.Fanchon, M.Diaz. *An efficient multi-channel distributed coordination protocol for collaborative engineering activities*. 2002 IEEE International Conference on Systems Man and Cybernetics (SMC'02), Hammamet (Tunisie), 6-9 Octobre 2002, 6p.

[16] S.Pomares Hernandez, J.Fanchon, K. Drira, M.Diaz. *Causal broadcast protocol for very large group communication systems*. 5th International Conference on Principles of DIstributed Systems (OPODIS'2001), Manzanillo (Mexique), 10-12 Décembre 2001, 9p.

[17] L. Rodrigues, R. Baldoni, E. Anceaume and M. Raynal. *Deadline-Constrained Causal Order,* In The 3rd IEEE International Symposium on Object-oriented Real-time distributed Computing, Mar. 2000.

[18] S. Sakata, T. Ueda, *Real-time desktop conference system based on integrated group communication protocols*, Seventh Annual International Phoenix Conference on Computers and Communications, 16-18 March 1988 Page(s): 379 -384.

[19] A.Shiper, R.Guerraoui. *Total order multicast to multiple groups*. Proceedings of the 17th International Conference on Distributed Computing Systems (ICDCS-17)

[20] C. Sibertin-Blanc, C. Hanachi, J. Cardoso. *Communication protocols as first-class components of multiagent systems*, Fourth International Conference on MultiAgent Systems, 10-12 July 2000 Page(s): 437 -438

[21] T. Tachikawa, M. Takizawa. *Causality and A-Delivery for Wide-Area Group Communications*, Computer Communications Journal, Vol. 23, No. 1, 13–21, 2000.

[22] D. M. Yellin, R. E. Strom. *Protocol specifications and component adaptors,* ACM TOPLAS 19 (1997), pp. 292-333.