User-aware Videoconference Session Control using Software Agents*

Juan A. Botía, Pedro Ruiz and Antonio F. Gómez-Skarmeta
Departamento de Ingeniería de la Información y las Comunicaciones
Universidad de Murcia
{juanbot, skarmeta}@um.es, pedrom@dif.um.es

Abstract

This work is focused in the proposal of a new approach based on intelligent agents for the initiation, control and adaptation of a multimedia session. Typically, multimedia sessions are managed through the SIP (Session Initiation Protocol) Internet protocol. When the session is negotiated, very little is said at the SIP specification about how to decide on which configuration, among the possible ones, is the most suitable for the agents playing the protocol interaction. In this paper, we propose a negotiation mechanism based on the monotonic concession protocol to obtain an optimal joint utility for both peers.

1 Introduction

With respect to end to end videoconferencing, it is obvious that users may have their own preferences on media to be used in a session: some users may prefer a video of poor quality but with a high frame refresh rate, other users may prefer a good sound quality rather than a high quality video representation. This fact should have an effective impact on the negotiation of the videoconference session parameters before the session starts. This paper focuses on taking into account user preferences about videoconference media, in the negotiation of the session parameters.

If we talk about multimedia session management, we have to refer to SIP. From among the Internet protocols which work at the application level, SIP (Session Initiation Protocol) [3] describes how to perform the whole management. The main idea of this paper is to reuse the key concepts and procedures of SIP, which work at the application level over a transport protocol like, for example, TCP, and to redefine them at the agent level. The reasons to redefine SIP and not simply reusing any existing implementation are three. The first one is that the complexity of SIP is considerable and we only need a small part of it. The second is that FIPA agents allow us to locate mobile users easily by means of the agent directory service which is a integrating part of the standard and we do not need the naming scheme and the proxy mechanisms of SIP for that. Finally, the third is that SIP says nothing about negotiation of the multimedia session parameters, an issue for which agents are specially suited [4, 7].

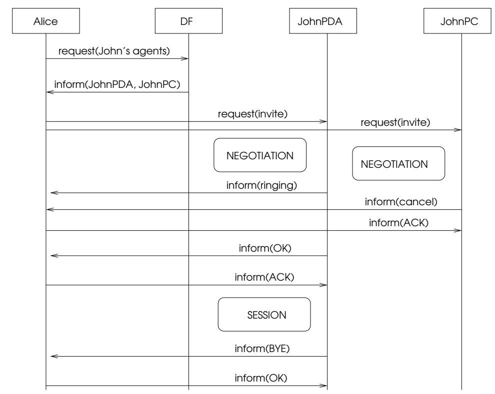


Figure 1. Approach to SIP from a FIPA application

We will describe our approach by using an example. Alice is represented at the system by its corresponding FIPA user agent running on a PDA with LEAP [1] (a FIPA compliant platform capable or running in PDAs). The PDA has a videoconference software with a GSM audio codec and a MJPEG video codec. She wants to communicate with John through an audio/video session. John is also represented at the system by two different agents, the first is similar to that of Alice, and the second is running on his desktop computer. A possible sequence diagram including the exchange of FIPA ACL messages appears depicted at figure 1. In this case, the directory facilitator (DF) will inform the user agent of Alice about the different possibilities it has to

*Part of this work has been funded by the Spanish Science and Technology Ministry (MCYT) by means of the 'Ramon y Cajal' workprogramme as well as the *SAM (TIC2002-04531-C04-03)* project. We want also to thank the Spanish Transports Ministry for its support to our research.

connect with John's user agent. John is associated to each agent which represent him at the system. In this case, as he is in his office, he can be contacted through his personal computer. Also, as he always carry his PDA with him, this represent another possibility for connection. Then, the DF will return the two possible connections. From this point, a negotiation process will be initiated between Alice's user agent and the other two from John. In this particular example, a connection between the two PDAs is made. Notice that the body of the FIPA messages are SIP commands.

The INVITE message is conveyed by a `fipa-inform` FIPA communicative act. The main fields of the FIPA message appears in the following example:

```
(inform
  :sender (agent-identifier :name AlicePDA@jade.um.es)
  :receiver
    (set (agent-identifier :name JohnPDA@jade.um.es)
    (agent-identifier :name JohnPC@jade.um.es))
  :protocol sip-um
  :ontology sdp-ontology
  :content
    <sip>
      <head>
        <method name=''INVITE''
          contact=''AlicePDA@jade.um.es''/>
      </head>
      <session>
        [descriptive content of the SDP session]
      </session>
    </sip>
  :language xml)
```

Due to the absence of proxies which should redirect messages and also include conversation control fields at the agent message, the rest of SIP fields are not needed. If we now represent the SDP session Alice proposes by means of AlicePDA, in which it requires to use GSM audio and MJPEG video, transmitted with RTP (i.e. a real time transport protocol) we have the following SDP description, translated into XML:

```
<head>
  <sdp:origin username="alicia" session-id="2890844526"
    version="2890842807" network-type="IN"
    address-type="IP4" address="155.54.12.11"/>
    <meta name="title" content="Talk with John" />
    <sdp:info xml:lang="en">
      A simple talk with John to ask him for a professional
      issue from my PDA
    </sdp:info>
    <sdp:email>alicia@um.es </sdp:email>
</head>
<body>
  <audio src="155.54.12.11" sdp:port="49170"
    sdp:transport="RPC/AVP" sdp:fmt-list="GSM">
  <video src="155.54.12.11" sdp:port="51372"
    sdp:transport="RTP/AVP" sdp:fmt-list="MJPEG">
  <sdp:attribute attribute-name="sendrecv"/>
</body>
```

The main reason for using XML in the content of messages is openness. By using this, we leave the door open to real SIP agents to communicate with ours through an ad hoc gateway. XML only defines the syntax, but the semantics can be found in the ontology `sdp-ontology` (see the `inform` message from above).

When both agents of John receive the message, the negotiation process starts. Once it is finished, the winning agent (lets say JohnPDA, for example) will send an `inform(ringing)` and the looser (JohnPC) an `inform(cancel)`. The rest of messages are very simple.

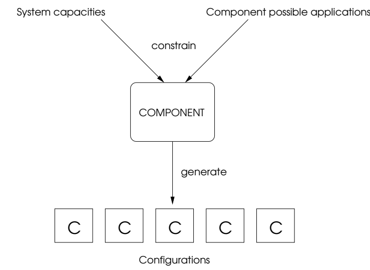


Figure 2. Obtaining configurations from the components and system capacities

For example, if JohnPDA agrees with the session kind offered, it will respond with an `inform` containing a SIP `OK` message. In the other hand, the user agent representing John at his personal computer, will send a DECLINE SIP command again into an `inform`.

2 The negotiation process

When stablishing a session between two or more users, each will join it with a concrete terminal. Each terminal in turn will be characterized by the hardware and software resources it needs. Resources define the limitations about what, and what not, can be visualized, printed and captured with the terminal (or device). **System capacities** are defined as all the features offered and limited by the system. Moreover, when the session is going to be established, the different media that will be used have to be specified. **Session Components** are the different communication elements that will be used. This paper considers only audio and video components. A component is defined by (a) the functionality it provides and (b) one of more ways to provide the functionality. Each one of the different ways of providing functionality is named a **configuration**. The relation between the three concepts is represented at figure 2. The negotiation process generates two different kinds of configurations: potential and real configurations. A potential configuration is the set of possible configurations which can be adopted by the system. A real configuration is a concrete instantiation of one of the potential ones.

When designing the negotiation mechanism for session establishment, two different problems must be solved in turn (a) the design of a mechanism by means of which the user can specify preferences on audio and video and (b) the definition of a protocol and negotiation strategy to be used by the agents implied in the determination of the real configuration the session will have meanwhile reaching an acceptable satisfaction level for the set of agents. We dedicate the rest of the section to the second problem.

2.1 Negotiating a session configuration

In this point, we should make us the following question: it is really worth making a negotiation? Would not be enough to exchange the valuation of the agents for each possible point of negotiation (i.e. each possible multimedia session actual configuration) and seek the better solution for both? The solution would simply be the configuration for with the sum of utilities of both agents is highest, if we use the criteria of social welfare as the measure for a good negotiation [7]. However, it has to be taken into account that there is potentially an infinite number of possible multimedia configurations as one of the five domain describing sessions is continuous, i.e. the video quality factor. Even although we could reduce it into a discrete one with one hundred different values for the video quality factor, this would lead to a total of 28800 possible configurations that eventually should have to be exchanged between agents. Instead of that, we could avoid such an exchange of information by bargaining. With bargaining, agents alternatively propose offers until they reach an agreement.

The negotiation process will be defined with the specification of two elements: a protocol and a strategy for negotiation [7]. More formally, let $C = \{c_1, c_2, \ldots, c_n\}$ be the space of possible agreements for the negotiation in where $c_i \in C$ is an actual configuration compound by a video codec, an audio codec, a number of frames by second, a quality factor and a frame size. Each agent a_1, a_2 will have its corresponding utility function denoted by $u_i : C \rightarrow [0,1]$ with $i = 1, 2$. This function will be used to valuate each possible multimedia session configuration. Notice that, in this work, both utility functions have the same form. The only things that differentiate them are the different preferences expressed by both users. Utility functions are defined by using user preferences on each topic which compound the configurations. This part of the work is out of the scope of this paper and we use a fuzzy inference system to model each u_i, using as input the preferences mentioned [2].

Globally, we will adopt the monotonic concession negotiation protocol proposed in [6], page 40. In this simple protocol, each agent alternatively makes a possible offer. If the agent that receives the offer is happy with it, the agreement is reached. If not, it has to make a counteroffer. The strategy will describe when an agreement is reached and how to make a new offer when not. Lets suppose that, in some point of the negotiation process, agent i receives the offer $o_j(c_k, v_{kj})$ where c_k is the k-th possible agreement point (i.e. multimedia session configuration) of C and v_{kj} the value $u_j(c_k)$, i.e. the valuation of agent a_j for the $k-$th configuration. Then, the basic strategy of the protocol is as follows:

> The agent a_i will consider $o_j(c_k, v_{kj})$ as an agreement if
>
> - $v_k = \max_{t|c_t \in C} v_t$
>
> Otherwise
>
> - $C \leftarrow C - \{c_k\}$
> - Compound a counteroffer with $c_{k'}, \; v_{k'i}$ where $v_{k'} = \max_{t|c_t \in C} v_t$

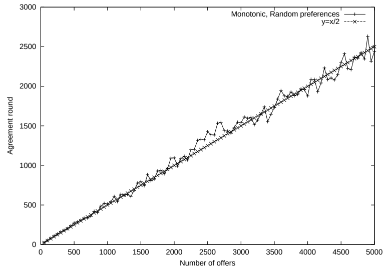
Both protocol and strategy were already studied also in [5]. The author claims there that the algorithm converges to an agreement in any situation. We give here some empirical results obtained from some simulations just to test the speed of convergence to an agreement, in terms of the number of exchanged offers. Simulations use two agents performing this kind of negotiation (i.e. protocol and strategy) varying the size of C and with random preferences over the elements $c \in C$ for both. Results appear depicted at curves (a) and (c) of figure 3. Each point at these curves represents the mean of 100 different meassures. Notice that, with random preferences, a number of aproximately $n/2$ counteroffers are needed to reach an agreement, being n the number of elements in C. Moreover, difference between the obtained common wealth (i.e. the arithmetic mean for $u_1(v_k)$ and $u_2(c_k)$, being c_k the agreement point) and the optimum is very high.

This poor solution could be improved by adding an additional feature to the strategy. This consists in each agent being able of recall all the offers it has previously made at the encounter. The key idea is that if the counterpart of an agent makes an offer that the agent already made, this should be accepted as the counteroffer the agent could immediately made will inevitably have lower utility value (as it is performed later).

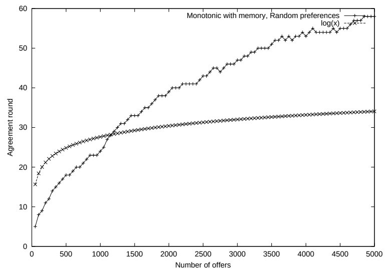
In this way, when agent a_i receives the offer $o_j^t(c_k, v_{kj})$, being t the total number of offers made we can use the strategy that follows:

> Agent a_i will consider $o_j^t(c_k, v_{kj})$ as an agreement if one of the two conditions is satisfied
>
> - being $o_i^{t+1} = (c_{k'}, v_{k'})$, then $c_k = c_{k'}$
> - or $o_i^l = (c_{k'}, v_{k'})$, with $l < t$ and $c_k = c_{k'}$
>
> Otherwise
>
> - $C \leftarrow C - \{c_k\}$
> - Compound a counteroffer with $c_{k'}, \; v_{k'i}$ where $v_{k'} = \max_{t|c_t \in C} v_t$

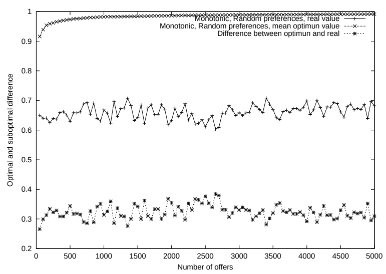
It can be seen that, in the one hand curve (b) at figure 3 shows how the agreement point is reached pretty before than in the case of the simple strategy and in the other hand, curve (c) reflects that the common wealth obtained is also very close to the optimum.



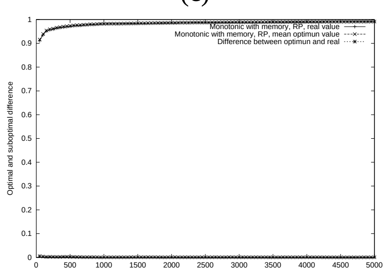
(a)



(b)



(c)



(d)

Figure 3. Simulation results for the monotonic concession protocol, with basic strategy (upper) and memory based strategy (lower). Curves (a) and (b) reflect the necessary number of counteroffers needed to reach an agreement, depending on the number of elements of C, for basic and memory based strategies, respectively. Curves (c) and (d) indicate differences between real and optimum common wealth for basic and memory based strategies, respectively.

3 Conclusions and future trends

In this paper, a general framework for multimedia session management is offered. It is based on the general ideas provided by SIP and FIPA agents technology. It shows how the philosophy of SIP can be used at the FIPA agent level to build, with little effort a mechanism for session establishment. We have also shown that a simple modification to the monotonic concession protocol can make it suitable for the negotiation mechanism we propose. Moreover, the modified algorithm can be applied in any situation, not just only video conferencing.

Future works include considering not only media preferences but also device and generic preferences following the same approach as the one introduced in this paper. Also, the suitability of the modified monotonic concession protocol has to be studied from the point of view of heterogeneity of users.

References

[1] Federico Bergenti, Agostino Poggi, Bernard Burg, and Giovanni Caire. Deploying fipa-compliant systems on handheld devices. *IEEE Internet Computing*, 2001.

[2] Juan A. Botía, Mercedes Valdés, Pedro Ruiz, and Antonio F. Gómez-Skarmeta. Applying fuzzy logic to model user preferences on multimedia sessions. In *Proceedings of the ESTYLF*, Jaen, September 2004.

[3] Handley and Jacobson. Sdp: Session description protocol. Request for Comments 2327, April 1998.

[4] N. R. Jennings, P. Faratin, A. R. Lomuscio, S. Parsons, C. Sierra, and M. Wooldridge. Automated negotiation: prospects, methods and challenges. *Int. J. of Group Decision and Negotiation*, 2(10):199–215, 2001.

[5] Jorg P. Muller. *The Design of Intelligent Agents. A Layered Approach*, volume 1117 of *Lecture Notes in Artificial Intelligence*. Springer, 1996.

[6] Jeffrey S. Rosenschein and Gilad Zlotkin. *Rules of Encounter. Designing Conventions for Automated Negotiation among Computers*. MIT Press, 1994.

[7] Tuomas W. Sandholm. Distributed rational decision making. In Gerard Weiss, editor, *Multi-Agent Systems. A Modern Approach to Distributed Artificial Intelligence*. MIT Press, 1999.