Human-Computer Interaction in Real Time Workflow System using Voice Interface

Abstract. This work explores the use of a voice interface to interact with mobile users in Workflow systems. Voice interface is a technology that supports human-computer interaction by processing verbal communication. Voice interface extends the Workflow systems capability of managing mobile users by using common communication features available in wireless devices. The main goal of this work is to present contributions for modeling, implementing and evaluating the integration of the voice interface for mobile users and Workflow systems. Another interesting point addressed in this paper is the real-time support offered by the voice interface with respect to the interaction with the mobile users.

Key-words. Workflow, Voice Interface, Coordination of Mobile Users, Telephonic System, Java API.
1. Introduction

New communication technologies and the proliferation of the use of cellular devices have extended the use of WFMS (Workflow Management Systems) to environments where interfaces of type "desktop" are not available. A cellular device supports a diversity of interface systems, such as DTMF (Dual Tone Multiple Frequency), e-mail, SMS (Short Message Service) and of voice. Among these technologies, voice is important because has a synchronous property. That means that the action of exchanging information between a user and the computational system can happen in synchronization (i.e., within a known-finite interval) with the event that produced such an action. Such a facility it not provided when e-mail messages are used (asynchronous communication). Asynchronous communication is the standard method for supporting human-WFMS interaction (Min et al, 2001, pp 3-7), and exploring the synchronous approach is one of the main motivations of this work.

One important speak that can benefit of the synchronous So that people are synchronized with information managed for Workflow system is necessary a mechanism of localization these users. Additionally, processing of speaks contribute to development applications for wireless devices (Boozer, 2001), as the cellular one, that it has the paper to interact with the located resource (person).

The goal of this work is to explore Workflow systems for mobile users and resources for Voice Interface, adapting technologies to obtain communication in real time. Analyses about these resources produce an integration proposal, commented in section 5. After that, they are presented data about tests in a critical environment scene, where excess of time in deciding processes can generate fines. Through these stages, it proves the relevance of this work, which present last section.

2. Workflow Systems

A Workflow Management System (WMFS) is a combination of programs and technologies employed to automate the management of processes (Hollingsworth, 1995). A process is formed by a set of tasks (activities). Tasks can be performed with or without human interaction.

Workflow systems are classified according to the flexibility of the rules associated to the processes. (Nicolau, 1998) classifies WFMS into tree classes: Workflow oriented to people, oriented to systems and transactional. Workflow oriented to people control and coordinate tasks performed by human resources. Workflow oriented to systems control the execution of tasks performed by computational programs, with little human interaction. Transactional workflow coordinates multiple tasks with our without human interaction, by assuring some transactional properties such as atomicity, consistence and isolation. (Bortoli and Price, 2000) proposes different classes of Workflow: Ad-hoc, Administrative and Production. Adhoc workflow systems are also called collaborative. In such systems, the work flow is just partially defined, as the participants have the capacity of modifying the flow, by interfering in the ordering and coordinating of the tasks. In Administrative workflow systems, tasks coordination can be automated, as i generally involve repetitive tasks which are not dependent of a complex data processing. Finally, Productive workflow systems employ well-defined process rules, dependent of complex data processing, and there is no margin to fail.
Those classification proposals, although distinct, follow the same architectural principles defined by the WfMC (Workflow Management Coalition) (Hollingsworth, 1995). A generic Workflow system is formed by a database, a management module, engines and other type of data controllers. The management module, as called WfMS (Workflow Management System), controls data about the processes – a kind of observer. Meanwhile, an engine is a system capable of executing a process instance, which can activate several activities as a way of solving a problem indicated by the management module. Other types of controls can be used as support tools. Figure 1 presents the components and relationships that define a Workflow system.


The workflow system modules communicate by means of interfaces. The following interfaces are recommended by the WfMC (they are also presented in Fig. 2):

- Interface 1 → defines standards for representing and downloading process definitions into the Workflow system;
- Interface 2 e 3 → defines methods for controlling external applications used to contact and support users, respectively;
- Interface 4 → defines a standard way to communicate with other Workflow systems;
- Interface 5 defines methods for supporting the integration with administration and monitoring tools.

*Figure 2. Workflow Reference Model – Components and Interfaces.*


Zur Muehlen (2004:39-50) presents a case of use for these interfaces, where the triggering of a process can be executed by using interface 5, or resulting from actions from other processes. When that happens, the WiMS activates at least on Workflow engine in order to follow up the execution and, next, activates the sequence of activities that represent a solution path. The responsibility for executing an activity, i.e., the action of delegating a task to a user, can be performed according two strategies: compulsorily or by negotiation. The task assignment is represented by the user’s worklist. The first contact of the workflow system with the assigned users is performed by means of Interface 3, traditionally by using e-mail messages (Becker et al., 2002). Such a contact can is related to the assignment of a task, e.g. “Can you do such a task?” The user answers the assignment by accepting, denying or simply omitting, which can be performed by means of the Interface 2. Omitting happens when the user ignores the contact, leading to another resource to assume the task.

### 2.1 Real Time Workflow of Mobile Users

The real-time workflow of mobile users is characterized by the facility of contacting and obtaining answers from users within known and controlled time intervals. Workflow systems can benefit from mobile communication facilities, such as cell phones. In order to do that, one must extend the workflow activities; which requires the adaptation of Interfaces 2 and 3 to the new wireless technologies (portable, mobile and location independent). The mechanisms that permits to locate resources and users in such environment is call “mobile user coordinator” (presented in the next session), which can be part of an invoked application.
Applications can be invocated by the Interface 2 or the software agents from the Interface 3 (Zur Muehlen, 2004:39-50). However, because the WfMC intents to be a generic reference model, new invocation mores are being developed. In this section, some related works, corresponding to different Workflow systems, are analyzed with respect to invocation facilities, synchronization facilities and support to coordinate mobile users. We have selected three representative proposals of works handling new methods of invocation and coordination. A summary of these proposals is presented as follows:

1) AdaptWeb → is a distance learning environment, with the facility of coordinating and controlling the availability of learning content for working groups (Freitas, 2002:10-23). In this work, the module responsible for invoking applications is called adaptable interface, and it interacts with a monitoring module. The monitoring module operates directly on the Workflow system database, informing other on-line users about activities completed and new activities requests. According to the AdaptWeb proposal, a user is responsible for invoking the applications. A user can be mobile, at the condition of having a mobile device connected to the Internet. AdaptWeb can be considered a traditional Workflow, without coordination of mobile users, and with the support to synchronous communication only for on-line users.

2) SIH → is an information system for Hospitals (Graeber, 1998), employed by the endoscopies department of the Saarland University, in Homburg, Germany. According to SIH, a worklist for a medical team is manually initiated as a new patient arrives. The invocation of applications is then automated by the worklist module, informing the medical team about the new patient, and controlling the details of the treatment process. The application employs a proprietary technology operating over TCP/IP for communication with work stations. In this environment there is no support to mobility, and there is no invocation of synchronous applications. The workflow members receive messages and must use a workstation in order to consult their worklist.

3) BWE → Binding Workflow Engine is a generic Workflow system, with the potential of being adapted to an organization. All processes are written in XML (Extensible Markup Language) and can operate in a cooperative way, sending e-mails to all workflow members. The process definition facility is one of the most important features in this project. It is implemented by a graphical tool that generates BPML (Business Process Modeling References) code by means of Interface 1. The use of e-mail messages indicates a communication approach based on synchronous messages. The application invocation can on not be automated (Teleterra, 2005). The automated invocation is implemented by means of triggers (i.e., invocations are a result of modifications in a database), while non-automated invocations are implemented by activities executed by the user access interface.

Table 1 summarizes the main features of the works discussed in this section.
Table 1. Workflow system feature summary.

<table>
<thead>
<tr>
<th>Interfaces 2 e 3 implementation</th>
<th>AdaptWeb</th>
<th>HIS</th>
<th>BWE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notification by e-mail</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Coordination of mobile users</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Real time synchronization</td>
<td>(only for online users)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Invocation of applications</td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

An effective analysis of the models shows that workflow systems usually employ the email as the communication facility between the system and the users, as also observed by (Becker et. al., 2002:39-50). This type of technology is important as a complementary facility, but it is not satisfactory for coordinating activities that require real-time interactions. Still, about the analyzed models, users can be mobile, but there is no real-time synchronization with the Workflow database. The mobility facility is implemented by sending e-mail or SMS messages to mobile devices. But the interaction between the user and the workflow system depends on a fixed environment, i.e., a desktop interface. An interface can be considered mobile only when the interaction with the workflow system can be implemented by means of a portable wireless device. In this environment, a facility for locating workflow members is essential. Such a mechanism is presented in the next section.

2.1.1 Coordination of Mobile Users

The process of locating resources and controlling the workflow members in a wireless environment is defined as coordination of mobile users. In workflow systems, such a mechanism assumes that working groups are responsible for solving tasks. A working group can be formed by distinct member types, e.g., a first level of task assignment corresponding to technicians and the second level corresponding to supervisors. The task assignment process must be capable of supporting fail management, by navigating through a database of candidate resources and assigning failed tasks other resources, or by triggering supervisory tasks by contacting the higher level staff. Such a procedure can largely benefit from a real-time interaction between the human resources and the workflow system.

The process of interacting with human resources can be implemented by distinct communication technologies. Considering a mobile environment, a user’s host can be addressed by an IP address or a telephone number. Some mobile devices, such as handheld computers and some cellular phones, can implement emulate an interface between a user and the workflow system similar to a fixed desktop. Such a interface can be implemented by combining technologies such as SMS, e-mail, WML (Wireless Markup Language) pages, J2ME (Java 2 Micro Edition) applications, or still, by using resources universally available in cellular phones, such as voice interface and DTMF, which are going to be presented in the next section.

A client’s host is, therefore, equipment that is capable of invoking the user’s interface application. Meanwhile, the coordination of mobile users is an approach used to contact mobile users by the client’s host, whom from the Workflow system viewpoint, is represent by roles. Figure 3 presents this concept integrated to a Workflow system.
In order to build a voice system one can use a standard named VoiceXML, recommended by the W3C (Word Wide Web Consortium), as exemplified by Glass in 1998. Such type of interface is also known as voice system, where (Mccandless, 1998) presents an analysis and several applications.

In voice systems it is necessary to translate words into signals that could be computationally processed. There are two mechanisms to implement such a translation: (1) TTS – Text-To-Speech and (2) ASR – Automatic Speech Recognition. The TTS mechanism is capable of translating text into speaking language by combining phonemes to emulate the human speech (Min et al, 2001: 3-7). The ASR mechanisms, by the other hand, can be implemented by three different approaches:

1. Natural language – recognizes works in a context of normal speech, emulating an human-to-human interaction;
2. Voice commands – recognizes works only in limited contexts (e.g., a menu of choices); and
3. Authentication – permits to use speech as a biometric password.

The speech systems permit to exchange data in real-time (Mccandless, 1998), implying in a synchronous process for taking decisions and it can contribute for the success of Workflow rules. Figure 4 presents components for supporting the integration between users and the voice interface.

When the dialog between man-and-machine happens, the data presented by the system can be implemented by pre-recorded files with human voice, replacing the voice synthesis (i.e., the TTS mechanisms). However, the TTS approach is more flexible, being capable of presenting information generated by consulting a data base, for example.

In order to build a voice system one can use a standard named VoiceXML, recommended by the W3C (Word Wide Web Consortium), as exemplified by Glass in...
(Glass et. al., 2001). Alternatively, Nuance System\(^1\) and CMUSphinx\(^2\) offer an API (Application Programming Interface) implemented in Java, which presents some flexibility advantages with respect to the VoiceXML approach.

![Figure 4. Dialog mechanisms for the voice system.](image)

Cellular phones offer support to voice interaction regardless the user location. Therefore, mobile telephonic networks can be considered an ideal environment (Polifroni et. Al, 1998), even though the voice interface could also be available on IP based networks, by using sound boards available in desktop computers.

In order to adapt the workflow system to the voice interface, it is necessary to employ specialized telephonic boards, capable of placing and receiving telephonic calls, and also pre-processing audio. Such telephonic boards work as a sort of modem to the workflow system. The main resources supported by such boards are presented in Table 2.

**Table 2.** Features available in telephonic boards used for implementing the voice interface.

<table>
<thead>
<tr>
<th>Recurso</th>
<th>Descrição</th>
</tr>
</thead>
<tbody>
<tr>
<td>Placing a call</td>
<td>Placing a phone call is the feature that permits to contact a workflow member. By the other hand, a workflow user can also place a call to the workflow system in order to have access to the worklist.</td>
</tr>
<tr>
<td>Waiting a call</td>
<td>It is the state when the system is ready to receive a phone call.</td>
</tr>
<tr>
<td>Answering a call</td>
<td>Answering a phone call can be performed by a user, by a TTS or just triggers an automatic system action.</td>
</tr>
<tr>
<td>Transferring a call</td>
<td>It is the capability of transferring a call to another user or to a TTS.</td>
</tr>
<tr>
<td>Detecting the end of a communication (hangup)</td>
<td>This feature permits to immediately detect that the user has abruptly finished an established call.</td>
</tr>
<tr>
<td>Detecting DTMF</td>
<td>DTMF are tones emitted by the telephonic system.</td>
</tr>
</tbody>
</table>

\(^1\) [http://www.nuance.com](http://www.nuance.com) – Speech solutions.

This feature permits a user to utilize the telephone keys to input data to the workflow system.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interrupting (barge-in)</td>
<td>The barge-in is responsible for keeping several channels in a single connection (Strom, 2000). This feature permits to detect a user input (by DTMF or voice), even when a channel is being user for playing an audio to the user. Then, an action such as DTMF or voice recognition can be performed, in order to accelerate the process of interacting with the user.</td>
</tr>
<tr>
<td>Voice recording</td>
<td>Voice recording is a resource used to store the speech in a database, permitting to audit the development of a voice-based process.</td>
</tr>
</tbody>
</table>

Boozer (2001) declares that, independently of the resources available, one cannot assure 100% of success in the voice recognition process, because of the perturbations related to the user emotion, pronunciation failures and environmental noise. However, combining voice recognition with DTMF can be a satisfactory approach for supporting a reliable data exchange.

### 4. Integration Proposal

The integration proposal consists in using the voice interface as a mechanism for supporting synchronous communication between the human resources and the Workflow system. The proposal is based on the workflow reference model proposed by the WfMC. Figure 5 presents the communication model adopted in this proposal.

![Proposed Integration Framework](image)

**Figure 5.** Proposed Integration Framework.

The proposal defines a Java API for coordinating the mobile users and another Java API for building the voice interface, represented by Figure 6 and describe in the Tables 3 and 4, respectively.
Figure 6. UML Diagram for Voice Interface.

Table 3. API for coordinating mobile users.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>addUser(String pstlIdentify) : void</td>
<td>Creates a user entry in a work group.</td>
</tr>
<tr>
<td>addTelephone(String pstlIdentify, String pstlTelephone) : void</td>
<td>Creates a host entry mapped to a user</td>
</tr>
<tr>
<td>next() : Boolean</td>
<td>Permit to navigate by the users entries.</td>
</tr>
<tr>
<td>getCurrentUser() : String</td>
<td>Returns the currently selected user</td>
</tr>
<tr>
<td>getCurrentTelephone() : String</td>
<td>Returns the client host information of the selected user</td>
</tr>
</tbody>
</table>

Table 4. API for building the voice interface.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>placeCall() : int</td>
<td>Places a call for contacting a user.</td>
</tr>
<tr>
<td>waitCall() : Boolean</td>
<td>Waits for a call. This method is used for implementing the worklist</td>
</tr>
</tbody>
</table>
In our proposal, the workflow system is generic. The invocation of applications and the application for the users interface (Interfaces 2 e 3), from the original reference model, have been adapted to the mobile environment and renamed to “activation interface”. Such interface has the capability of evocating applications and is integrated with Workflow engine by means of a message queue.

From Figure 5, one see the voice interface is requested by the workflow engine (step 1) and it is invoked by the activation interface, opening a speech channel with the client’s host. Mobile users receive the notification and initiate the communication with the invoked application (step 2), which by its turns, collects the data about the task execution and updates the process data in real-time (step 3). Finally, the workflow engine queries the process data and take new decisions (step 4), completing the communication cycle.

The activation interface used for instantiate the worklist is the same used for contacting the mobile users. However, as soon invocating, the worklist is placing in a waiting call state, and it can be managed by the monitoring and management interface (Interface 5). The invocated applications are implemented by using the proposed APIs.

The proposed model defines a transparent invocation strategy, supporting both, synchronous and asynchronous communication, being independent of the type of address of the client host. The asynchronous communication can be implemented by e-mail or SMS.

### 5. Use Case and Evaluation

The proposed framework have been implemented, and used to control a real time-critical process. The scenario corresponds to a maintenance service offered by a telecommunication company, which supplies equipments for building cellular network infrastructures. The company is responsible for monitoring and accepting emergency maintenance calls for repairing equipment’s failures. Time-critical constraints are imposed, i.e., the maintenance operation must be performed within a limited time, otherwise, the maintenance company is charged from penalties proportional to the overlapped time.

The maintenance calls are performed by e-mail send manually or automatically by a supervisory system. From the WfMC reference model viewpoint, the e-mail messages

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>playTTS(ustrMessage: String): void</td>
<td>Present data by text to voice synthesis.</td>
</tr>
<tr>
<td>playWAV(ustrWAVFileName: String):void</td>
<td>Presents data by using pre-recorded files (WAV).</td>
</tr>
<tr>
<td>playTTSAndRecognize(ustrMessage: String):String</td>
<td>Presents TTS/WAV data and recognizes Voice and DTMF.</td>
</tr>
<tr>
<td>playWAVAndRecognize(ustrWAVFileName: String):String</td>
<td>Presents TTS/WAV data and record the user’s speech.</td>
</tr>
<tr>
<td>playTTSAndRecord(ustrMessage: String):boolean</td>
<td>Presents TTS/WAV data and record the user’s speech.</td>
</tr>
<tr>
<td>playWAVAndRecord(ustrWAVFileName: String):boolean</td>
<td>Presents TTS/WAV data and record the user’s speech.</td>
</tr>
<tr>
<td>Hangup( ): void</td>
<td>Disconnects the user, releasing the phone line to another call.</td>
</tr>
</tbody>
</table>
correspond to the Interface 5, because they represent the communication between the workflow and other systems. The e-mail messages follow a standard template. A message corresponding to a new notification failure triggers the creation of a process instance by the workflow engine.

The workflow engine will then execute the tasks corresponding to the process activities. Each activity is mapped to an application. Most applications correspond to Java classes developed using the proposed APIs. The applications perform tasks such as consulting the resource database, sending e-mail or SMS messages, or still, locating a resource by using the voice interface. According to the implemented strategy, the human resources are grouped into specialized groups, corresponding to their capacity to manage specific types of failures. As soon a failure notification is received, a maintenance group is selected based on the type of failures and the maintenance process starts.

Initially, an activity sends e-mail and SMS messages to all members of the selected maintenance group. The message describes the type of failure, the site location where the failure occurred and the deadline imposed to solve the task. In parallel, another activity contacts each group member individually, using the voice interface. The application corresponding to this activity has been implemented using the proposed Java API, which encapsulates the underlying Nuance System implementation. This approach turns our implementation independent of the TTS, ASR and telephonic technology.

Once a technician accepts the task by using the DTMF interface, a task assigned to his worklist. At any moment, the technician can interact with the worklist by phoning to the workflow system, and modifying the task status by using the voice interface and DTMF. The worklist can be accessed also by e-mail or SMS, but in this case, only consulting is available. A timer is triggered as soon as a new notification failure arrives, controlling the maximum time for solving the problem. During the process, the technicians are contacted by the workflow system, by using the voice interface, in order to anticipate eventual delays for solving the problem and triggering alternative procedures. By using the voice/DTMF worklist interface the technician can inform that the task has is progressing well, has been concluded or, if he has not the resources for solving the problem, ask for help. In this case, the group supervisor is contacted to execute an alternative procedure.

The workflow procedure has been evaluated by two months, using two telephone lines: one for calling the workflow members and another for receiving worklist consulting. During this period, 167 new process instances have been created. The proposal has been evaluated according to the following logic:

\[ \text{If } ( \Sigma t_E < t_{LA} ) \text{ then} \]

\[ \text{“the workflow is suitable for this class of problem.”} \]

\[ \text{else} \]

\[ \text{“the sistem is not suitable.”} \]

\( \Sigma t_E \) represents the total time elapsed from the failure notification until the maintenance process is considered concluded. \( t_{LA} \) represent the time constraint to solve the problem.
imposed by the agreement with the maintenance company. The proposal is considered suitable for the problem if the delay component introduced by the workflow system does not have an impact in the ability of the technicians to solve the problem. In order to isolate the delay component introduced by the system, the following variables have been considered:

$$\Sigma tE = \mu C + \mu Wf + \mu IA + \mu TA$$

The time is measured in minutes and the $\mu$ symbol denotes the average time elapsed in each component. $C$ refers to the time consumed to recognize the incident; Wf is the time consumed by the workflow system; IA is the time consumed for instantiate the activities corresponding to the voice interface. Finally, TA represents the time consumed to locate a responsible for executing the maintenance task.

In our evaluation, the $\mu C$ could not be considered because some notification failures have been sent to the workflow system manually by e-mail, by a human operator.

By taking into account the data collected during the experiments we found the following average values for the delay components. The workflow system has consumed about 30 seconds to initiate a new process ($\mu Wf$). The activation interface consumed about 15 seconds to invoke an activity for locating a human resource ($\mu IA$). The results corresponding to $\mu TA$ are presented as a histogram, as show in Table 5.

<table>
<thead>
<tr>
<th>Class (minutes)</th>
<th>fi</th>
<th>xi</th>
<th>fi * xi</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 ¬ 8</td>
<td>81</td>
<td>4,5</td>
<td>364,50</td>
</tr>
<tr>
<td>8 ¬ 15</td>
<td>46</td>
<td>11,5</td>
<td>529,00</td>
</tr>
<tr>
<td>15 ¬ 23</td>
<td>32</td>
<td>19</td>
<td>608,00</td>
</tr>
<tr>
<td>23 ¬ 31</td>
<td>6</td>
<td>27</td>
<td>162,00</td>
</tr>
<tr>
<td>31 ¬ 39</td>
<td>2</td>
<td>35</td>
<td>70,00</td>
</tr>
<tr>
<td>Total</td>
<td>167</td>
<td></td>
<td>1733,50</td>
</tr>
</tbody>
</table>

$fi \rightarrow$ frequency  
$xi \rightarrow$ average

The following equation calculates average the time consumed for assigning the task to a mobile resource:

$$\mu TA = \frac{\sum (fi * xi)}{\sum fi} = \frac{1733,50}{167} = 10,38$$

The 10,38 value corresponds to 10 minutes and 23 seconds. The total time, including the fixed components, is calculate as follows:

$$\Sigma tE = \mu C + 0,50 + 0,25 + 10,38 = \mu C + 11,13$$
By this expression, excluding $\mu$C and tLA, which are dependent to human intervention and the maintenance operator agreement, it is possible to evaluate if the system is suitable for the problem or not.

In order to evaluate the gain obtained with the automated approach, it is necessary to compare the obtained results with a manual approach, where a call center is responsible for accepting the notification failure and finding a human resource to execute the maintenance procedure.

By evaluating the database available using a traditional the call center operation for this exact process, one have observed that the time elapsed for assigning a task to a maintenance resource is about 20 minutes. Considering this result, we can elaborate the following expression:

If \((\mu C + 11.13) + \alpha < 20\) then

“the system is suitable for this class of problem – where $\alpha$ determines the confidence in the values found.”

else

“the system is not suitable.”

Therefore, application where the time required for assigning a task is at least 11 minutes can be considered suitable for being automated by our proposal.

6. Conclusions

Traditional workflow systems address the problem of assigning tasks non mobile users, by using a desktop-based system interface. However, the availability of cellular phones and other mobile technologies has permitted to extend the workflow context to mobile users.

Considering cellular networks, the voice interface can be used as an important instrument for building workflow interface to mobile users. There approach is particular advantageous for controlling process with rigid time constraints, because the voice interface (combined with DTMF) provides a synchronous interaction between the workflow system and the human resources.

This work has proposed and evaluated a framework that uses a voice interface to permit workflow system to coordinate mobile users. We have defined Java APIs for simplifying the process of developing such an interface. The Java API encapsulates functionalities that are dependent of the TTS, ASR and telephonic board implementation, such as recognizing DTMF.

We have evaluated the proposal in a real scenario, and concluded that the human-machine interaction has been simplified by the user of the voice interface. The synchronous feature related to this interface has also been very useful, because the workflow system was able to quickly locate a resource for solving a task, simply by inquiring each resource candidate about his availability to perform the task. The system has also improved the control about the task execution, by querying the workflow
members about its activities, and triggering corrective procedures before the time
constraints would be elapsed.

Finally, we conclude that the voice interface approach is suitable for processes where
the time constraints are within a range of some hours. Longer time constraints could be
handled by asynchronous interfaces, such as e-mail. Shorter time constraints would be
severely penalized by the time consumed to serially contacting the resources for
accepting the task.

References

Architectures: Classification and Characteristic of Workflow-based Information

Dialogues”. Master Thesis, Massachusetts Institute of Technology, Massachusetts,
United States of America.

Requisitos. Terceiro Workshop de Engenharia de Requisitos, Universidade Federal
do Rio Grande do Sul.

Conference on Information and Communication Technologies in Education,
Badajoz, Spain, Nov. 10-23.

Developing Conversation User Interface”. Massachusetts Institute of Technology,
Massachusetts, United States of America.

Hospital Information Systems”. University of Saarland, Homburg, Germany.

Coalition, Winchester Hampshire, United Kingdom.

Speech Research”. Doctor Thesis. Massachusetts Institute of Technology,
Massachusetts, United States of America.

Synthesis to Mammalian Vocalizations”. EuroSpeech, Aalborg, Denmark, Sep. 3-7.

Temporal Orientado a Objetos com Papéis. Dissertação de Mestrado em Ciência da
Computação. Universidade Federal do Rio Grande do Sul.

Telephone-Based Conversation System. Massachusetts Institute of Technology.

Massachusetts Institute of Technology.