FuzzyMAN: An Agent-based E-Marketplace with a Voice and Mobile User Interface

Frank Teuteberg and Iouri Loutchko

Abstract. The popularity of wireless devices stimulates widespread efforts to develop voice and mobile user interfaces for software systems in general and for e-marketplaces in particular. This paper focuses on conceptual foundations and the architecture of an agent-based job e-marketplace supporting mobile negotiations. The negotiation model is based on many negotiation issues, a fuzzy utility scoring method, and simultaneous negotiation with many negotiation partners in an environment of limited negotiation time. The design and implementation of voice and mobile user interfaces for accessing information on an e-marketplace called *FuzzyMAN* (Fuzzy Multi-Agent Negotiations) is described. A comparison of technologies for the development of voice and mobile user interfaces is given.

Keywords: software agents, e-marketplace, mobile negotiations, multimodal user interfaces, VoiceXML, J2ME, XHTML, XSLT.

1 Introduction

Recent advances in digital speech processing technologies and standards such as SALT (Speech Application Language Tags) [1] and VoiceXML (Voice Extensible Markup Language) [2] enable developers to enhance mobile applications with voice user interfaces to interact with e-marketplaces and other applications.

Mobile and especially voice-based access to e-marketplaces has several advantages in comparison to web-based access. Voice user interfaces enable users to directly access information via spoken commands instead of browsing through several drop down menus before coming to the desired information. Furthermore, handicapped persons (e.g. blind persons) who are unable to access e-marketplaces due to physical disabilities may gain access through voice-enabled mobile devices. The mobile and voice user interfaces described in this paper allow users to access applications such as e-marketplaces in a manner appropriate to the users' situation, for example, using speech, keyboard, web browser, or stylus.

In this paper, we also demonstrate the concepts of a multi-agent approach to electronic marketplaces using a prototype of a job e-marketplace called FuzzyMAN (Fuzzy Multi-Agent Negotiations). On the FuzzyMAN marketplace users create their individual agents that then act autonomously and proactively. Mobile, voice-enabled, and web-based user interfaces can be used to initialize the agents, manage them, and receive their feedback. Once initialized, the agents continuously search for new positions (employees' agents) or for new employees (employers' agents). To achieve their objectives (in our case, to find an appropriate position or an appropriate employee) the agents negotiate about the conditions of a contract according to several criteria like salary per hour, working hours per week, and social benefits. As soon as a suitable position or a suitable candidate for a vacant position is found, the agents inform their owners about the conditions of the contract. Other useful features are that users can permanently monitor the performance of their agents and the situation on the marketplace, and that they may change their preferences for salary per hour, working hours per week, the negotiation tactics employed by their agents, the dates their agents expire, etc.

The remainder of this paper is organized as follows. In section 2, we describe FuzzyMAN's architecture. Section 3 discusses some essential features of the agents acting on the FuzzyMAN e-marketplace. Section 4 introduces the underlying multilateral negotiation model. Section 5 focuses on the design and implementation of voice and mobile user interfaces based on VoiceXML and J2ME technology. In section 6 research efforts of other researchers related to our work are described. In the final section some experimental results, further research questions, and some concluding remarks are given.

2 FuzzyMAN's Architecture

The agent-based e-marketplace FuzzyMAN was developed in a research project supported by the German Research Foundation and can be accessed through web-browsers as well as mobile devices.

The FuzzyMAN marketplace is a multi-agent system with the architecture shown in figure 1. In this section we describe the structure and functionality of the marketplace. The architecture includes both a specific mediator agent playing the role of a market coordinator and agents representing individual users with their own goals and tactics. The mediator is a centralized 'super-agent' handling the communication of other agents. Initially, all users have to register when entering the marketplace to create their own agents on the server. Employees' agents search for adequate jobs and employers' agents offer jobs. Each agent is uniquely identified by an agent-id. Users may access FuzzyMAN through mobile, voice-enabled, and web-based interfaces. Here the user can

create an agent, set the agent's preferences, change personal settings, and access information provided by the agent.
An agent created by a user contacts the mediator agent to enter his profile, the user's data, and/or job offers or search requests into the database. The agents' preferences are used by the mediator agent to find and rank agents suitable for negotiations.
The mediator agent continuously seeks for pairs and/or groups of employees' and employers' agents with corresponding preferences. It initiates and manages the negotiations in different threads once suitable agents are found. After several rounds of exchanging offers and counter-offers, the negotiation process either ends with an agreement between an employer's and an employee's agent in the ideal case, or terminates without success. The mediator agent and the agents' owners are informed about the negotiation results.
The entire negotiation process comprises several stages, including:

- Discovery of potential negotiation partners in a pre-selection process with the help of the mediator agent.
- Determination of the negotiation issues through a communication process based on FIPA-ACL/XML (FIPA-ACL = FIPA Agent Communication Language [3]; XML = Extensible Markup Language).
- Automated negotiation by exchanging XML messages with values of negotiation issues generated by means of the agents' tactics.

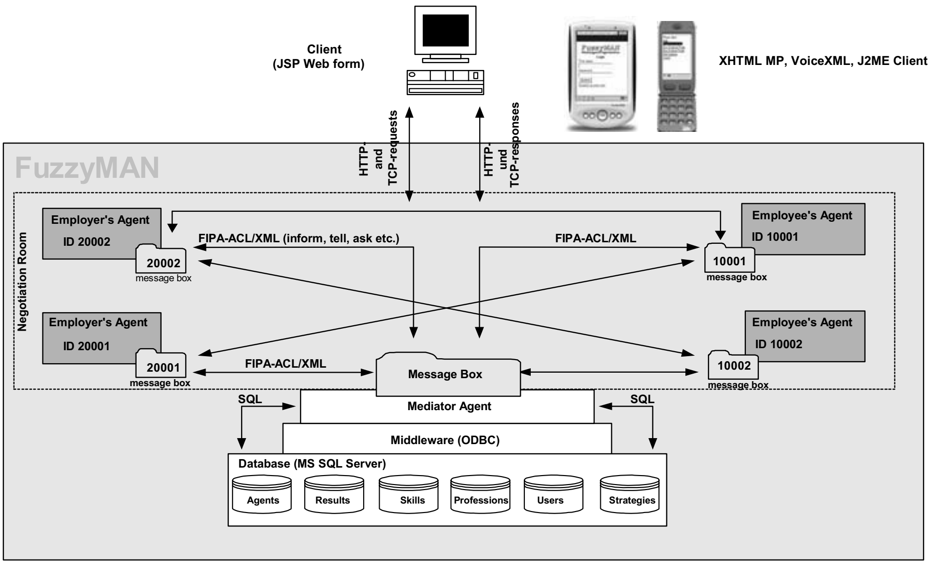

Fig. 1. Multi-agent system architecture

FuzzyMAN is a platform-independent application implemented in Java. In order to make FuzzyMAN widely accessible, the implementation was based on the following technologies:

- FIPA-ACL: Used as the agent communication language.
- XML: Used as the data interchange language for representing and processing fuzzy negotiation vocabulary from a syntactic and a semantic perspective. In XML domain-specific entities, attributes, and relationships between entities can be defined with tags like <salary>, <profession>, etc. The XML content of messages is constructed based on Document Type Definitions (DTDs). DTDs can be used as templates to define content models, the valid order of elements, and the data types of attributes.
- JDBC (Java Database Connectivity): Used to access FuzzyMAN's relational database from Java.
- MS SQL Server 2000: Used as database management system.
- JSP (Java Server Pages): Used for the generation of dynamic content from the database, writing to the database, and for the web forms to enter user data.
- XSLT, XHTML, VoiceXML, Java 2 Platform Micro Edition (J2ME): Used for realizing mobile and voice user interfaces.

FuzzyMAN is currently running on a server with a Windows 2000 operating system, 40 GB hard disk, Intel Pentium processor 1 Ghz and 256 MB DDR-RAM. The system was developed and tested using the following software versions: Java Software Development Kit 1.3, Tomcat Server 3.3, Internet Explorer 5.5 and 6.0.

The underlying database mainly stores information that is needed for the agents to do their work. This information includes, for example, preferences set by the users and statistical data about earlier negotiation results. Figure 2 gives an overview of the database tables and their interrelations in UML (Unified Modeling Language) notation.

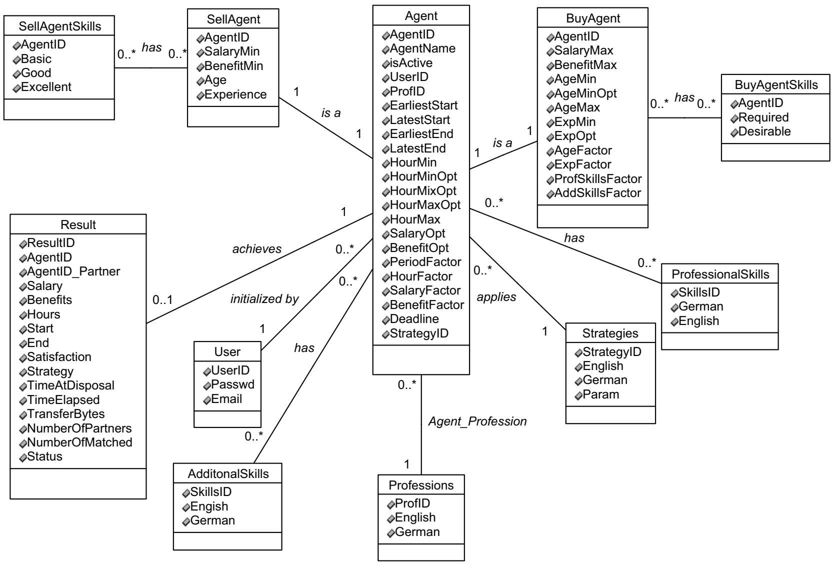

Fig. 2. Schema of the underlying database in UML notation

The multiplicity specifications used in figure 2 indicate how many elements (entities) of one database table (relation) may be connected to the elements of another database table. Multiplicity in UML is analogous to cardinality in other modeling languages. Professions, skills and strategies of agents (sell agents = employees' agents; buy agents = employers' agents) are stored in the database in both English and German to enable bilingual user interfaces.

3 FuzzyMAN's Agents

3.1 Employees' and Employers' Agents

In this section, some essential features of the agents acting on the FuzzyMAN e-marketplace are discussed. Two types of agents have to be considered:
a) *employees' and employers' agents* that communicate and negotiate with each other to achieve their respective objectives, and
b) a *mediator agent* responsible for managing the other agents, collecting statistical information, and selecting candidate agents for a specific negotiation process. We give a short overview of both agent types.
The architecture of employees' and employers' agents consists of three main modules: a communication module, a module for interaction with the agents' owners, and a negotiation module.
Whereas the communication module governs the exchange of messages between agents, the interaction module is responsible for communication between an agent and its owner. Interaction is bilateral: Users can manage their agents through the user interfaces (mobile, voice-enabled, and web-based user interfaces) and agents can contact their owners in certain situations (e.g. successful negotiation, approaching the deadline) via e-mail or SMS. A part of the web-based user interface is illustrated in figure 3. Users can manage their agents through this interface, including initializing, updating, and deleting agents.
For example, to initialize an employee's agent as in figure 3, the user has to specify the profession (e.g. "network administrator"), preferred employment dates, number of working hours per week, expectations about the salary per hour and social benefits, age, professional experience, and expiration date of the agent. Then information about professional skills (like knowledge of Unix, Java, SAP R/3) and additional skills (like foreign languages, driving license, flexibility), divided into three categories (basic, good, or excellent level of knowledge), has to be entered. Finally, the user specifies the tactic that the agent is supposed to employ in the negotiation process, and the weights for the negotiation issues, i.e. salary, employment period, working hours, and social benefits.
The negotiation module consists of three components: negotiation object, decision-making model, and negotiation protocol.

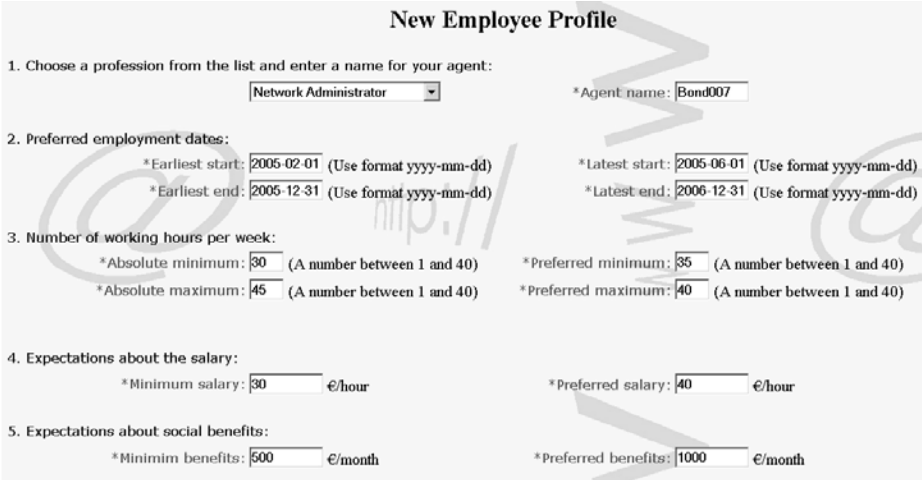

Fig. 3. Part of the input form for initialization of a new employee's agent

The *negotiation object* is characterized by four issues: salary per hour, number of working hours per week, duration of the employment, and social benefits. We note that neither the specific types of negotiation issues nor their number are essential for the negotiation model under consideration; the same model can be used in other situations with different negotiation issues.

The *decision making model* consists of an assessment part (that evaluates an offer received and determines the corresponding action) and of an action part (that either generates and sends a counter-offer or stops the negotiation).

The value of negotiation issues is modeled by scoring functions. Larger scoring function values for particular issues reflect their suitability for negotiation. The scoring functions represent private information not known by and not given to other market participants.

For illustration, an example of a scoring function that can be used by employees' agents on the FuzzyMAN marketplace is presented in figure 4, where h^n_{Min}, h^n_{Max}, $h^n_{Opt\_Min}$, $h^n_{Opt\_Max}$ are the employee's minimal, maximal, and optimal number of working hours per week. The parameter n_h defines an employee's level of satisfaction with the worst case regarding the number of working hours per week.

In a real negotiation, different negotiation issues are of different importance to each partner. To model this situation, the notion of relative importance that a participant assigns to an issue under negotiation is introduced.

Using such scoring functions and the relative importance of issues under negotiation, a general scoring function is defined. The general scoring functions of FuzzyMAN's agents are additive functions [4] summing the scoring functions for individual negotiation issues multiplied by their weights (or their relative importance) that are kept in the interval [0, 1]. The sum of these weights is taken to be equal to 1. We note that this method of defining a general scoring function can only be used when the negotiation issues are independent, as they are in the FuzzyMAN marketplace.

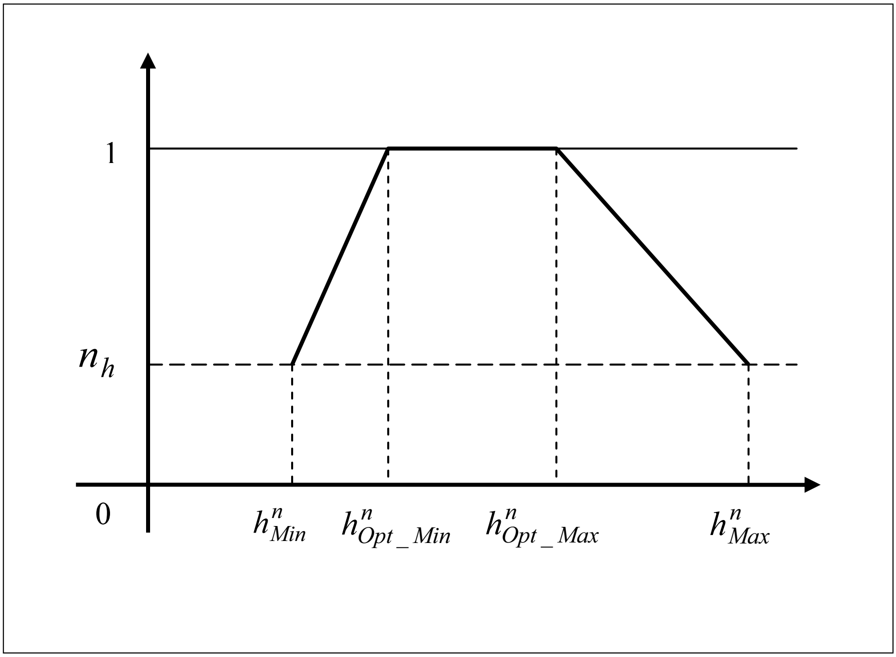

Fig. 4. Scoring function for negotiation issue "number of working hours/week"

The scoring functions are used both in the pre-selection procedure and in the negotiation process, in the latter case to evaluate the offers received from other agents. The pre-selection procedure is described in the next section.

3.2 Mediator Agent and Pre-selection Procedure

The most important functionalities of the mediator agent are management of other agents, collecting statistical information, and pre-selection of agents for a negotiation. In this part of the paper we focus on the pre-selection procedure that plays an essential role in the negotiation protocol.

Pre-selection means selecting and ranking those agents of the marketplace that will start a specific negotiation over certain issues with a given agent. In the pre-selection process, factors like profession, age, working experience, professional and additional skills of the potential employees are taken into consideration. The values of these factors are fixed on the employees' side and cannot be involved directly in the negotiation process although they are very important for the employment process.

For the ranking of employees' agents, additional scoring functions for age, working experience, professional and additional skills have to be constructed [5]. Then an extended set of weights (relative importance) of negotiation and pre-selection issues is defined for each employer's agent: In addition to the weights introduced before, this set contains the relative importance of the employee's age, of his or her working experience, and professional and additional skills.
For evaluation of an offer received by an employer's agent from an employee's agent, the extended scoring function in form of the sum of the general scoring function and the "fixed" scoring function is used. The "fixed" scoring function takes into consideration the fixed characteristics of the employees like their age, working experience, and professional and additional skills. Like the general scoring function described before, the "fixed" scoring function employed by the FuzzyMAN's agents is also an additive function.
For a given employer's agent an order of the set of all employees' agents is introduced according to the following rule:
An employee's agent y is more preferable for the employer's agent than another employee's agent z (or equally preferable) if and only if the value of the "fixed" scoring function for the agent y is greater than (or equal to) the value of the same function for the agent z. In this case, the relation between the two employees' agents is denoted by $y \succ z$ (or by $y = z$).
Using the above procedure, an ordered sequence n_1, n_2, n_3,..., n_l of employees' agents is determined for a particular employer's agent. This sequence is sorted according to the level of suitability as a negotiation partner for the employer's agent determined for each employee's agent:

$$n_1 \succ n_2 \succ n_3 \succ \ldots \succ n_l \,. \tag{3.1}$$

The results of the pre-selection procedure are heavily used in the multilateral negotiation protocol.

4 Negotiation Model

In this section, the negotiation protocol underlying the agents' negotiation on the FuzzyMAN marketplace is briefly presented. A detailed discussion of the multilateral negotiation protocol can be found in [5].
In general, there might be several employees who are interested in a position offered by an employer. Vice versa, several employers might be interested in a certain employee. So the negotiation model underlying a job marketplace should be a multilateral one.
First we discuss how the agents generate their offers and counter-offers, i.e. their negotiation strategies. A *negotiation strategy* consists of choosing an appropriate negotiation tactic depending on the time and other environmental conditions like the situation on the e-marketplaces, the number of negotiating agents, or the status of the current negotia-

tions. The simplest negotiation strategy is a constant one, i.e. only one negotiation tactic is employed all the time. By a *negotiation tactic* a vector function with several components is understood. These components determine the way the agent changes the values of the negotiation issues (in the case of FuzzyMAN's agents: salary per hour, number of working hours per week, employment duration, and social benefits) over time.

In the literature various cases of negotiation tactics are discussed (see for example [6]). Families of tactics which can be employed by agents are time-dependent, resource-dependent, and behavior-dependent tactics. An important family of tactics consists of functions which are only time-dependent, i.e. neither the status of the marketplace nor the status of the negotiation influences the negotiation.

On the FuzzyMAN marketplace agents employ constant strategies with time-dependent tactics in the form:

$$f(t) = f_1 + \left(\frac{t - t_{Init}}{t_{Max} - t_{Init}} \right)^{1/\beta} (f_2 - f_1), \quad f : [t_{Init}, t_{Max}] \to [f_1, f_2]. \qquad (4.1)$$

Tactics of this form are often used in the literature about agents' negotiations. Other approaches to define time-dependent tactics have also been proposed, but the formula (4.1) suited our purpose best because it describes the whole spectrum of time-dependent tactics. In formula (4.1), f_1, f_2 are user-defined values of the negotiation issue (minimal and maximal values of a certain negotiation issue for a given agent), $t \in [t_{Init}, t_{Max}]$ where t_{Init} is the time the agent was initialized by the user, t_{Max} is the time the agent has to complete the negotiation (deadline), and $\beta > 0$ is a parameter that determines how the agent changes the values for the given negotiation issue over time. The larger the value of the parameter β is, the sooner the agent is ready to make larger concessions in the negotiation process.

For multilateral negotiations in FuzzyMAN we employ a negotiation protocol based on the bilateral protocol described in [6] (see also [7] for an extended version of this protocol) and a partial order of the set of all employees' agents generated by the pre-selection procedure discussed in section 3.2.

The general idea behind the bilateral negotiation protocol is that two agents participating in the negotiation sequentially send each other offers and/or counter-offers. To generate the offers negotiation tactics are used. The decision to accept or reject an offer received from the other agent is based on scoring functions constructed according to the agents' preferences, restrictions, goals, and relative importance of negotiation issues. A negotiation is completed either if an offer is accepted by one of the agents (negotiation is successful) or if one of the agents reaches its deadline (negotiation fails). For a formal presentation of the protocol for bilateral negotiation see [6].

On the FuzzyMAN marketplace, a multilateral negotiation is modeled as a finite set of bilateral negotiations between employers' and employees' agents. However, these bilateral negotiations are not independent, but are influenced by other ongoing bilateral negotiations, just as they would be on a real marketplace.

In the following, the general idea of the multilateral negotiation protocol for multiple issues is outlined. The interested reader is referred to [5] for details. Suppose t_{Max} is the deadline of an employer's agent and n_1, n_2, n_3,..., n_l is the ordered sequence of the employees' agents suitable for negotiation with the given employer's agent and ranked by the pre-selection procedure: $n_1 \succ n_2 \succ n_3 \succ \ldots \succ n_l$. The idea behind the protocol is that the total value of offers the employer's agent sends at a particular time $t = t_i$, $i = 0,1,....,n$ (n = number of negotiation steps) to the employees' agents it is negotiating with should be the same for all offers. The total value of an offer takes into consideration both the negotiation issues and the pre-selection issues (e.g. age, working experience, etc.) which are not involved directly in the negotiation.

If a new employer's agent comes to the marketplace, the mediator agent first performs the pre-selection procedure for it. Then the newcomer starts negotiation according to the multilateral negotiation protocol. When a new employee's agent comes, the pre-selection procedure is performed for every employer's agent once again. As a result, the newcomer may become involved in negotiations with one or several employers' agents.

In the case that an agent accepts an offer sent to it by one of the agents it is negotiating with, the agent stops negotiating with all other agents and informs those agents through a special message about the cancellation of the negotiation process. If such a cancellation message comes from one of the agents, that agent is withdrawn from the set of agents negotiating with the agent that received the cancellation message. Likewise at time $t = t_n$, $t_n \geq t_{Max}$ the agent stops the negotiation with all agents it was negotiating with and informs those agents about the cancellation of the negotiation process.

According to the bilateral negotiation protocol both employees' agents and employers' agents always send offers they themselves are satisfied with. Therefore, as soon as an offer is accepted by one of the agents, the agent that sent the offer and the agent that received the offer come to an agreement. Another point to mention is that the offers an agent receives are evaluated first-in-first-out: An offer received earlier than another one is evaluated first and corresponding actions are taken prior to the evaluation of the second offer.

Representing and processing negotiation vocabulary in electronic marketplaces is a difficult task. The explicit specification of common syntax and semantics of negotiation vocabulary is commonly referred as the ontology problem [8], which is still one serious problem in the area of multi-agent research. The problem is to ensure that the negotiation partner have the same understanding of the negotiation issues (e.g. "salary", "extra pay" etc.). On our marketplace we apply an XML-based approach for representing and processing fuzzy negotiation vocabulary from a syntactical and semantical perspective. The vocabulary is represented in a fuzzy way to make electronic negotiations more flexible. As a tag-based language for describing tree structures with a linear syntax, we use XML for the representation of ontologies and inter-agent communication during the negotiation process. XML provides the ability to define negotiation ontologies and to

declare entities, entity attributes and relationships between entities. See [9] for details about the XML representation and its advantages in inter-agent communication.
Figure 5 illustrates our approach to represent an agent's fuzzy scoring function in a linear tree structure. An employee's agent searching for a specific job can apply an XML representation of a fuzzy scoring function to describe its preferences. In fuzzy set theory [10] a fuzzy set is described by a membership function, which can be approximated by a set of points. To each of those points a unique degree of membership is assigned. For example, a fuzzy term like 'average' of the linguistic variable working hours per week may have an attached degree of membership of 0 to 1 within the interval [30, 35]. Membership degrees between those specified points can be calculated by interpolation.

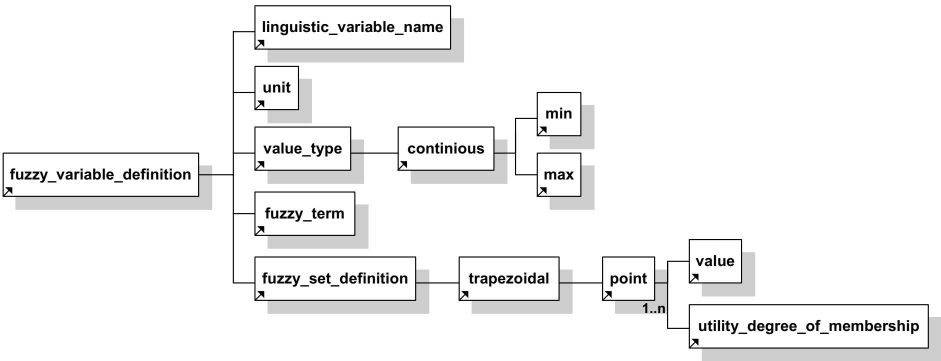

Fig. 5. Representation of a fuzzy scoring function in linear tree structure

5 Mobile and Voice User Interfaces

In this section, we describe the design and implementation of voice and mobile user interfaces for FuzzyMAN.
Mobile Negotiations (M-Negotiations) is an emerging research area that bridges the domains of Internet and Mobile Computing in order to provide time and location independent mobile negotiation services to users.
However, limited screen sizes or low data rates associated with mobile devices are some of the shortcomings which have to be addressed in this research area. Furthermore, mobile devices are equipped with different browsers that support various markup languages and media formats. Thus, the content has to be generated in different markup languages that fit the specific browsers. When delivering information it is necessary to address the diversity of mobile devices such as limited data input capabilities, small displays, limited graphical capabilities, small memory, limited processing power, etc.

Currently, we are doing research to present the XML formatted data of FuzzyMAN depending on the relevant features for different devices such as browser type, size of displays, graphical formats, etc. To realize device dependent transformations of the XML formatted data into formats such as WML and XHTML we apply XSLT stylesheets [11] and XPath [12]. XPath defines mechanisms for addressing specific elements in XML documents whereas XSLT specifies transformations on XML documents. In XSLT, rules are described for transforming the XML-based source document (also called source tree) into a new document structure (also called result tree). The transformation process is achieved by means of templates, which match some set of elements of the source document and then describe the transformation by associating patterns and styles with the templates to transform the XML-formatted data into multiple formats such as HTML, XHTML, WML, VoiceXML [13], etc.
One of our research goals in the FuzzyMAN project was to employ different technologies for implementation of the user interfaces on mobile devices, to analyze them and to compare them with each other.
FuzzyMAN's mobile user interfaces are available in the form of J2ME applications, voice user interfaces and dynamic pages displayed in different (mobile) browsers in HTML and XHTML to allow voice and mobile access to FuzzyMAN depending on the users' preferences and devices. Java 2 Micro Edition (J2ME) defines the minimum required Java technology components and libraries for small and resource constrained mobile devices. Java language, core libraries, input/output, networking, and security are the primary topics addressed by the J2ME specification [14].
In our approach, details about mobile devices such as the device manufacturer and the version of the mobile browser used are based on the HTTP headers (e.g. the UserAgent header). These HTTP headers are delivered when the user accesses the FuzzyMAN e-marketplace via the HTTP protocol. They help to automatically detect the most appropriate presentation form. Figure 6 shows the basic scheme for content presentation depending on the device type and browser.

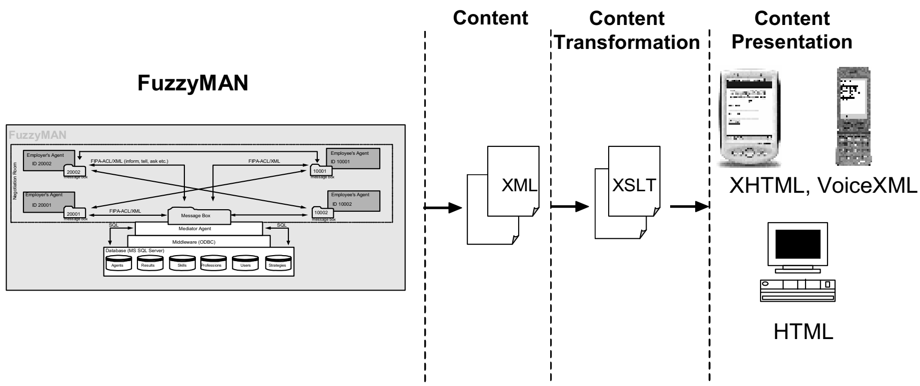

Fig. 6. Process of Content Transformation

In all cases the content requested by the user is first presented as an XML document. Using XSLT stylesheets [11] this document is then transformed into multiple formats such as HTML, XHTML or VoiceXML according to the information about the user's device and its browser. Finally, the content in the corresponding format is delivered to the user and presented on their end device.

5.1 Mobile User Interfaces

Fig. 7. FuzzyMAN's mobile front-end

Figure 7 illustrates FuzzyMAN's mobile front-end showing a selection of mobile user interfaces. Users can manage their agents through this mobile interface, including initializing, updating, and deleting agents as well as retrieving a statistical overview of the marketplace. The functionality of the mobile front-end is the same as the Web front-end which is described in section 3.1.
Figure 8 shows the presentation of FuzzyMAN's Login User Interface on a PDA.

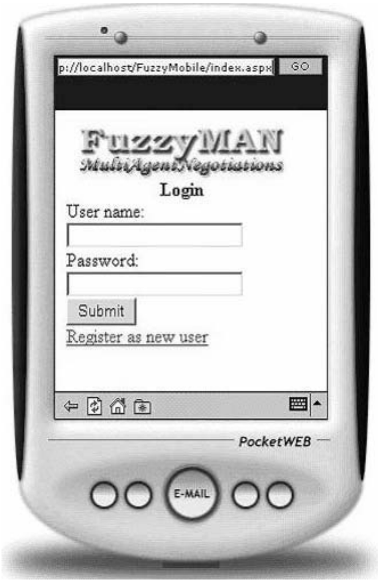

Fig. 8. PDA with FuzzyMAN's Login User Interface

Figure 9 illustrates the mobile user interface of the German version of FuzzyMAN giving a statistical overview. This user interface is developed by means of J2ME technology, which inherits Java's rich graphic libraries and thus enables developers to present statistics as diagrams. The distribution of the 5 top positions that employees' agents are searching for is represented.

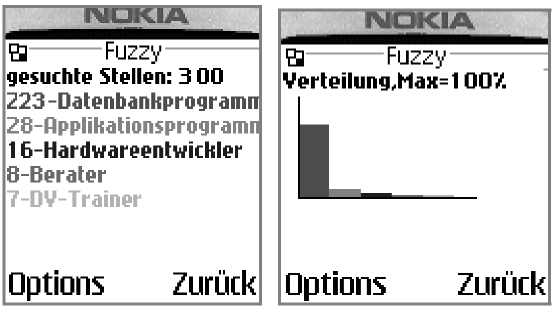

Fig. 9. Mobile User Interface with Statistical Overview

5.2 Voice User Interfaces

In this section, we describe a voice user interface for FuzzyMAN based on VoiceXML technology [2].

Voice enabled user interfaces have several advantages over traditional interfaces, for example:

- More convenient access to information on e-marketplaces in situations in which the interaction with graphical user interfaces via mobile devices is not possible (e.g. while driving a car).
- Voice enabled access to e-marketplaces is also possible by handicapped persons who are not able to use traditional user interfaces.
- Automatic voice-enabled services on e-marketplaces are available 24 hours a day and 7 days a week without the need for additional human operators (e.g. in call centers).

Automatic speech recognition technologies allow users to access e-marketplaces using their voice. A speech recognition engine processes speech as input to an application (e.g. an e-marketplace) and translates it into text. At present, even the best speech recognition engines do not achieve speech recognition rates of 100 %. Therefore, voice-based e-marketplaces have to cope with recognition errors and the dialog flow has to be considered very carefully.

A typical dialog between a user (**U**) and the FuzzyMAN e-marketplace (**F**) is illustrated in figure 10. The right window in figure 10 shows the dialog flow represented by means of VoiceXML.

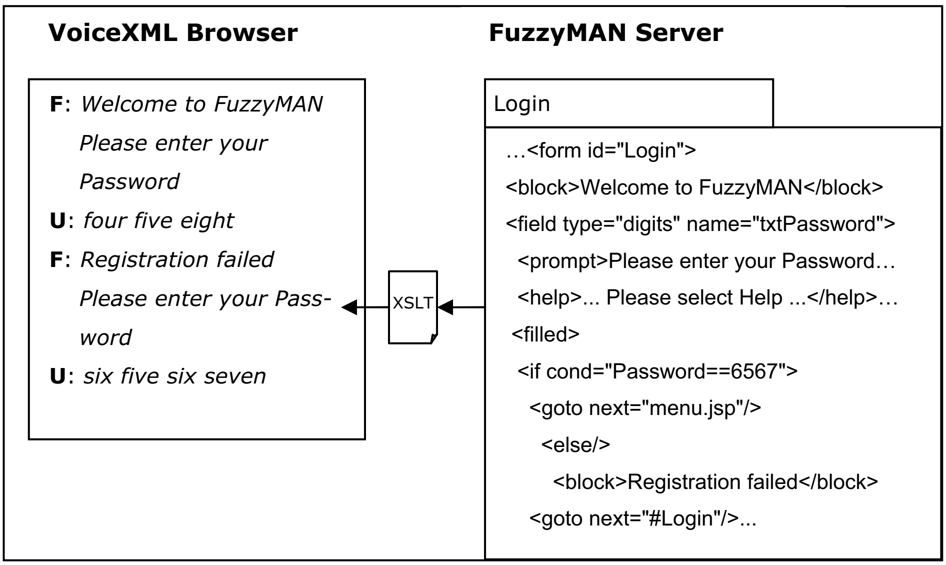

Fig. 10. Code Snippet in VoiceXML

VoiceXML documents such as a statistical overview were extracted from FuzzyMAN's database in XML format and were transformed into VoiceXML using predefined XSLT templates as illustrated in figure 11. Tags in the XML document (e.g. Dialog) are matched with templates that determine the corresponding processing of user data or the next step in the dialog flow that has to be taken.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:output method="xml" indent="yes" version="1.0" encoding="ISO-8859-1" />
- <xsl:template match="Dialog">
  - <vxml version="1.0">
    - <form id="Login">
      - <block>
          <xsl:value-of select="Title" />
        </block>
      - <field name="Password" type="digits">
        - <prompt>
            <xsl:value-of select="Form/Input[@name="txtPassword"]/@voice" />
          </prompt>
        - <help>
            <xsl:value-of select="Form/Help" />
          </help>
        - <filled>
          - <if>
            - <xsl:attribute name="cond">
                Password==
                <xsl:value-of select="Form/Input[@name='txtPassword']" />
              </xsl:attribute>
              <goto next="menu.jsp" />
              <else />
            - <prompt>
                <xsl:value-of select="Form/Error" />
              </prompt>
              <goto next="#Login" />
            </if>
          </filled>
        </field>
      </form>
    </vxml>
  </xsl:template>
</xsl:stylesheet>
```

Fig. 11. XSLT document for transformation

Figure 12 illustrates the process of a speech-based dialog with FuzzyMAN using a voice user interface via a mobile device. The user (**U**) dials the phone number of the VoiceXML interface, which answers giving the user the opportunity to choose between several options. By means of the speech recognition engine the VoiceXML interface interprets the user’s answer (e.g. "Statistical overview") and makes an HTTP request to the FuzzyMAN server where the appropriate XML document contains the statistical overview. The speech recognition engine needs so-called utterances and grammars in order to process spoken input and translate it into statements that the underlying program can execute. Utterances are streams of speech delimited by two periods of silence

(at the beginning and at the end of the stream). Grammars are finite sets of words expected as the user's spoken input.
The VoiceXML interface produces speech output based on the corresponding XML documents transformed by means of XSLT into a VoiceXML document. In order to realize the voice user interface for FuzzyMAN requests we used the IBM Voice Server Development Kit [15], which fully supports the VoiceXML standard. Additionally, grammars used as input to the Automatic Speech Recognition engine were developed.

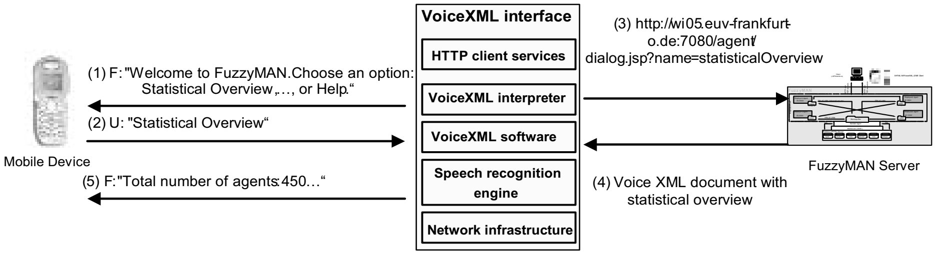

Fig. 12. Voice XML interface

5.3. Comparison of Technologies

In this section, we give a short comparison of the technologies we have applied to develop voice and mobile user interfaces for FuzzyMAN. Table 1 provides an overview of relevant criteria to compare the core technologies that can be employed to implement such user interfaces.
One of the main advantages of XHTML MP is that it enables precise positioning of text, graphics, borders, and other elements on the screen. In addition, colors can be used for better formatting of content on mobile devices with color capability.
J2ME inherits Java's rich graphic libraries and thus enables developers to significantly improve both the appearance and functionalities of the application's GUI on mobile devices. J2ME, for example, has a rich set of facilities allowing the presentation of diagrams based on statistics as described in section 5.1. Thus, J2ME is a good choice for visually representing numeric data.
VoiceXML can be used in situations where users might be simultaneously involved in several other activities (e.g. while driving a car) or where a task is complex and graphical user interfaces are not effective. However, flexibility and efficiency of using the VoiceXML technology is limited, because there is no possibility to speed up the interaction (e.g. to shorten the predefined dialog significantly).
At present, the use of voice user interfaces is not universally successful, because the dialog flow is a deterministic one. More flexible dialogs and sophisticated conversations would be desirable. Expert users have to use the voice interface in almost the same way concerning the dialog flow as the novice user.

During the test phase of the FuzzyMAN voice user interface we observed that correct voice recognition is heavily dependent upon the length of words and the similarity of words within a dialog. Therefore, it was necessary to replace similar words with less problematic synonyms (e.g. employee and employer agent replaced by buy and sell agent) to improve the recognition rate. Thus, we achieved quite good voice recognition rates of approximately 80 %.
In summary, the technology choice is heavily dependent upon the graphical aspects (e.g. diagrams have to be presented), the desired access mode (voice or mobile), and the application domain.

Criteria/Requirements	XHTML MP	J2ME	VoiceXML
Learning curve (Mobile and voice application development simplicity)	+++	+	++
Capabilities of designing graphical user interfaces	++	++	–
Ease of use (navigation through menus)	++	++	+
Extensibility	+++	++	–
Availability of code examples	++	++	++
Portability and Interoperability	++	+++	+
Flexibility and Efficiency of use	+++	++	+
Standardization of technology	W3C Standard	Proprietary standard developed and promoted by SUN	W3C Standard
Development environments (selection)	Nokia WAP toolkit, Openwave Mobile SDK	J2ME wireless toolkit, JBuilder	IBM Voice server development toolkit
Legend: +++: very good: ++: good; +: satisfactory; –: not available			

Table 1. Comparison of Technologies for Mobile and Voice User Interfaces

6 Related Work

In this section, we provide an overview of related work in the context of agent-based e-marketplaces focusing on systems that support the users' mobility.

Most of today's electronic marketplaces either do not use agent technology at all or they employ only restricted forms of this technology. For example, some online auctions can be regarded as multi-agent e-marketplaces. However, they are of a very simple type. The only issue under consideration is the price. Additional issues like terms and conditions, payment method, who will pay for shipment, timings, penalties, etc. are not taken into account. In the business-to-business area, for example, those issues play an essential role; sometimes they are even more important than the price. For such situations the agent models employed by today's marketplaces are not appropriate.

Several approaches to agent-mediated negotiation on electronic marketplaces have been introduced in the literature, see for example, [5-7] and [16-25]. In [26] a multi-agent system and analysis for multidimensional matchmaking in the human resources application domain is introduced. In [27] the negotiating environment Fuzzy eNAs (Fuzzy e-Negotiation Agents) is presented. Agents representing buyers and sellers can autonomously negotiate with fuzzy constraints and preferences in bi-lateral negotiations.

In fact, there is no single best approach or technique for automated negotiation. The negotiation strategies and protocols need to be set according to the situation and application domain. At present, the research activities in the area of agent-based e-marketplaces providing mobile and/or voice user interfaces are in an early stage [28]. We could not found any references to agent-based e-marketplaces with voice user interfaces in the literature. However, there are some agent-based e-marketplaces with mobile interfaces like Agora [29] (providing access to stationary agents via PDAs), Impulse [30], and MB (an agent-based framework for mobile commerce) [31].

Agora is a research project conducted at HP Labs where some applications of agent technology to mobile shopping are investigated. The mobile users of the Agora system can interact with the online store services through their personal digital assistants (PDA) from anywhere and anytime. The base of the Agora system is intelligent agents that represent both customers and the store. They participate in online auctions to bid for desired products based on customers' preferences. The agents have been implemented with a multi-agent system Zeus from BT Labs [32] and additional Java-based support software.

Impulse [30] is a project at MIT Media Lab that explores a scenario in which the buying and selling agents can run on wireless mobile devices and engage in multi-parameter negotiation for comparison-shopping at the point of purchase. In fact, the FuzzyMAN project uses the same approach differing only in that the point of sale is a fixed one, i.e. the FuzzyMAN e-marketplace.

MB, an agent-based framework for mobile commerce, provides a general architecture for developing applications employing mobile agents. MB agents can be of three different types. The device agent is a stationary agent that resides on a mobile device and provides access to wireless services such as location-based comparison-shopping. The service agents are owned by service providers and handle service requests from the users. The courier agents are lightweight mobile agents that can migrate from a service agent to a mobile device in order to establish communication with the user. MB has been implemented with the use of Java-based tools including Aglets SDK for service agents and KVM SDK for the device agent and courier agents.

For more examples we refer the interested reader to [28] where several other multi-agent systems using mobile agents/mobile devices are discussed and compared.

7 Conclusion, Future Work and Open Research Questions

7.1 Towards Efficiency Criteria

There is no single best negotiation protocol which is suitable for all negotiation scenarios. Standard ways to measure the efficiency of a negotiation protocol are, for example, symmetry, pareto-efficiency, and computational efficiency.

Our negotiation protocol is *symmetric*. No agent within both groups of agents (employers' and employees' agent) will be privileged with regard to others within these groups. However, the agents in both groups are treated differently. Employees' agents with low-ranking positions on the employers' agents' sequence lists must wait before they receive their first offers, because employers' agents start their simultaneous negotiation processes only with the highest ranked employees' agents. Therefore, the group of employers' agents has a superior position in our marketplace relative to the group of employees' agents. However, our proposed concept is comparable to real life employment situations. Employees also have to wait before they are invited to interviews with different employers.

A negotiation outcome is said to be *pareto-efficient* if there is no other outcome that will make at least one agent more satisfied without making at least one other agent less satisfied. Our proposed multilateral negotiation model is in most cases pareto-efficient but not in all: Assuming an employer's agent **A** receives two offers, one from employee's agent **B** (under deadline pressure) and one from employee's agent **C** (with a longer deadline) and both offers have equal utility values for the three agents **A**, **B** and **C**. In our proposed multilateral negotiation model, employer's agent **A** will choose the first offer in its message box. With regard to pareto-efficiency it would be better to choose the offer from employee's agent **B**, which has the earlier deadline, because the employee's agent **C** (with the longer deadline) has a better chance to reach an agreement before its deadline with another employer's agent and to achieve the same or even a better utility value. In the case that agent **A** chooses employee's agent **B** it is possible that agent **B** failed and due to its deadline pressure reach a utility value of 0 (e.g. no agreement). So in this special negotiation scenario there could be an outcome that will make employer's agent **A** and employee's agent **C** just as satisfied while making employee's agent **B** (under deadline pressure) more satisfied. However, in our multilateral negotiation protocol other agents' deadlines (as well as their tactics) are private information (as in real life negotiation situations for jobs). Therefore, we had to make a decision between pareto-efficiency in all negotiation scenarios and information privacy.

Agents are not *cooperative* or *benevolent* on our job marketplace, because each agent wants to get the agreement which satisfies its preferences best. If the negotiation tactic of agent **A** is known to agent **B**, agent **A** may be at a significant disadvantage.
An agent could get a better deal when putting negotiation partners under deadline pressure by not revealing his true negotiation deadline. However, on our job marketplace an employee's agent, for example, is unaware of the number of employees' agents simultaneously negotiating with one of its negotiation partners. Due to this lack of information agents on our marketplace reply to an offer immediately.
Concepts for concealing the agents' true preferences and deadlines or selfish negotiation behavior are not implemented yet. In this way the negotiation model can be extended. However, a negotiation model supporting selfish agents is not necessarily optimal with respect to reaching the best negotiation outcome for both negotiation parties (employees' and employers' agents) in terms of pareto-efficiency.
We have conducted a set of simulation experiments to evaluate our negotiation model.
The values and parameters for our simulation experiments were generated automatically, because we had no real employees and employers with real data available. Since there are infinitely many potential negotiation situations in which we can evaluate our negotiation model, we had to limit the possible values of the agents' utility functions for each issue. Therefore, we generated these values automatically based on empirical statistics. For example, to generate values for the issue working hours per week, we used empirical statistics from Germany's Federal Statistical Office, describing the distribution of German employees' working hours per week. 900 agents were initialized for our simulation experiments.
The main experimental results can be summarized as follows:

- The utility degrees of agreements achieved by agents only shortly before the end of their negotiation time (deadline) are on an average only 80 % of the utility degrees of agreements achieved by agents just at the beginning of their negotiation processes.
- Agents willing to make concessions early at the beginning of their negotiation processes have the best chance to achieve an agreement. Only a few deals were made when at least one negotiation partner was not willing to make concessions or both negotiation partners have applied divergent tactic functions.
- 'Extreme' tactical behavior (not willing to make concessions) results in a higher standard deviation with regard to achieved utility degrees. Agents unwilling to make concessions obviously have a higher risk of being unable to achieve an agreement during their disposable negotiation time. On the other side, when these agents achieve agreements the observed utility degrees were 20 % above the average utility degrees.

For more statistical results we refer the interested reader to [33] where simulation results of the FuzzyMAN marketplace were presented.
FuzzyMAN is a complete, robust, and working prototype of an agent-based electronic marketplace. In October 2002 the FuzzyMAN marketplace was made available for the scientific community in the Internet. Initially 200 employees' and 100 employers' agents were generated. Since then, 154 new agents were added by anonymous users from the scientific community to the FuzzyMAN marketplace. The current negotiation

statistics are as follows: 116 out of 454 agents on our marketplace were successful and achieved agreements (contracts) before they reached their deadlines. The average utility value (in %) of agreements achieved by successful agents is 65 %. This low average utility value and low rate of matched agents is due to the divergent agent profiles initialized by their users. Most employees' agents failed in finding negotiation partners because the employers' agents looked for employees with different professional backgrounds.

7.2 Future Work and Open Research Questions

During the test phase of FuzzyMAN a number of starting points for improvements of its performance were observed. The following is a summary of these points:

Multimodality: At present, a mobile user has the possibility to interact with FuzzyMAN via either speech, web, or mobile browser. The next step to be considered in future work is multimodal access to the FuzzyMAN e-marketplace. Multimodal access means that content (e.g. a statistical overview) is displayed on the (mobile) browser and at the same time delivered and explained via speech. The main advantages of multimodal access via one (mobile) user interface are a better understanding and memorization of the delivered information [34].

Natural Language Understanding: The voice user interface presented in this paper is not capable of free speech input. The user can only choose one of the valid predefined responses specified by the corresponding grammar. More sophisticated approaches like Natural Language Understanding [35] could accelerate and improve the user interaction with regard to a natural way of communication.

Personalization: Personalization usually means adjustment of system functionalities to individual preferences. Adjustment of content display, output formatting, choice of exposed services, or creation of personal views on services that FuzzyMAN provides are some typical examples of personalization. By applying *XSL:FO (*XSL Formatting Objects), for example, it would also be possible to transform XML documents into formats such as PDF, RTF, SVG, etc., which are currently not supported by FuzzyMAN. More complex personalization concepts would require acquisition of the users' personal preferences and characteristics, e.g. competences, interests, preferred devices and browsers, or other relevant information with the aim to utilize them by interacting with FuzzyMAN. To take advantage of this extended personalization concept FuzzyMAN could use information stored in either static user profiles, which have been created based on the users' input, or in dynamic profiles, for which information is gathered through permanent observation of the users' behavior within the system. FuzzyMAN could use these profiles to personalize services according to user preferences.

Session management: Another concept closely related to personalization is the management and tracking of state and context of users' interaction with FuzzyMAN in the form of 'personalized' session management. FuzzyMAN provides mobile and voice-enabled access to its services. Thus, maintaining state spanning multiple connections is very important to avoid loss of information by almost uncontrollable breaks in wireless connections (e.g. by passing through a tunnel, transferring from one wireless hot-spot to another). It is personalized and persistent management of interaction state and context that would allow FuzzyMAN's users to resume their work at the next available opportunity.

Peer-to-Peer mobile communication infrastructure: FuzzyMAN's agents are not lightweight enough to directly operate and negotiate on mobile devices. At present, there are too many problems which have to be solved to realize a scenario of mobile agents migrating between a Peer-to-Peer communication infrastructure of mobile devices and negotiating on the mobile devices. In such a scenario, a central server would no longer pose a bottleneck. Furthermore, user sensitive data (e.g. user profiles/preferences) would be stored on various mobile devices and not in one central database, where such data could be misused by violating privacy issues. Issues that still have to be solved to realize such a scenario are, for example, security, bandwidth and performance issues. Furthermore, the heterogeneity and diversity of mobile devices must also be addressed.

FuzzyMAN is available at http://www.wiwi.euv-frankfurt-o.de/wi-www/agent.htm. Interested readers are welcome to create their own agents, send them out to negotiate, watch their behavior, and see if they are successful in their negotiations.

Acknowledgements

We are very grateful to the German Research Foundation that supported our work through grant no. Ku 569/15-1 and to Karol Fil for useful discussions concerning our implementation of voice and mobile user interfaces.

References

[1] SALT Forum, *Speech Application Language Tags 1.0 Specification*; available at: http://www.saltforum.org/saltforum/downloads/SALT1.0.pdf, 2003.

[2] VoiceXML Forum, *Voice eXtensible Markup Language*; available at: http://www.voicexml.org/specs/VoiceXML-100.pdf, 2003.

[3] FIPA – Foundation for Intelligent Physical Agents. *FIPA Specifications*, 2003; available at: http://www.fipa.org/specifications/index.html, 2003.

[4] Bichler, M. and Kalagnanam, J., *Bidding Languages and Winner Determination in Multi-attribute Auctions*, 2004. To appear in: European Journal of Operational Research. (IBM Research Report RC22478); http://ibis.in.tum.de/staff/bichler/docs/RC22478.pdf.

[5] Kurbel, K. and Loutchko, I., *Multi-agent negotiation under time constraints on an agent-based marketplace for personnel acquisition*. In: Dittmar, T., Franzzyk, B., Hofmann, R. et al. (Eds.): Proceedings of the 3rd Intern. Symposium on Multi-Agent Systems, Large Complex Systems, and E-Business (MALCEB), 2002, 566-579.

[6] Faratin, P., Sierra, C., and Jennings, N. R., *Negotiation decision functions for autonomous agents.* Robotics and Autonomous Systems **24** (1998), 159-182.

[7] Fatima, S., Wooldridge, M., and Jennings, N. R., *Multi-issue negotiation under time constraints*. In: Proc. 1st Int. Joint Conf. On Autonomous Agents and Multi-Agent Systems, 2002, Bologna, Italy, 143-150.
[8] Nwana, H. S. and Ndumu, D. T., *A perspective on software agent research*. The Knowledge Engineering Review **14** (1999) 2, 125-142.
[9] Glushko, R. J., Tenenbaum, J. M., and Meltzer, B., *An XML Framework for Agent-based E-commerce*. CACM **42** (1999) 3, 106-114.
[10] Zadeh, L., *Fuzzy Sets. Journal of Information and Control* **8** (1965), 338-353.
[11] W3C, *XSL Transformation*, Version 1.0; available at: http://www.w3.org/TR/xslt, 2003.
[12] W3C, *XPath*; available at: http://www.w3.org/TR/xpath, 2002.
[13] W3C, Voice *Extensible Markup Language (VoiceXML)*, Version 2.0; available at: http://www.w3.org/TR/voicexml20/, 2003.
[14] Sun Microsystems, *Java 2 Platform, Micro Edition* (2004); available at: http://java.sun.com/j2me, 2004.
[15] IBM, *Voice Server Software Development Kit*, (2002); available at: http://www-3.ibm.com/software/voice/, 2002.
[16] Adams, G., Rausser, G., and Simon, L., *Modelling Multilateral Negotiation: An Application to California Water Policy*. J. Economic Behaviour and Organization **10** (1996), 97-111.
[17] Aknine, S., Pinson, S., and Shakun, M. V., *An extended multi-agent negotiation protocol.* International Journal on Autonomous Agents and Multi-Agent Systems **8** (2004), 5-47.
[18] Anthony, P. and Jennings, N. R., *Evolving Bidding Strategies for Multiple Auctions.* In: van Harmelen, F. (Ed.), Proc. 15th European Conf. Artificial Intelligence, IOS Press: Amsterdam, 2002, 178-182.
[19] Faratin, P., *Automated Service Negotiation Between Autonomous Computational Agents*. PhD thesis, Univ. of London, Queen Mary College, 2000.
[20] Faratin, P., Sierra, C., and Jennings, N. R., *Using similarity criteria to make trade-offs in automated negotiations*. Artificial Intelligence **142** (2002), 205-237.
[21] Fatima, S., Wooldridge, M., and Jennings, N. R., *An agenda based framework for multi-issues negotiation*. Artificial Intelligence Journal **152** (2004), 1-45.
[22] Jennings, N. R., Faratin, P., Lomuscio, A. R., Parsons, S., and Wooldridge, M., *Automated negotiation: prospects, methods and challenges*. Group Decision and Negotiation **10** (2001), 199-215.
[23] Kraus, S., *Strategic negotiation in multi-agent environments*. The MIT Press, Cambridge, Massachusetts, 2001.
[24] Raiffa, H., *The art and science of negotiation*, Harvard University Press, 1982.
[25] Sandholm, T., *Agents in electronic commerce: component technologies for automated negotiation and coalition formation*. Autonomous Agents and Multi-Agent Systems **3** (2000), 73-96.
[26] Veit, D., *Matchmaking in Electronic Markets: An Agent-Based Approach towards Matchmaking in Electronic Negotiations*. Lecture Notes in Artificial Intelligence, Vol. 2882, Springer: Berlin, 2003.
[27] Kowalczyk, R. and Bui, V., *On Fuzzy e-Negotiation Agents: Autonomous negotiation with incomplete and imprecise information*. DEXA Workshop on e-Negotiation, UK, 2000.
[28] Kowalczyk, R., Ulieru, M., and Unland, R., *Integrating Mobile and Intelligent Agents in Advanced e-Commerce: A Survey*; available at: http://citeseer.nj.nec.com/551656.html, 2002.
[29] Fonseca S., Griss M., Letsinger, R., *An Agent-Mediated E-Commerce Environment for the Mobile Shopper.* HP Technical Report HPL-2001-157, 2001.

[30] Maes, P., Youll, J., and Morris, J., *Impulse: Location-based Agent Assistance*; in: Software Demos, Proc. of the 4th Int. Conf. on Autonomous Agents, June 2000; available at: http://citeseer.nj.nec.com/youll00impulse.html.
[31] Mihailescu, P. and Binder, W.: A Mobile Agent Framework for M-Commerce. Agents in E-Business (AgEB-2001), Workshop of the Informatik 2001, Vienna, Austria, 2001; available at: http://www.ifi.unizh.ch/events/GI/GI01/Proceedings/mihailescu.pdf.
[32] Nwana, H. S., Ndumu, D. T., and Lee, L. C., *ZEUS: A Tool-Kit for Building Distributed Multi-Agent Systems*. Applied Artificial Intelligence Journal **13** (1999), 129-186.
[33] Teuteberg, F., *Experimental Evaluation of a Model for Multilateral Negotiation with Fuzzy Preferences on an Agent-based Marketplace*. In: Müller, J., Maass, W., Schmid, B., and Pavlikova, L. (Eds.): Electronic Markets: The International Journal of Electronic Commerce & Business Media **13** (2003) 1, 21-32.
[34] Jankowska, A. M. and Dabkowski, A., *Voice User Interfaces for Mobile ERP System*. In: Branki, C. et al. (Eds.), MKWI 2004, Vol. 3, Duisburg, Essen, 2004, 44-53.
[35] IBM, *An Introduction to IBM Natural Language Understanding*, IBM White Paper, 2001; available at: http://www-3.ibm.com/software/voice/.

Frank Teuteberg
Research Center for Information Systems in Project and Innovation Networks (ISPRI)
c/o Department of Business Administration/E-Business and Information Systems
University Osnabrueck
Katharinenstr. 1, D-49074 Osnabrueck, Germany
e-mail: frank.teuteberg@uos.de

Iouri Loutchko
Department of Business Informatics
European University Viadrina Frankfurt (Oder)
POB 1786, D-15207 Frankfurt (Oder), Germany
e-mail: loutchko@uni-ffo.de

Information about Software

Software is available on the Internet as

(X) prototype version
( ) full fledged software (freeware), version no.:
( ) full fledged software (for money), version no.:
( ) Demo/trial version
( ) not (yet) available

Description of software:

FuzzyMAN is an implementation of an electronic marketplace where software agents negotiate, buy, and sell. The marketplace is open for public access. That is, you can create your own agents, provide them with your personal preferences, send them out to negotiate, and watch what they are doing. The application domain we chose is buying and selling "labour". Agents act for employers (looking for employees) and employees (looking for jobs). Negotiation models cover both bilateral and multilateral negotiations about multiple issues, based on fuzzy logic. XML is used as the agent communication language. User interfaces for accessing information on the e-marketplace are web-, voice and mobile interfaces.

Access address:

http://www.wiwi.euv-frankfurt-o.de/wi-www/agent.htm

Contact person for question about the software:

Name: Frank Teuteberg

email: frank.teuteberg@uos.de