

# Agent Evaluations for Role Assignment

Haibin Zhu, *Senior Member, IEEE* and Matthew Grenier  
*Department of Computer Science and Mathematics, Nipissing University*  
*100 College Drive, North Bay, Ontario, P1B 8L7, Canada*  
E-mail: [haibinz@nipissingu.ca](mailto:haibinz@nipissingu.ca)

**Abstract**—Role assignment is an important task in Role-Based Collaboration (RBC). Agent evaluation is the first step of role assignment. The other two steps are group role assignment and role transfer. This paper clarifies the requirement of agent evaluations, defines the structure of roles and agents, and designs and implements an initial evaluation method. Furthermore, it proposes the requirement to extract information from resumes, present the method of information extraction, and demonstrates an example. At last, it outlines the contributions of this paper and points out the future work.

**Index Terms**—roles, agents, role assignment, agent evaluation, resume information extraction, and role-based collaboration.

## I. INTRODUCTION

Job hunting and recruiting are a regular activity for people and companies to take in modern societies. Efficient job hunting and recruiting tools can attract applicants and give them easier access for identifying employment opportunities, and assist people to quickly find good jobs and companies to increase their advantages among their competitors [8]. To organize a software development team, it is important to evaluate the staff's qualifications so as to assign relevant tasks (roles) to appropriate people. Job hunting, recruiting and evaluation of staff can be translated into the terms *role/agent matching* or *agent evaluation* in Role-Based Collaboration (RBC).

RBC is an emerging methodology to facilitate an organizational structure, provide orderly system behavior, and consolidate system security for both human and non-human entities that collaborate and coordinate their activities with or within systems [12]. The life cycle of RBC includes three major tasks: *negotiate roles*; *assign roles and play roles* [15]. Therefore, role assignment is an important aspect of RBC. It largely affects the efficiency of collaboration and the degree of satisfaction of the members involved in the collaboration.

To be more understandable, role assignment can be categorized into three parts: *agent evaluation*, *group role assignment*, and *role transfer*.

*Agent evaluation* is used to rate the qualification of an agent for a role. It is required to check the capabilities, experiences, and credits of agents based on role specifications.

*Group role assignment* [14] is used to initiate a group by assigning roles to agents, the group members, to obtain the group's highest performance based on the agents' qualifications which are the results of agent evaluation.

*Role transfer* (also called *dynamic role assignment*) is used to re-assign roles to agents or transfer roles of agents [13, 16] to meet the requirement of the system changes.

In the business world, a person is required to have new knowledge, skills and habits to be qualified for a new position. Successfully finding a new job is dependent on similarities between new roles and those previously performed by a person [3], i.e., qualifications are the basic requirements for possible role-related activities. Agent evaluation is a fundamental problem that requires advanced methodologies, such as information classification, data mining, pattern searching, and matching, to produce high-quality evaluations.

Although agent evaluation is evidently an important problem in management [3], human resource [5, 8], organizational behavior and performance [1, 2, 3], scheduling, training and commanding [9], there is no fundamental research on such problems. Most job hunting and recruiting tasks are done manually without computer-based tools to analyze the applications and provide recommendations.

This paper concentrates on the first part of role assignment, i.e., *agent evaluation* and contributes a set of methods and computer-based tools to support the accomplishment of this task. The assumption is that a set of resumes from agents and a set of role requirements are ready. The result is a matrix of qualifications of agents for each role.

The rest of this paper is arranged as follows: Section 2 discusses related work; Section 3 revises the E-CARGO model in order to describe the problem of agent evaluation; Section 4 discusses the templates of agents and roles, the basic formula for evaluation, and an evaluation result for a group of agents and roles; Section 5 presents how to extract agent information from resumes, and

Section 6 concludes this paper and points out the future work.

## II. RELATED RESEARCH

Agent evaluation is rarely found in the literature of agent field. Moore et al. [6] mentioned such problems but demonstrated idea in a different way. They discussed a list of problems in selecting agents to execute a specific task. They presented initial thoughts by considering roles and capabilities and but failed to fully clarify the problem of agent evaluation.

Some other research related with this work is sparingly recognized in the field of information extraction. Several resume extraction tools exist and claim to accurately extract data. However, no technical details of the algorithms or processes used are available. There are not publications found on agent evaluation for resume extraction which is our goal here.

Yu et al. [11] present an effective approach for resume information extraction to support automatic resume management and routing. They design a cascaded information extraction framework: in the first pass, a resume is segmented into a consecutive blocks attached with labels indicating the information types and then in the second pass, the detailed information. Their goal is slightly different, i.e., there is no predefined template of specific data they are interested in. They simply attempt to extract all information for categorization and classification. The problem is that their methods extract most objective information such as *name, address, phone number, and email addresses* that are not very useful to evaluate agents' qualifications.

Ciravegna and Lavelli [4] applies  $(LP)^2$  in a toolkit of information extraction, to learn information extraction rules for resumes written in English. The information defined in their task includes a flat structure of *Name, Street, City, Province, Email, Telephone, Fax and Zip code*. Similarly, their method is not appropriate for this paper's requirement of agent evaluation.

Sitter and Daelemans [7] perform information extraction on any document using a double classification method which first identifies where relevant data may be located, then does an in-depth analysis of those areas of interest for specific data to extract.

Wu et al. [10] propose the problem of resume mining of social network communities. Their aim is in the study of three aspects: the characterization of community, the discrimination among communities, and the community evolution mining. Their mining problem is totally different from the problem discussed in this paper. Their solution is not what we need in agent evaluations.

Their research shows that there are indeed strong needs to investigate fundamentally agent evaluation problems. Their work also demonstrates the importance of the work depicted in this paper.

## III. REVISED E-CARGO MODEL

With the E-CARGO model [12-16], collaboration is based on roles. In E-CARGO, a system  $\Sigma$  can be described as a 9-tuple  $\Sigma ::= \langle C, O, \mathcal{A}, \mathcal{M}, \mathcal{R}, \mathcal{E}, \mathcal{G}, s_0, \mathcal{H} \rangle$ , where  $C$  is a set of classes,  $O$  is a set of objects,  $\mathcal{A}$  is a set of agents,  $\mathcal{M}$  is a set of messages,  $\mathcal{R}$  is a set of roles,  $\mathcal{E}$  is a set of environments,  $\mathcal{G}$  is a set of groups,  $s_0$  is the initial state of a collaborative system, and  $\mathcal{H}$  is a set of users. In such a system,  $\mathcal{A}$  and  $\mathcal{H}$ ,  $\mathcal{E}$  and  $\mathcal{G}$  are tightly-coupled sets. A human user and his/her agent play a role together. Every group should work in an environment. An environment regulates a group. With this tight coupling, it is emphasized that a role-based collaborative system is composed of both computers and human beings.

With the participation of people  $\mathcal{H}$ , such as joining in a team  $\Sigma$ , accessing objects of the team, sending messages through roles, forming a group in an environment,  $\Sigma$  evolves, develops and functions. The results of the team work are a new state of  $\Sigma$  that is expressed by the values of  $C, O, \mathcal{A}, \mathcal{M}, \mathcal{E}, \mathcal{G}$ , and  $\mathcal{H}$ . For this paper, roles and agents are mainly discussed.

**Definition 1:** *role*. A role is defined as  $r ::= \langle n, q_e, q_d, q_m, Q_o, Q_l, Q_s, Q_{op}, Q_{pl}, Q_3 \rangle$  where,

- $n$  is the identification of the role;
- $q_e$  is a level of education (degree) requirement;
- $q_d$  is the major required;
- $q_m$  is a math level required;
- $Q_o$  is the experience requirement that is a set of roles required to be played before playing this role;
- $Q_l$  is a set of language requirements;
- $Q_s$  is a set of software application requirements;
- $Q_{op}$  is a set of operating systems requirements;
- $Q_{pl}$  is a set of programming language requirements; and
- $Q_3$  is a set of other skills required to play this role.

**Definition 2:** *agent*. An agent is defined as  $a ::= \langle n, p_e, p_d, p_m, P_o, P_l, P_a, P_{op}, P_{pl}, P_s \rangle$ , where

- $n$  is the identification of the agent;
- $p_e$  is the degree of the agent;
- $p_d$  is the major of the agent;
- $p_m$  is the math level of the agent;
- $P_o$  is a role set (as experience) the agent has played before;
- $P_l$  is a set of languages the agent understands;
- $P_a$  is a set of software applications the agent is familiar with;
- $P_{op}$  is a set of operating systems the agent is familiar with;
- $P_{pl}$  is a set of programming languages the agent is familiar with; and

- $\mathcal{P}_s$  is a set of other skills the agent has.

**Definition 3:** *agent qualification matrix.* Suppose  $\mathcal{A}$  denotes the set of agents and  $m (=|\mathcal{A}|)$  the size of the set  $\mathcal{A}$ .  $\mathcal{R}$  denotes the set of roles and  $n (=|\mathcal{R}|)$  the size of  $\mathcal{R}$ .  $Q$  is an *agent qualification matrix* of  $m \times n$ , where,  $Q[i, j] \rightarrow [0, 1]$  ( $0 \leq i \leq m-1$ ;  $0 \leq j \leq n-1$ ) be the value that represents how well a given agent  $i$  plays a given role  $j$ , 0 means the lowest and 1 the highest.

In the problems of agent evaluation,  $\mathcal{A}$  and  $\mathcal{R}$  are taken as the inputs and  $Q$  the output.

**Definition 4:** *agent evaluation.* Suppose that there are one set of agents  $\mathcal{A}$  and one set of roles  $\mathcal{R}$ . The problem is to find an agent qualification matrix  $Q$ .

For example, Fig. 1 shows an agent qualification matrix  $Q$  with 6 agents and for roles.

$$\begin{bmatrix} 0.71 & 0.6 & 0.0 & 0.22 \\ 0.29 & 0.67 & 0.44 & 0.76 \\ 0.69 & 0.92 & 0.92 & 0.6 \\ 0.0 & 0.0 & 0.53 & 0.0 \\ 0.97 & 0.51 & 0.77 & 0.65 \\ 0.58 & 0.64 & 0.24 & 0.0 \end{bmatrix}$$

**Fig. 1. An agent qualification matrix.**

Agent evaluation can be further divided into two steps: 1) agent formation; and 2) agent role matching.

**Definition 5:** *agent formation.* Given an input resume  $V$  which is a sequence of words  $w_1, w_2, \dots, w_k$ , the result of agent formation is a filled agent structure in

**Definition 2.**

#### IV. AGENT EVALUATION

Based on the above definitions, the major task of agent evaluation is to match the properties of agent  $a$  with the required skills and experiences by role  $r$  and assign a number between 0 and 1, where 0 means that  $a$  is not qualified to play  $r$ , and 1 means that  $a$  is perfectly match the requirement of  $r$ .

The basic algorithm is to obtain a rate between an agent and a role and the following formula is used.

$$Q[i, j] = \sum_{k=0}^8 w[k](j) * v[k](i, j).$$

Where,

- $0 \leq i \leq m-1, 0 \leq j \leq n-1, 0 \leq k \leq 8$ ;
- $w[k](j) \rightarrow [0, 1]$  is the weight for the education ( $k = 0$ ), major ( $k = 1$ ), math ( $k = 2$ ), experience ( $k = 3$ ), language ( $k = 4$ ), software application ( $k = 5$ ), operating system ( $k = 6$ ), programming language ( $k=7$ ), and other skills ( $k = 8$ ) requirement of role  $j$ ;
- $v[k](i, j) \rightarrow [0, 1]$  is a rate for the education ( $k = 0$ ), major ( $k = 1$ ), math ( $k = 2$ ), experience ( $k = 3$ ), language ( $k = 4$ ), software application ( $k = 5$ ), operating system ( $k = 6$ ), programming language

( $k = 7$ ), and other skills ( $k = 8$ ) of agent  $i$  for role  $j$ ; and

- $\sum_{k=0}^8 w[k] = 1.$

**Fig. 1. A role requirement.**

In the above formula,  $w[k]$  ( $0 \leq k \leq 8$ ) can be justified to accommodate special considerations in a group. To accomplish the above task, we need to define an available agent template, a role template, and the editors for agents and roles (Figs. 1, 2 and 3).

To obtain the rate of each items stated above, i.e.,  $v[k](i, j)$ , the following criteria is used:

$k = 0$ :

$$v[k](i, j) = \begin{cases} 1 & \text{if } i.q_e = \text{PhD}; \\ 0.8 & \text{if } i.q_e = \text{MSc, MA or MEd}; \\ 0.6 & \text{if } i.q_e = \text{BSc, BA or BED}; \\ 0.4 & \text{if } i.q_e = \text{Diploma}; \\ 0 & \text{others.} \end{cases}$$

$k = 1$ :

$$v[k](i, j) = \begin{cases} 1 & \text{if } i.q_e = i.p_e; \\ 0 & \text{others.} \end{cases}$$

$k = 2$ :

$$v[k](i, j) = \begin{cases} 1 & \text{if } i.q_m = \text{University}; \\ 0.8 & \text{if } i.q_m = \text{College}; \\ 0.6 & \text{if } i.q_m = \text{High School}; \\ 0.4 & \text{if } i.q_m = \text{Middle School}; \\ 0.2 & \text{if } i.q_m = \text{Elementary School}; \\ 0 & \text{others.} \end{cases}$$

$k = 3, 4, 5, 6, 7, 8$ :

$$v[k](i, j) = \frac{\text{the number of } i.Q_o(Q_1, Q_a, Q_{op}, Q_{pl} \text{ or } Q_s)}{\text{the number of } j.P_o(P_1, P_a, P_{op}, P_{pl} \text{ or } P_s)}.$$

Fig. 2. An agent qualification.

For example, suppose  $w = [0.3f, 0.05f, 0.1f, 0.1f, 0.1f, 0.05f, 0.1f, 0.15f, 0.1f]$ , the qualification of *agent 0* (Fig. 2) for *role 0* (Fig. 1) is 86.25%.

Fig. 3. Editing a role requirement.

Fig. 4. A group with 6 agents and 4 roles.

0.9250	0.7693	1.0000	1.0000
0.9000	0.7511	1.0000	1.0000
0.7800	0.7709	0.8250	0.9250
0.6650	0.4947	0.9500	0.9250
0.7100	0.5736	0.8250	0.8250
1.0000	0.6488	1.0000	1.0000

Fig. 5. An agent qualification matrix created for 6 agents and 4 roles.

Fig. 3 shows the editor to edit *role 0*, where the requirements of the role are chosen from the provided list.

Fig. 4 is a group with 6 agents and 4 roles. Fig. 5 is the agent qualification matrix created by the tool that applies the evaluation methods discussed above.

## V. RESUME INFORMATION EXTRACTION

Section 4 presents the evaluation of an agent for a role based on formally pre-defined templates of roles and agents. The information of agents must be edited by a special editor and must be operated by a human user to edit. This is a time consuming and boring job. Automation based on some text mining technologies is required.

After analyzing the format of ordinary resumes, our first attempt is to use Hypertext Markup Language (HTML) files as our source files to be extracted. This is reasonable because most resumes can be transformed into the HTML format. From common methods of text mining, classification and data extraction are the major two steps [7, 8, 11].

### A. Agent Definition

The goal of this work is to populate relevant fields regarding agent's abilities. An agent can be thought of as a person. In order to do this we must define the possible properties an agent can possess. In the current RBC tool agent abilities are based on 3 fields: education (including major), experience, and skills (including math, language, software application, operating system, programming language, and others).

A template is developed which can be populated according to the specific abilities of the input resume. First it is necessary to define the set of attributes  $A = \{a_1, a_2, \dots, a_m\}$  we are interested in. Each attribute  $a_i$  has a set of properties  $P_i = \{p_{i1}, p_{i2}, \dots, p_{in}\}$  describing specific abilities or experiences which an agent possesses (Section 4). This data can be customized for different uses.

### B. Classifying Data

In order to obtain an accurate extraction of the data of a resume, it must be classified into standard resume categories. The categories used in this work have been created and are stored in text files for easy alteration and

customization. Different people have different writing styles and they may use different adjectives to describe these general sections, a list of categories containing synonyms used in resumes is compiled (Fig. 6). These synonyms are used to locate category headings.

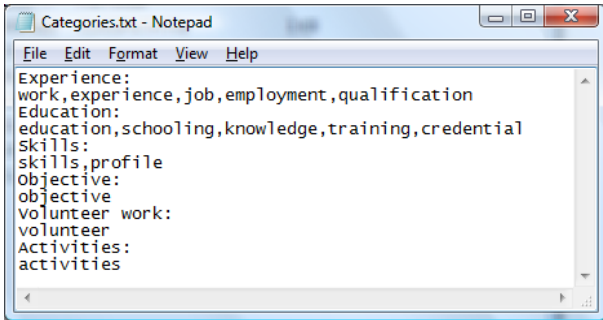


Fig. 6. The sample categories.

The plain text resume is searched for headings. An extra measure is added here to help eliminate sentences which may contain heading keywords. The line is only considered a heading if it has less than a number of characters in length (50 in this work). Generally, the fact that headings are shorter can increase the accuracy of the extraction process. If the text is deemed to be a category, the heading from the resume is stored.

Originally, the HTML document is used to search for keywords in the category file (Fig. 6). This is done to take advantage of the fact that many headings are encapsulated in tags such as `<H1></H1>` or `<B></B>`. Currently this tactic is no longer being employed because in certain cases misdiagnoses of headings was experienced. This is an excellent way to more accurately distinguish between headings and attribute text, but more time is needed to better implement this.

The headings from the resume are stored and not the keywords from the template that was found, this is done to achieve greater accuracy when searching through the plain text resume to categorize the data. Category headings are generally found on their own line that is usually unique in the resume, that is, another line with only that text usually does not occur. Thus by storing the title on the resume, which could be “Skill Summary” representing the skill section, would accurately represent the beginning of the skills section, where as the single word skill may be found in multiple places. The word skill may be found throughout the resume in other area, searching just the word hit on in the plain text to define the starting point of the skill section could produce undesired effects.

### C. String Extraction

The extracting of information is implemented in Java, this language was chosen for two reasons. First, it is the language the role transfer tool [13] is written in, thus

allowing easy integration of the products. Secondly, it offers a String extraction library which can be utilized to remove all tags and formatting data from the original HTML file. The library, called *HTMLParser*, offers a *StringExtractor* class. The *StringExtractor* takes a resource, which in our case is the name of the resume file. Once the extractor is created we use the *extractString()* method to store only the String information from the HTML and store it. We are left with a plain text resume.

The document can now be scanned for the category headings known to be in the document and all data from the current heading to the next heading found is stored as data relevant to the preceding category. This allows for more accurate classification of the data. For example, if the words *Civil Engineering* appear in the education field we know this refers to schooling and does not necessarily mean the agent has experience as a civil engineer.

### D. Item Extraction

Our extraction model for personal information is based on two assumptions:

- 1) Personal information is stored at the beginning of the resume. This is normal on most resumes, and since there are generally no keywords to identify personal information we will take advantage of this fact.
- 2) The data is in a standard format. That is, the personal information appears in order:

```

<Name>
<Address>
<Phone Number>
<Email Address>
  
```

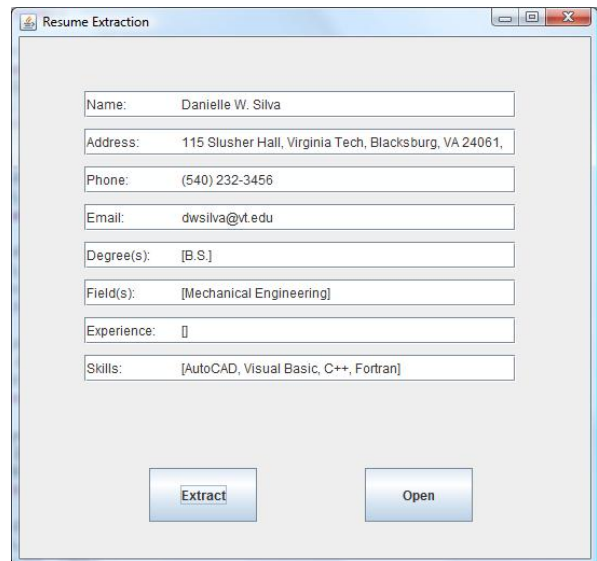


Fig. 7. The Information Extracted by the Resume Parser.

# Danielle W. Silva

Current Address:  
115 Slusher Hall, Virginia Tech  
Blacksburg, VA 24061  
(540) 232-3456  
[dwsilva@vt.edu](mailto:dwsilva@vt.edu)

Permanent Address:  
1234 Forest Lane  
Frederick, MD 20872  
(301) 555-6677

**OBJECTIVE** Cooperative Education position related to manufacturing

**EDUCATION** **B.S. Mechanical Engineering**, Expected graduation as a co-op: May 2004  
**Minor: Statistics**  
Virginia Polytechnic Institute & State University (Virginia Tech), Blacksburg, VA  
GPA: 2.8/4.0  
Earning and financing 50% of college education and expenses

**COMPUTER SKILLS** **Software:** AutoCAD MiniTab TK solver Mathematica **Languages:** Fortran Visual Basic PowerC C++

**EXPERIENCE** **Waiter**, Boudreaux's, Blacksburg, VA Fall 2002 – present

- Work 20 hours per week to help fund college education.

**Self-employed**, Lawn care business, Frederick, MD Summer 2002

- Solicited business through cold calls and visits.
- Increased customer base through referrals for quality work.
- Performed all necessary equipment maintenance and repairs.
- Covered all expenses and yielded net profit of \$4000.

**Server**, Family Restaurant, Frederick, MD, 1999 – 2001, High school & summers

- Trained on and used excellent customer service practices in fast-paced, work environment.
- Participated in corporate program to increase sales; achieved all sales goals.
- Worked 40 or more hours per week in summers; worked 15 hours per week during school year.

**ACTIVITIES** Student Engineers Council (SEC), Membership Committee Chair, 2001-02  
Symphonic Band, Virginia Tech, 2000-present  
Big Brother, Big Sister Program, 1999-2000  
High School Varsity Volleyball Team, 1998-2000  
High School Symphonic Band, 1997-99

**HONORS** Maryland Distinguished Scholar, Honorable Mention, 2000  
Scholar Athlete Award, 1999-2000  
Merit Award, Honorable Mention, 1999 Media Festival, Photography Division

**AVAILABILITY** January 2003 preferred; also available August 2002

Fig. 8. The Resume Corresponding to Fig. 7.

Validation will be performed to ensure that a phone number and email address are what is actually in these

locations, but it is difficult to validate a name and address. Upon analyzing sample resumes for testing purposes

several resume were found to contain description line *Address* and prominent in student resumes was *Current Address* and *Permanent Address*, not wanting to include these lines in the personal information I exclude all personal information lines which contain the word *address*.

With all the data in the resume stored as Strings those Strings can be scanned for relevant keywords. Each category has its own set of keywords, which are stored in a text file. The keywords are stored here, once again, for easy customization and editing. All keywords found are store in Vectors. We are then left with a set of categories and corresponding attributes which we can assess the capabilities of the agent from.

#### E. An Example

A resume parser is built based on the above discussed methods. Fig. 7 shows the information extracted from the resume shown in Fig. 8.

## VI. CONCLUSIONS

Agent evaluation is important in role-based collaboration. This paper contributes in the following aspects: clarifies the problem and the requirement of agent evaluation; proposes an initial solution for this problem; analyzes the format of resumes; composes a method to extract information from resumes; and implements a tool to evaluate a group of agents for a group of roles and a tool to extract agent information from resumes. The proposed method is a good preparation for the next step of group role assignment [14]. The proposed tool can be applied into recruiting and team organizations to recommend appropriate staff.

Because this is the first attempt to evaluate agents' qualification to play roles (or people's qualification for a job), there are many further investigations to be done along the following directions:

- 1) Analyzing the skills in different domains and apply the tool into different domains to improve the precision of evaluations.
- 2) Analyzing the contents of job posting and resumes for different domains and make the tools more usable to practical applications.
- 3) Designing interactive tools for human resource managers to provide heuristic information to evaluate related agents (people).
- 4) Formalize the format of extraction to match the agent template in order create direct group qualifications by entering a list of resume files.

## ACKNOWLEDGMENTS

This research is in part supported by National Sciences and Engineering Research Council, Canada (NSERC: 262075-06), Ontario Partnership on Innovations of Commercialization, and IBM Eclipse Innovation Grant.

## REFERENCES

- [1] B. E. Ashforth, *Role Transitions in Organizational Life: An Identity-based Perspective*. Mahwah, NJ: Lawrence Erlbaum Associates, Inc., 2001.
- [2] J.S. Black, "Work Role Transitions: A Study of American Expatriate Managers in Japan", *Journal of International Business Studies*, vol. 19, no. 2, Summer, 1988, pp. 277-294.
- [3] R.P. Bostrom, "Role Conflict and Ambiguity: Critical Variables in the MIS User-Designer Relationship", *Proc. of the 17th Annual Computer Personnel Research Conference*, Miami, Florida, United States, 1980, pp. 88-115.
- [4] F. Ciravegna and A. Lavelli, "Learning Pinocchio: adaptive information extraction for real world applications", *Journal of Natural Language Engineering*, 10(2), 2004, pp. 145-165.
- [5] I. Kotlyar and K. Ades, "Don't Overlook Recruiting Tools", *HR Magazine*, vol. 42, no. 5, 2002, avail: [http://www.realestatesimulator.com/pdf/HR\\_May\\_02.pdf](http://www.realestatesimulator.com/pdf/HR_May_02.pdf).
- [6] J. Moore, R. Inder, P. Chung, A. Macintosh and J. Stader, "Who Does What? Matching Agents To Tasks In Adaptive Workflow", *International Conference on Enterprise Information Systems*, 2000, avail: <http://citeseer.ist.psu.edu/old/moore00who.html>.
- [7] A. Sitter and W. Daelemans, "Information Extraction via Double Classification", *Proc. of the International Workshop & Tutorial on Adaptive Text Extraction and Mining*, 2003, pp. 62-73, avail: <http://www.dcs.shef.ac.uk/~fabio/ATEM03/ATEM03-Proceedings.pdf#page=68>.
- [8] S. L. Thomas and K. Ray, "Recruiting and the web: High-tech hiring", *Business Horizons*, vol. 43, no. 3, May-June 2000, pp. 43-52.
- [9] M. Turoff, M. Chumer, B. Van de Walle and X. Yao, "The Design of a Dynamic Emergency Response Management Information System (DERMIS)", *J. of Information Technology Theory and Application*, 5(4), 2004, pp. 1-35.
- [10] B. Wu, X. Pei, J. B. Tan and Y. Wang, "Resume Mining of Communities in Social Network," *7<sup>th</sup> IEEE International Conference on Data Mining Workshop*, 2007, pp.435-440.
- [11] K. Yu, G. Guan and M. Zhou, "Resume Information Extraction with Cascaded Hybrid Model", *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages Ann Arbor, 2005, pp. 499-506, avail: <http://acl.ldc.upenn.edu/P/P05/P05-1062.pdf>.
- [12] H. Zhu, "Fundamental Issues in the Design of a Role Engine", *Proc. of the 6<sup>th</sup> International Symposium on Collaborative Technologies and Systems*, Irvine, CA, USA, May 19-23, 2008, pp. 399-407.
- [13] H. Zhu, M. Grenier, R. Alkins, J. Larcarte and J. Hoskins, "A Visualized Tool for Role Transfer", *Proc. of the IEEE International Conference on Systems, Man and Cybernetics (ICSMC)*, Singapore, Oct. 12-15, 2008, pp. 2225-2230.
- [14] H. Zhu and A. Rob, "Group Role Assignment", *Submitted to the 7<sup>th</sup> International Symposium on Collaborative Technologies and Systems*, Baltimore, Maryland, USA, May 19-23, 2009.
- [15] H. Zhu and M.C. Zhou, "Role-Based Collaboration and its Kernel Mechanisms", *IEEE Trans. on Systems, Man and Cybernetics, Part C*, vol. 36, no. 4, July 2006, pp. 578-589.
- [16] H. Zhu and M.C. Zhou, "Role Transfer Problems and Algorithms", *IEEE Trans. on Systems, Man and Cybernetics, Part A*, vol. 36, no. 6, Nov. 2008, pp. 1442-1450.