

ERGASIOGNOMON - a Model System of Advanced Digital Services Designed and Developed to Support the Job Marketplace

Giorgos M. Milis¹, Andreas L. Symeonidis^{1,2} and Pericles A. Mitkas^{1,2}

¹Department of Electrical and Computer Engineering
Aristotle University, 54124 Thessaloniki, Greece

²Laboratory of Intelligent System and Software Engineering
Informatics and Telematics Institute - CERTH, 57001Thessaloniki, Greece

Abstract. The continuous expansion of Internet has enabled the development of a wide range of advanced digital services. Real-time data diffusion has eliminated processing bottlenecks and has led to fast, easy and no-cost communications. This primitive has been widely exploited in the process of job searching. Numerous systems have been developed offering job candidates with the opportunity to browse for vacancies, submit resumes, and even contact the most appealing of the employers. Although effective, most of these systems are characterized by their simplicity, acting more like an enhanced bulletin board, rather than an integrated, fully functional system. Even for the more advanced of these systems user interaction is obligatory, in order to couple job seekers with job providers, thus continuous supervising of the process is unavoidable. Advancing on the way primitive job recruitment techniques apply on Internet-based systems, and dealing with their lack of efficiency and interactivity, we have developed a robust software system that employs intelligent techniques for coupling candidates and jobs, according to the formers' skills and the latter's requirements. A thorough analysis of the system specifications has been conducted, and all issues concerning information retrieval and data filtering, coupling intelligence, storage, security, user interaction and ease-of-use have been integrated into one web-based job portal.

1 Introduction

The Information Society initiative has led to the exploitation of information diffusion in a wide range of application domains. Sufficient efforts have been made in the development of electronic systems that aim to assist in the job seeking processes that promise a fairly quicker and much more effective contact between those who are asking for a job and those who are offering one. Such an automated system can provide numerous advantages, i.e. the reduction of the operating cost due to minimization of human interaction, the reduction of time and man-effort requirements, since the end-user can do everything from remote locations interacting with the system through World Wide Web connections, and the increment of the supplied services reliability.

Nevertheless, it is very difficult for such a system to embed intelligence so as to check, combine and decide on jobs and candidates. Most of the software systems that

have been developed stop short at automating only simple procedures. A detailed survey on the most powerful tools that provide job-seeking support ([10-28]) has led to the conclusion that all these network places offer well-designed user interfaces and electronic maintenance for submitting resumes, as well as job announcements using suitable electronic forms for importing required data. Data are stored in databases for the end users to carry out simple searches; the procedure involves the insertion of certain keywords, according to the interests of the user. The result of the query is usually a rather long list meeting user requirements ([10-21]). The most advanced of the systems carry out regular searches and inform users for recent results via electronic mail ([22-25]).

None of the already developed systems, though, efficiently faces the problem of dynamically coupling candidates and jobs. Our system, Ergasiognomon (Figure 1), is an integrated tool minding to improve the job seeking and offering procedures in the Greek marketplace through WWW (World Wide Web) services. Through an “intelligent” procedure of matching desirable and available qualifications, Ergasiognomon suggests suitable candidates for employment to the users that offer jobs, while at the same time it informs the users that are asking for a job that their resume has been forwarded to some employer. The interaction between the end users and Ergasiognomon takes place through a suitable user interface, which has the ability to adapt to the end user’s category.

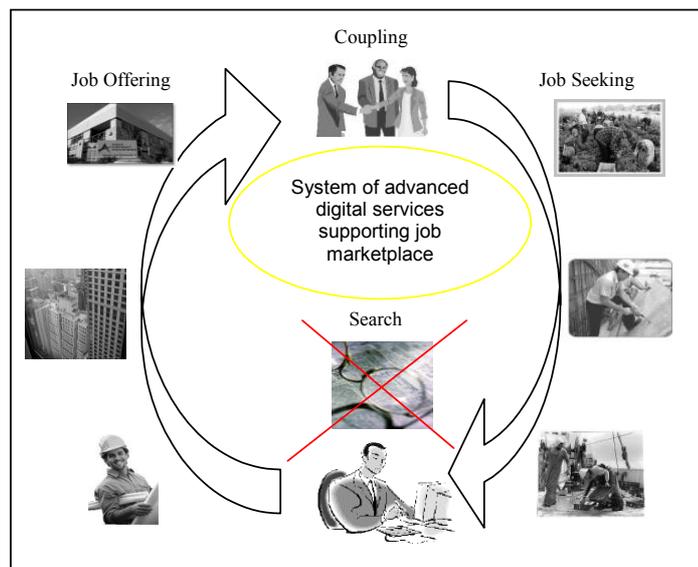


Fig. 1. The Environment of Ergasiognomon

The rest of this paper is organized as follows. Section 2 presents the specifications and the information flow throughout Ergasiognomon, whereas section 3 describes in detail the intelligence of Ergasiognomon, including the information retrieval and “coupling” procedures. Finally, section 4 summarizes the work presented and concludes this paper.

2 System Specification

2.1 System Use Case Description

Since Ergasiognomon is aiming to address a wide range of users that might not be sufficiently familiar with computer technology, an attempt has been made so that the final product would be as end-user friendly as possible. The sole task of the end user is to provide the system with several personal data as well as data concerning the job that he/she seeks or offers. For data collection Ergasiognomon provides two possibilities:

1. Filling in electronic forms, and
2. Uploading (and subsequently, parsing) a preprocessed text file (.doc, .txt, .csv)

It should be denoted that from the point that the user provides his/her personal data Ergasiognomon becomes self-ruling.

The core of the system is the database, serving the two basic categories of the end users:

- a) The Job Seeking users (JS), and
- b) The Job Offering (JO) users.

The first category involves people who want to find a convenient job. Ergasiognomon stores their personal profile, including skills, special abilities, requirements, personal data and any data considered essential. The data are being stored in such a way so that their farther processing becomes easy and feasible. The second category involves people, organizations or enterprises that offer jobs. In a rather similar way, Ergasiognomon generates the employer's profile (requirements, desirable qualifications, several selection criteria) and inserts it to its database.

Our system offers an intelligent procedure that implements the contact between job seekers and employers, without the need of human intervention. In particular, the procedures of submitting and reading resumes, submitting and reading job announcements, coupling employment candidates and jobs as well as informing anyone concerned, becomes automated and scheduled.

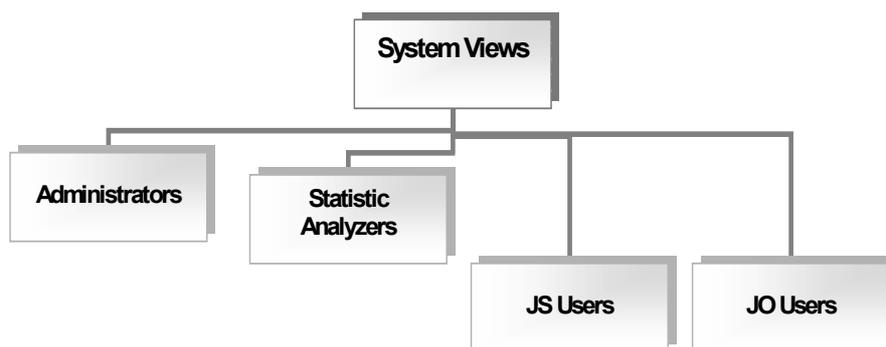


Fig. 2. User Categories

2.2 Flow of Information and System’s Views Hierarchy

Ergasiognomon has been designed to serve four (4) primary user categories (Fig. 2):

- the System Administrator
- the Statistic Analyzer
- the JS user
- the JO user

System Administrator. The user that is responsible for controlling and securing Ergasiognomon’s proper operation. Available services:

1. Access to database’s basic tables with the authority to insert, delete or update their content.
2. Ability to choose from and execute specific functions through the front-end (Table 1).
3. Delete/Create system roles.

Figure 3 illustrates all the available to the Administrator options.

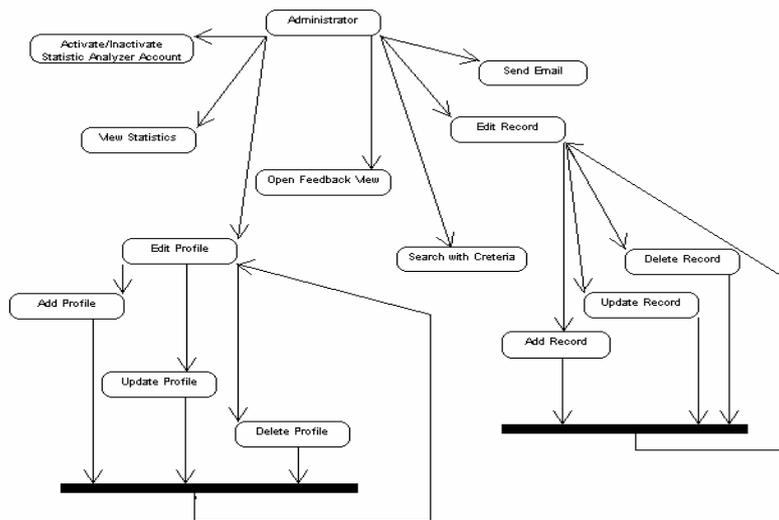


Fig. 3. The flow diagram of the Administrator

Specific functions in Administrator’s View	
i.	Insert a user of any category
ii.	Delete/Update user profile
iii.	Appearance of Statistic results
iv.	Delete/Create Statistics
v.	Supervision of coupling results
vi.	Contact users via email

Table 1. Available Functions for the Administrator

Statistics Analyzer. The Statistic Analyzer privilege is assigned to a user by the administrator and allows him/her to:

1. Study and review statistics of the system by executing the suitable scripts.
2. Update personal profile.
3. Access database's views containing statistical information.
4. Update the contents of these views.
5. Contact the Administrator.

Figure 4 illustrates the available options for the Statistic Analyzer.

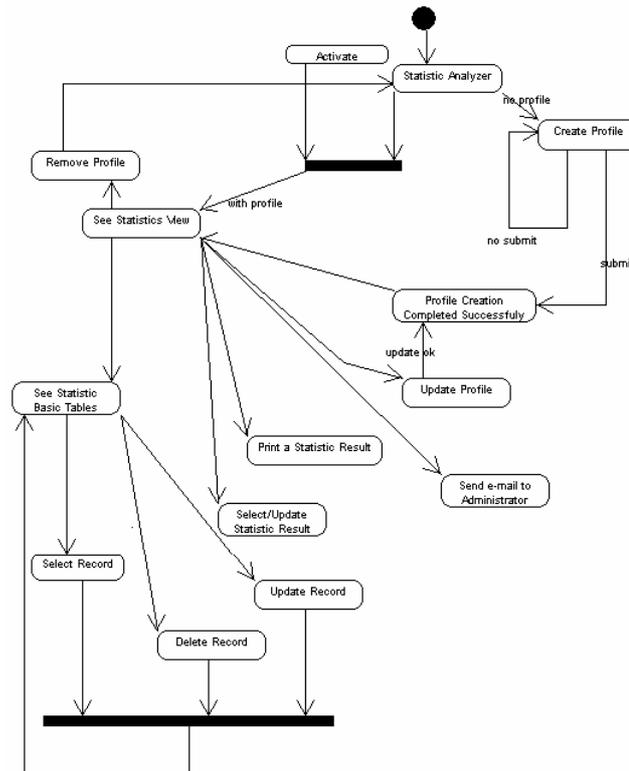


Fig. 4. The flow diagram for the Statistic Analyzer

Job Seeking User. The user has access to the system in order to:

1. Create a personal profile.
2. Submit a resume as a file of word document format.
3. Update his/her profile.
4. Access related statistic results.
5. Send feedback with suggestions.
6. Deactivate the profile.
7. Create a profile as a Job Offering user, too.

Figure 5 illustrates the options offered to the Job Seeking user.

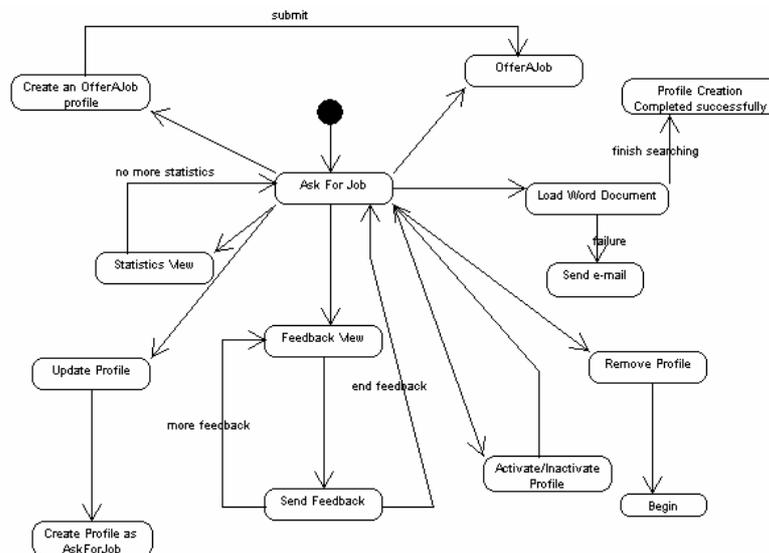


Fig. 5. The flow diagram for the JS user

Employer View. Finally, the JO user has all the options that JS user, as well as some additional ones. These are:

1. Access to a personal web page where each JO user can see the results produced from the data coupling. A list of JS users found to meet the job requirements is created; this list is sorted according to the level of suitability of the candidates to the job. Ergasiognomon takes into account a priority order of special criteria that the employer provides, and automatically generates levels of suitability for every place being offered.
2. Ability to remove (or not) a candidate profile from the list.
3. Ability to contact candidates through a mailing list created especially for this reason.

Figure 6 illustrates all of the options that JO user has.

2.3 Features of Ergasiognomon

Ergasiognomon is a distributed system, which involves several technologies and whose operations are executed at disperse locations. It is essential for such a system to retain synchronization between its modules and compact integration, while keeping their independency for facilitating easy maintenance and reusability. The Ergasiognomon architectural decisions ensure security, versatility, reusability and robustness.

Security. In order for such a web-based system to be considered as fully-functional, it is essential to succeed in facing mischievous attacks. Even normal use of the system may also cause its transition to an instable state. That's why recovery modes were implemented so as to revert the system to a safe state.

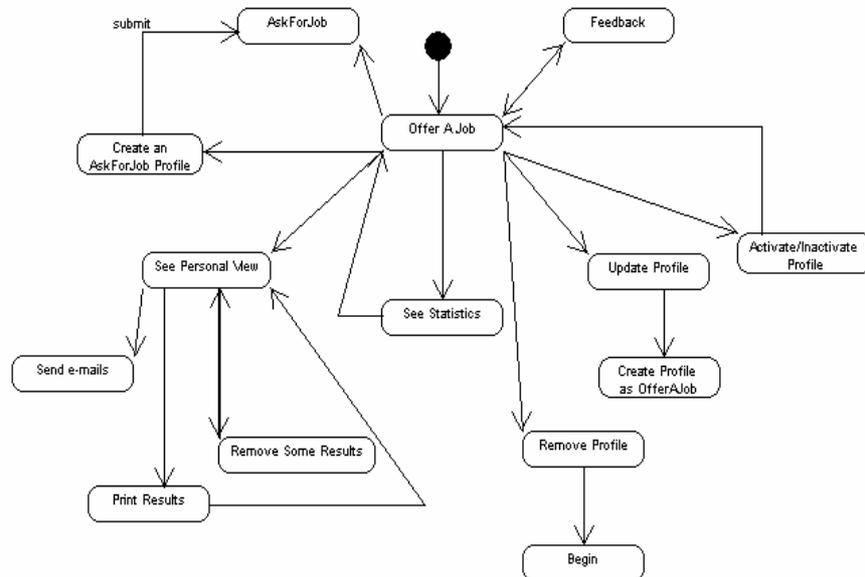


Fig. 6. The flow diagram for the JO user

In addition, while registering, every user has to follow a certain process that augments security. The user has to select a username and a password as well as provide the system with a valid email address. As soon as the registration process finishes, the user receives an electronic mail containing an activation password, which is requested the first time user attempts to log into Ergasiognomon.

Finally, every crucial information that flows throughout Ergasiognomon's environment is very well encrypted using suitable vbscript, javascript and sql functions.

The system also implements some extra modern security techniques in order to prevent the mischievous attacks. First of all, an internet firewall has been built, using IDS (Intrusion Detection System) functionality, in order to sniff out network packets and gain a good understanding of what is really happening on our LAN. Secondly, the web server, the application server and the database server are located in different internet zones. Every single request is being directed to the web server. This server has no crucial information. The requests are filtered there and only the save ones are being directed to the application server through a different session. Last but not least, the database server belongs to a separate V-LAN. By this way the system allows the network administrator to segment users requiring access to sensitive information into a different V-LAN than the rest of the general user community regardless of physical location.

Versatility. Ergasiognomon can "recognize" its users and taking into account the category that each user belongs to, adapt its interface and provide access to suitable services. If the user browser accepts cookies, Ergasiognomon can personalize information according to his/her interests.

Moreover, every task is being implemented as a transaction that has a start point and at least one roll back point. No process leaves unfinished work and each task returns a report on its state. Assuming that during the submission of a user's profile something goes wrong and the database server crashes, our system has the ability to recover from such a situation. After having received a failure report, Ergasiognomon saves the user profile in a text file in the application server. The user is being informed that his profile shall be imported. This is done by the exploitation of *DTS (Data Transformation Services)* ([34-37]) in order to manage Ergasiognomon. A DTS package is scheduled to be executed every morning which takes the profiles stored in text files and imports them into the system database. In case a success report is received, users are being notified for the insertion of their profile, through electronic mail. Otherwise, they are notified accordingly.

Robustness and Reusability. In order for Ergasiognomon to retrieve, manipulate and restore data and implement all the separate tasks, we have adopted the solution of Stored Procedures ([3, 4, 6, 33]). Each one of them takes a number of inputs, carries out specific operations behaving as a self-governed entity and provides desirable outputs in the form of SQL variables or SQL record-sets (arrays with records of special traits). By this mean, we managed to build an interface between the inner architecture of Ergasiognomon and its web pages. This interface also acts as a security "firewall" between the end-user and the data stored in the database.

3 The Intelligence of Ergasiognomon

This chapter describes the most innovative characteristics of Ergasiognomon. The automated CV parsing, where Ergasiognomon retrieves the needed information for creating Job Seeking profiles, and the execution and administration of the coupling process which takes place in the backstage are presented in detail.

3.1 Information Retrieval

According to Ergasiognomon's specifications, the final product is designed to successfully support the end users even if they might not be sufficiently familiar with computer technology. We have implemented a method that retrieves all the information needed to compose the Job Seeking user's profile by parsing the user's resume in whichever common text format. That is, JS can bypassing the time-consuming process of filling out all the appropriate electronic forms just by simply uploading his/her resume, thus saving time and trouble.

The whole procedure is divided into four steps. First of all, Ergasiognomon gets the document submitted by the user and stores it. Consequently, it "filters" the document through an ActiveX function and transforms it into a plain text file in order to initiate the processing procedure. The text file is stored into a database table row by row. Finally, the last step, which is the most significant one, is to execute a stored procedure that parses the table's rows, retrieves the information, declares local vari-

ables to temporary save this information and finally inserts the whole profile into the database.

Figure 7 illustrates the graphic representation of the whole package. Initially, the DTS package that gets the document, transforms it into a text file and imports it into the “TextFile” table is executed.

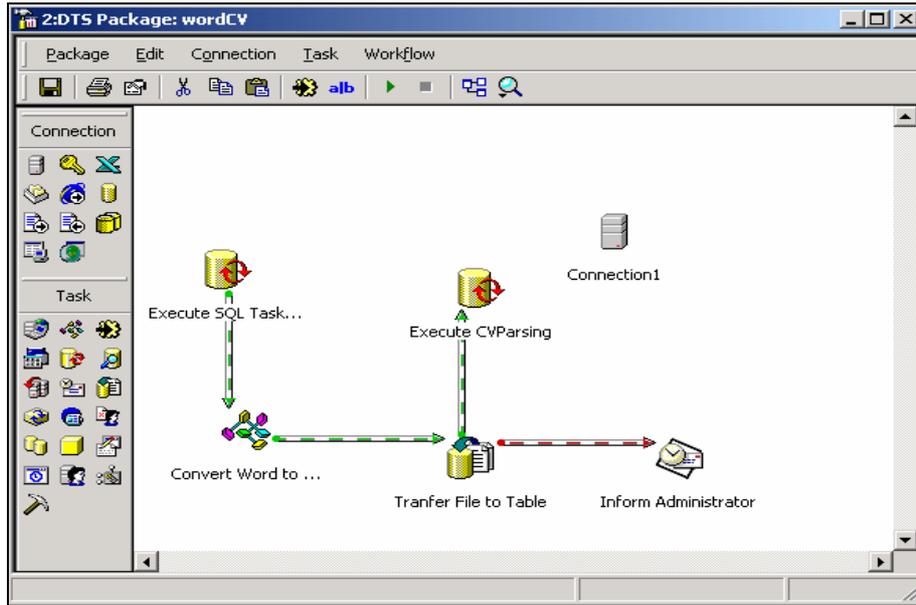


Fig. 7. The DTS package for retrieving resume information

A database connection is created and an SQL task deletes former data from “TextFile” table. Successful execution of this step triggers the execution of “Bulk Insert” command, which transfers the file into a database’s table. If anything goes wrong, an administrator is informed via email in order to take action immediately. Otherwise, the store procedure that retrieves the information is executed; the execution of this last procedure involves the cooperation with a number of other procedures.

Three tables have been created to support this procedure (“ontologism”, “ontologism2”, and “ontologism3”). Every single column of these tables represents ontology. Figure 8 and 9 illustrate some fields and a snapshot of these tables. For example, the column that represents the name ontology includes records of phrases, clauses, words etc. that are considered essential for detecting the point where the candidate’s name lies in the text file. All the other ontologies have been created in a similar way. Since the option values for every single field in the tables are predefined (in order to make “matching” feasible) the use of a great number of ontologies was unavoidable. The requisite information is divided into subcategories: personal data, previous employment, job categories, search location, education and candidate hobbies. Cursors of suitable types are declared in order for the procedure to parse the table in all directions through the record-set. After having detecting the range where the inquired in-

formation lies, the procedure stores the range bounds into proper variables. The next step is to call the corresponding procedures that parse segregate rows and, using the ontologies as well as the functions provided by SQL Server for string manipulation, retrieve the required information.

As soon as the procedure finishes, all the variables composing the profile have an acceptable value. These values are inserted into the database. From that point on, and if needed, the user can update the profile whenever he/she decides it is essential.

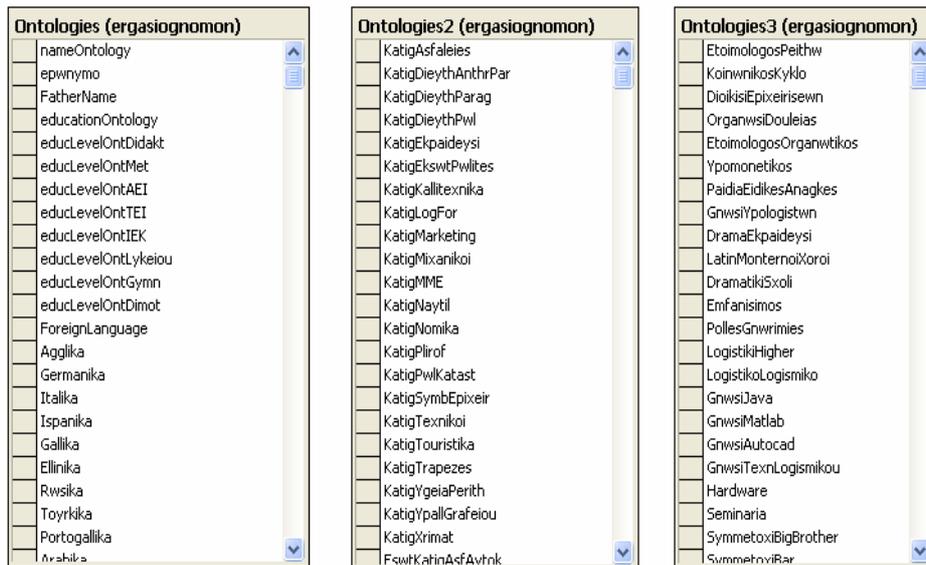


Fig. 8. Some fields of the three tables including the Ontologies

nameOntology	epwnymo	FatherName	educationOntology	educLevelOntDidakt	educLevelOntMet	educLevelOntAEI	educLevelOntTEI
Όνομα	Επώνυμο	Πατρώνυμο	Εκπαίδευση	Διδακτορικό	Μεταπτυχιακό	AEI	TEI
Όνοματεπώνυμο	ΕΠΩΝΥΜΟ	Όνομα Πατρός	Πτυχιούχος	Διδακτορικός	Μεταπτυχιακός	A.E.I	T.E.I
Βιογραφικό του	<NULL>	<NULL>	Μόρφωση	Διδάκτωρ	Master	Πανεπιστήμ	Τεχνολογικό Εκπαιδευτικό Ίδρυμα
Είμαι ο	<NULL>	<NULL>	Πτυχιό	Διδακτορική άτυπη	MSc	Πανεπιστήμ	Σπουδαστής
Όνομάζομαι	<NULL>	<NULL>	Θεωρητική Κατάρτιση	Doctor	MA	Ανώτατ	Ανώτερο Τεχνολογικό Ίδρυμα
Το όνομα μου είναι	<NULL>	<NULL>	ΕΚΠΑΙΔΕΥΣΗ	Dr	<NULL>	Πτυχιό	<NULL>
<NULL>	<NULL>	<NULL>	<NULL>	Διδακτορας	<NULL>	Πτυχιούχ	<NULL>
<NULL>	<NULL>	<NULL>	<NULL>	<NULL>	<NULL>	<NULL>	<NULL>
*							

Fig. 9. A snapshot of the first table

3.2 Executing and Administrating the “Matching” Procedure.

Ergasiognomon aims to entirely automate job seeking and offering procedures so that the client can simply provide his data and allow the system to take over for all the rest tasks. That also implies efficient data filtering and successful result matching.

More specifically, the “matching” procedure of Ergasiognomon inspects every available job offer, goes through a detailed search on the database basic tables, and

retrieves exactly those Job Seeking users that meet the minimum of each offer requirements. It should be denoted that a job might suit more than one candidates and every candidate might also be suitable for more than one job. Each job (and not each employer) is considered as an independent entity. The last step, informing both sides about the results, is also automated. The whole procedure is scheduled periodically (every Sunday on 12:00 for example), without any human intervention. The administrator is informed to take action immediately, only when needed. Every alert activates a procedure of sending an email to administrators.

Using *DTS* tool, we have created a custom “package for operation” which has the ability to execute different tasks, sequentially or parallel. Figure 10 illustrates the design sheet of the whole matching process taking place in Ergasiognomon.

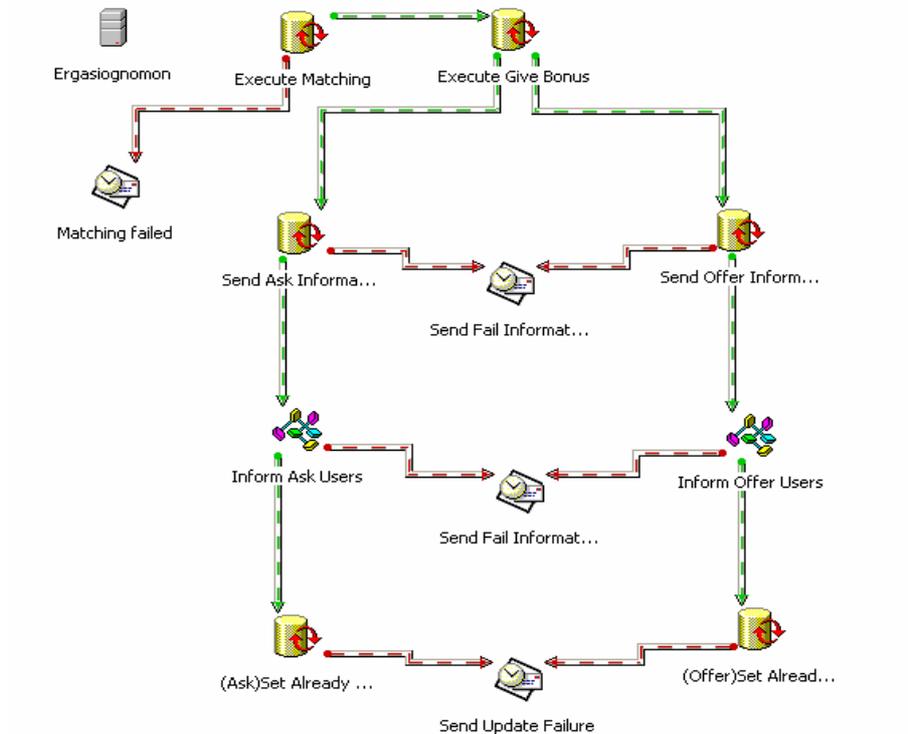


Fig. 10. The DTS package for “Matching” information

The “Matching” procedure takes into account only those users that have active profiles. The appropriate table variables are declared in order to store the passwords of the Job Seeking users found to meet a specific requirement. This process is repeated for each job offered. The system calculates the intersection of all generated tables, thus ending with one single table containing the relation of Job Seeker’s passwords and job identities. The next step is to retrieve the contact information for those users found in the “matching” table. Job Seeking users automatically receive an email, which informs them that their profile has been found to meet the requirements for some of the jobs offered and their sole task is to wait for to be contacted. On the other

hand, employers receive an email, which notifies them to visit their personal page on Ergasiognomon's web site and view extra details about the proposed candidates. Successful execution of the information sending task triggers another task, which changes the logical value of a field in the "matching" table, thus, declaring who has been informed and who hasn't yet. Table 2 illustrates a snapshot of the "matching" table.

positionID	password	activate	date	already_send	bonus
22	user3	1	9/9/02 5:35:04 PM	0	31
22	abcdef	1	9/9/02 5:35:04 PM	0	56
22	stuvwxyz	1	9/9/02 5:35:04 PM	0	56
23	ghiklm	1	9/9/02 5:35:04 PM	0	94
23	nopqrst	1	9/9/02 5:35:04 PM	0	94
24	αβγδεζ	1	9/9/02 5:35:04 PM	0	95
24	ηθκλμ	1	9/9/02 5:35:04 PM	0	95
27	νξοπστ	1	9/9/02 5:35:04 PM	0	105
27	ωπρστω	1	9/9/02 5:35:04 PM	0	105

Table 2. An instance of table "Matching"

Special attention should be given to the "bonus" field. It is the metric that determines the suitability of a candidate according to the requirements of the job and his/her skills. Sorting takes into account the bonus that every candidate gains for a specific job and lists them in descending order. The bonus metric is derived from the successful execution of the "Matching" procedure, which, in turn, triggers the execution of another procedure called "Give_Bonus". This procedure parses the database once more and gives bonus to Job Seekers according to their extra qualifications (that exceed minimum requirements) considered valuable for each job. Each employer has the opportunity of giving certain priorities to the criteria that taken into account to the "Give_Bonus" procedure. The priority order generates proper weights for calculating the total bonus value for each candidate. All these tasks are being integrated into a one scheduled program for periodic execution.

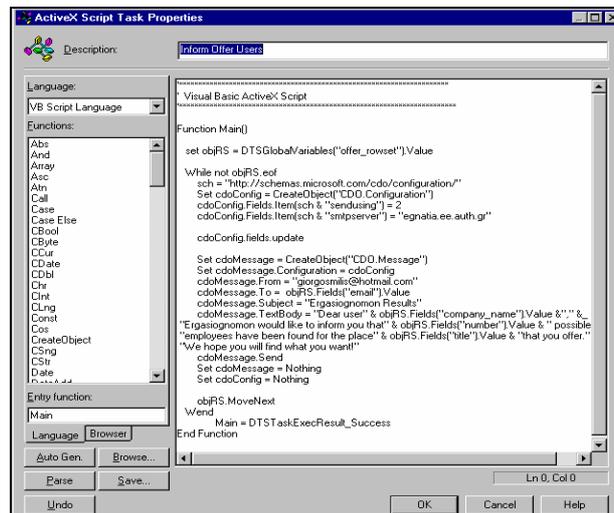


Fig. 11. Automated email sending function

It is by now obvious that the Data Transformation Services have provided us with the technology to integrate different technologies into one single application. Figure 11 illustrates the VBScript program that has been implemented for sending automatic emails to employers. It is important to point out that information can flow from one task to another inside the package in the form of global variables or record-sets.

4 Summary and Further Work

This paper has presented Ergasiognomon, a system that has been implemented to support the job market. More specifically, Ergasiognomon automates the processes of job searching and offering and aims to eliminate human intervention. Innovation lies in the provision of an intelligent machine for coupling candidates and jobs as well as the mechanism for extracting Job Seeking profiles through resumes in common text format, therefore saving time and effort.

Further work in the context of Ergasiognomon includes the development of a Multi-Agent System for taking advantage all the benefit of agent technology. The whole idea introduces mobile agents to Ergasiognomon, assigned to retrieving essential information. These agents will undertake the collection of the data for creating job-offering profiles, even from remote Internet locations. More specifically, these agents will be assigned to visit several sites of enterprises, organizations, and companies, to discover and to find announcements about search of cooperators. They will be responsible for performing database's updates after any retrieval of new data.

Moreover, Data Mining techniques could also be useful in the field of job searching. The storing of job market trends and the recognition of recursive behavior patterns in job marketplace data could lead to a joint powerful decision support system that can prove to be a significant aid for Professional Organizations in the public sector, which base on statistic analyses and economic appreciations for tracing employment policies.

The subsystems of the final product will be integrated into one, robust and promising software system.

References

1. P.A. Mitkas, Notes on Software Engineering (in Greek).
2. P.A. Mitkas, Notes on Relational Database Systems (in Greek).
3. C.J. Date, *An Introduction to Database System Design*, Volumes I & II- (in Greek)
4. R. Riordan, *Programming on Microsoft SQL Server 2000 Step by Step*.
5. E.A. Giakoumaki, *Software Engineering*, Volumes I & II- (1994) (in Greek)
6. B. Forta, *SAMS Teach Yourself SQL Quick steps for fast results*
7. M. Israel, J.S. Jones, *SQL Server 2000 Design* (Study Guide)
8. D. Flanagan, *JavaScript (The Definitive Guide)*.
9. D.D. Gutierrez, *Web Database Development for Windows Platforms*
10. C. Seidman, *Data Mining with Microsoft SQL Server 2000*
11. Work Force Employment Organization. The site of OAED. Available at: <http://www.oaed.gr/>

12. Just Jobs. Available at: <http://www.justjobs.gr>
13. Sky Walker. Available at: <http://www.skywalker.gr>
14. E-diana. Available at: <http://www.e-diana.gr>
15. Telejob. Available at: <http://www.telejob.gr>
16. Man Work. Available at: <http://www.manwork.gr>
17. Available at: <http://www.yyp.gr>
18. Adecco. Available at: <http://www.adecco.gr>
19. EURES. Available at: <http://europa.int/jobs/eures>
20. Careers on Web. Available at: <http://www.careersonweb.com>
21. Careers from Home. Available at: <http://www.CareersFromHome.com>
22. Career Path. Available at: <http://www.careerpath.com>
23. Job Search. Available at: <http://www.fedworld.gov/jobs/jobsearch>
24. Hot Jobs. Available at: <http://www.hotjobs.com>
25. Monster. Available at: <http://www.jobsearch.monster.com>
26. Job Web. Available at: <http://www.jobweb.com/>
27. Four Work. Available at: <http://www.4work.com/>
28. CareerSite. Available at: <http://www.careersite.com/>
29. Job Seeking. Available at: <http://www.jobseeking.net>
30. E. Skordalakis, B. Beskoukis, Object-Oriented Methodologies for Software (UML) Implementation. Available at: http://www.softlab.ntua.gr/~bxb/ppt/vv-uml-01_files/frame.htm
31. The JavaScript Source: Password Protection: Password Generator. Available at: <http://javascript.internet.com/passwords/password-generator.html>
32. Encrypt Query Strings. Available at: <http://a1lab.com/a1asp/sqs.phtml>
33. VBScript Interfaces in SQL Server 2000. Available at: <http://msdn.microsoft.com/msdnmag/issues/02/08/VBScriptandSQLServer2000/>
34. Programming a Microsoft SQL Server 2000 Database. Available at: <http://www.microsoft.com/TRAINCERT/SYLLABI/2073AFINAL.ASP>
35. Microsoft SQL Server 2000 DTS [Data Transformation Services] by Timothy Peterson. Available at: http://www.informit.com/isapi/product_id~%7BC9C8EB0B-3579-4C69-8028-2CE0285F82FB%7D/content/index.asp
36. Professional SQL Server 2000 DTS by Mark Chaffin, Brian Knight, Todd Robinson. Available at: http://www.sql-server-performance.com/sql_server_2000_dts.asp
37. Global Variables and SQL Statements in DTS By Darren Green. Available at: <http://www.sqldts.com/default.aspx?6,102,205,7,1>
38. DTS Programming. Adding DTS ActiveX Scripts. Available at: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dblibc/dtspapps_3y7n.asp