

# A P2P GRID ARCHITECTURE FOR DISTRIBUTED ARABIC OCR BASED ON THE DTW ALGORITHM

M. Khemakhem\* and A. Belghith\*\*

## Abstract

Arabic cursive optical character recognition (OCR) based on the dynamic time warping (DTW) algorithm provides simultaneously very interesting segmentation and recognition rates. However, the computing complexity of the DTW algorithm restricts its widespread utilization and its consideration at a commercial scale. Accelerating the DTW execution time has attracted many researchers and several solutions have already been proposed. These solutions are commonly based on very specialized processors and hardware architectures and as such they remain very expensive and not amenable to a large scale utilization.

In a previous work, we found that loosely coupled architectures can indeed provide viable infrastructures to implement a distributed Arabic OCR. Our objective here is to allow the recognition of huge quantities of Arabic documents such as those of certain national libraries. Undoubtedly, enough processing power and storage capabilities are needed. In this paper, we proposed and used a peer-to-peer (P2P) architecture using the scientific research Tunisian grid (SRTG). Conducted experiments testify that our proposed architecture provides very adequate speedups of the DTW-based Arabic OCR.

## Key Words

Arabic cursive OCR, the DTW algorithm, P2P grid computing, SRTG, experimental study

## 1. Introduction

Arabic optical character recognition (OCR) based on the dynamic time warp (DTW) algorithm provides a very interesting recognition rate. Conducted experiments achieved on high and medium quality documents composed of more than 30,000 Arabic words and pseudo words (blocks of connected characters not necessarily complete words) showed that a recognition average rate of more than 98% and a simultaneous segmentation average rate of more than 99%

are indeed reachable [1–3]. Moreover, we found, in particular, that the recognition rate (respectively the segmentation rate) increases as the font size gets larger. Besides, the DTW procedure presents two main advantages. The first advantage is its ability to properly recognize blocks of connected characters (cursive writing) without a prior segmentation. The second advantage is its total independency with respect to the vocabulary to be recognized as the reference library used is composed of only isolated characters. Unfortunately, the DTW procedure necessitates a huge amount of processing and as such will restrict its widespread use especially for large quantities of documents.

Several researchers attempted to speedup the execution time of the DTW algorithm. Nevertheless, the proposed solutions did not provide the presumed success because they commonly required very expensive specialized processors or hardware architectures [4–10]. In our previous works [2, 3] we have shown that loosely coupled architectures such as existing local area networks (LANs) and grid computing can provide very interesting, yet costless, infrastructures to speed up the execution time of the DTW algorithm through its data distribution; namely the distribution of the input binary image of the Arabic text to be recognized. The recognition speed is of utmost importance whenever we consider huge amounts of Arabic text such as an entire library of documents. Grid computing provides enough computing and storage capacities [11, 12], yet it presents various possibilities to perform the distribution of the text to be recognized.

In this paper, we pursue our investigations and show through a set of experimental studies that grid computing, and more specifically the scientific research Tunisian grid (SRTG) which is a peer-to-peer (P2P) architecture, provides a very interesting infrastructure to speed up the DTW execution time especially when huge amounts of text need to be recognized.

Our paper is organized as follows: Section 2 describes the Arabic OCR based on the DTW algorithm. Section 3 gives an overview on grid computing and the SRTG platform. Section 4 details the deployment of the Arabic OCR within the SRTG and the results of the conducted experiments. Concluding remarks and future investigation are presented in Section 5.

\* MIRACL Lab, FSEGS, University of Sfax, BP 1088, 3018, Sfax, Tunisia; e-mail: maher.khemakhem@fsegs.rnu.tn

\*\* HANA Research Group, ENSI, University of Manouba, 2010, Manouba, Tunisia; e-mail: abedelfattah.belghith@ensi.rnu.tn

## 2. Arabic OCR based on the DTW Procedure

The DTW is a well-known procedure especially in pattern recognition [4, 6, 8, 10, 13–16]. The purpose of this procedure is to perform an optimal time alignment between a reference pattern and an unknown pattern and evaluate their difference. What makes the DTW procedure very attractive is its ability to recognize properly cursive characters (connected blocks of characters such as words or parts of words in Arabic) without the need to a prior segmentation into characters according to a given reference library of isolated characters. The adaptation of this procedure to the Arabic cursive OCR has shown to provide very interesting results [1–3].

### 2.1 The DTW and Cursive Characters Recognition

Words especially in Arabic, as is the case of many other languages, are inherently written in blocks of connected characters. Although the segmentation of the text into blocks of connected characters is a preliminary phase to the recognition process, a further segmentation of these blocks into separate characters is usually adopted. Indeed, many researchers have considered the segmentation of Arabic words into isolated characters before performing the recognition phase [17–19]. The crux of the viability of the use of DTW technique is then its ability and efficiency to perform the recognition without the prior segmentation of blocks into separate characters.

Let  $V$  represents a reference library of  $R$  trained characters  $C_r$ ,  $r = 1, 2, \dots, R$ , defining the Arabic alphabet in some given fonts. We here stress the fact that several fonts could be considered even simultaneously. It suffices to get them trained which is easily done at the learning phase while constructing the reference library  $V$ . Let  $T$  represents a block of connected Arabic characters to be recognized.  $T$  is then composed of a sequence of  $N$  feature vectors  $T_i$  that are actually representing the concatenation of some subsequences of feature vectors, representing each an unknown character to be recognized. The text  $T$  is seen as lying on the time axis (the  $X$ -axis) in such a manner that feature vector  $T_i$  stands at time  $i$  on this axis. The reference library  $V$  is portrayed on the  $Y$ -axis, where the reference character  $C_r$  is of length  $l_r$ ,  $1 \leq r \leq R$ . According to [1–3]. Let  $S(i, j, r)$  represents the cumulative distance at point  $(i, j)$  relative to the reference character  $C_r$ . The objective is then to detect simultaneously and dynamically the number of characters composing  $T$  and recognizing these characters. There exists surely a number  $k$  and indices  $(m_1, m_2, \dots, m_k)$  such that  $Cm_1 \oplus Cm_2 \oplus \dots \oplus Cm_k$  represents the optimal alignment to text  $T$  where  $\oplus$  denotes the concatenation operation. The path warping from point  $(1, 1, m_1)$  to point  $(N, l_{m_k}, k)$  and representing the optimal alignment is therefore of minimum cumulative distance, i.e.,

$$S(N, l_{m_k}, k) = \min_{1 \leq r \leq R} \{S(N, l_r, r)\} \quad (1)$$

This path, however, is not continuous because it spans various characters. We therefore must allow at any time

the transition from the end of one reference character to the beginning of a new character. The end of reference character  $C_r$  is first reached whenever the warping function reaches the point  $(i, l_r, r)$  where  $i = \lceil \frac{l_r+1}{2} \rceil, \dots, N$ . The warping function always reaches the ends of the reference characters. At each time  $i$ , we allow the start of the warping function at the beginning of each reference character along with the addition of the smallest cumulative distance among the end points found at time  $(i-1)$ . The resulting functional equations are

$$S(i, j, r) = D(i, j, r) + \min_{\substack{1 \leq i \leq N \\ 1 \leq j \leq l_r \\ 1 \leq r \leq R}} \left\{ \begin{array}{l} S(i-1, j, r), \\ S(i-1, j-1, r), \\ S(i-1, j-2, r) \end{array} \right\} \quad (2)$$

with the boundary conditions

$$S(i, 1, r) = D(i, 1, r) + \min_{\substack{1 + \lceil \frac{l_r+1}{2} \rceil \leq i \leq N \\ 1 \leq j \leq l_r \\ 1 \leq r \leq R}} S(i-1, l_k, k) \quad (3)$$

To trace back the warping function and the optimal alignment path, we have to memorize the transition time from one reference character to the others [1–3]. This can easily be accomplished by the following procedure:

$$b(i, j, r) = \text{trace} \min_{\substack{1 \leq i \leq N \\ 1 \leq j \leq l_r \\ 1 \leq r \leq R}} \left\{ \begin{array}{l} b(i-1, j, r), \\ b(i-1, j-1, r), \\ b(i-1, j-2, r) \end{array} \right\} \quad (4)$$

Where *trace* min is a function that returns the element corresponding to the term that minimizes the functional equations.

## 3. Grid Computing

When we talk about grid, we can mention mainly computer grids, data grids, dedicated grids [20] and volunteer grids [21]. A dedicated grid is composed of a fixed but a big number of connected computers which can be used together for specific purposes after a prior reservation. Users of this category of grids must prepare in advance their distributed jobs (applications) in the form of a graph of tasks [20]. It means that any distributed application must be managed by its own user. However, a volunteer (benevolent) grid is composed of a variable number of computers connected benevolently to a specific middleware. Users must be subscribed and authorized in advance to access and use such grids. A volunteer grid has commonly a P2P architecture which allows voluntarily to its own users to connect or disconnect their participating computers. Abilities of these infrastructures to provide enough computing and storage capacities [12, 20–25] to allow multiple resources sharing constitute their most attractive side. We recall that our main objective here is to make possible the recognition of large quantities of Arabic documents by using the DTW procedure which requires certainly such capacities.

### 3.1 The Scientific Research Tunisian Grid

This grid is a P2P architecture [26–28] composed of a middleware which is similar to XtremWeb-CH [27] and XtremWeb [28] and a set of connected benevolent computers. These connected computers belong to different educational institutions and several Tunisian universities. To run a given distributed application on the SRTG platform, an authorized user gets first logged in to the corresponding middleware. Then, before the deployment of his/her distributed application, he/she must prepare (manually or automatically) an XML file [29, 30] which describes the way his application will be executed. It means that he/she must fix some useful parameters such as the number of target computers which will participate in the work, the load of computing (data to be computed) assigned to each participating machine, etc. The SRTG middleware provides specific interfaces to assist users to fix these parameters. More details will be given in the next section.

### 4. Deployment of the Arabic OCR on the SRTG

The idea is to split the binary image of the Arabic text to be recognized into subimages and then assign them optimally or pseudo optimally among the targeted computers of the SRTG. These computers which will participate in the work will be assigned according to, mainly, their computing power. To make easier the optimal assignment (load balancing) between the targeted computers, we considered homogeneous computers each having the following configuration: 3 GHz CPU frequency; 512 Mega Octets as RAM capacity and Windows XP-professional as operating system. We used 25 computers and considered three randomly chosen Arabic texts corpuses. The first corpus is composed of 90 words; the second one is composed of 15,000 words and the third one is composed of 75,000 words. These texts were scanned by an HP scanner with a resolution of 300 dpi (dots per inch). These words have been assigned easily among targeted computers of the SRTG. The principle of the assignment process consists simply to divide the total number of Arabic connected words and pseudo words to be recognized by the number of computers which will participate in the work.

To assess the benefit of our application deployment (the data distribution), we first focus on the amount of reduction attained in the execution time. The question here is to evaluate how much gain in the execution time is accomplished by using the SRTG grid computing facility (the data distribution mode) as compared to just running our OCR application on a single processor (the sequential mode). This effect is known as the speedup factor which is formally defined as the ratio of the execution time using a single computer to the execution time using the SRTG. The efficiency factor which expresses the percentage of the CPU utilization of all computers of the SRTG participating in the work constitutes another important factor which is commonly used to assess the performance of such deployment. In our case, this factor is formally defined as the ratio of the speedup factor to the total number of computers participating in the work. We note here that the execution time in the sequential mode is performed on one of the 25 homogeneous target computers of the SRTG previously described.

#### 4.1 The Deployment Results

We have conducted three experiments in which the entire application has been written in C# (.Net). The reference library used was composed of 103 characters, representing approximately the totality of the Arabic alphabet (including the characters' shape variation according to their position in words). The SRTG network links capacities are in the range of 0.8–1.6 Mbps.

The first experiment attempts to show and confirm that the distribution over a grid computing may not always be advantageous when the Arabic text to be recognized is rather small. In this experiment, we varied the number of targeted computers from 2 to 25. Each computer is assigned a copy of the first considered corpus that composed of 90 words. The entire text to be recognized depends then on the number of targeted computers used; when all 25 computers are used, then the entire text amounts to 2,250 words and pseudo words. Fig. 1 illustrates the results of this experiment.

The lowest curve of Fig. 1(a) corresponds to the distributed execution and provides the execution time as a

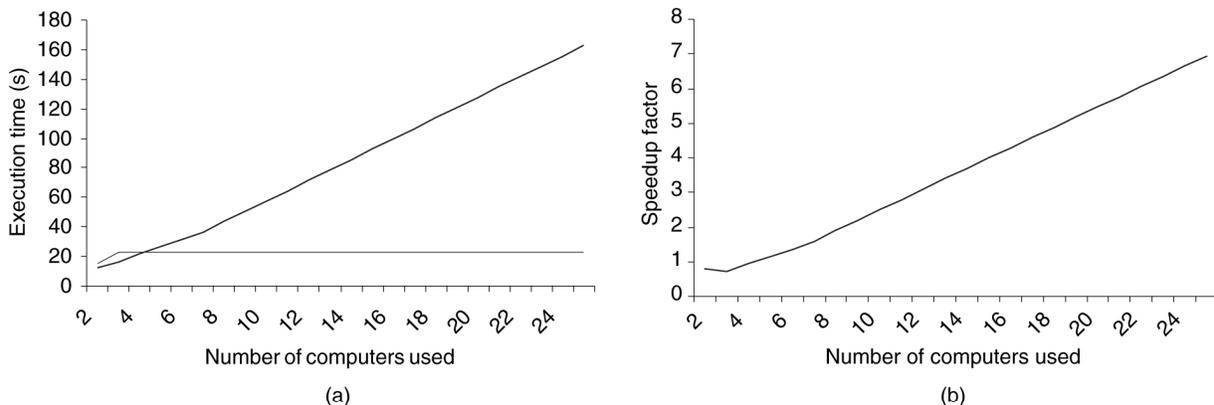


Figure 1. Results of the first experiment.

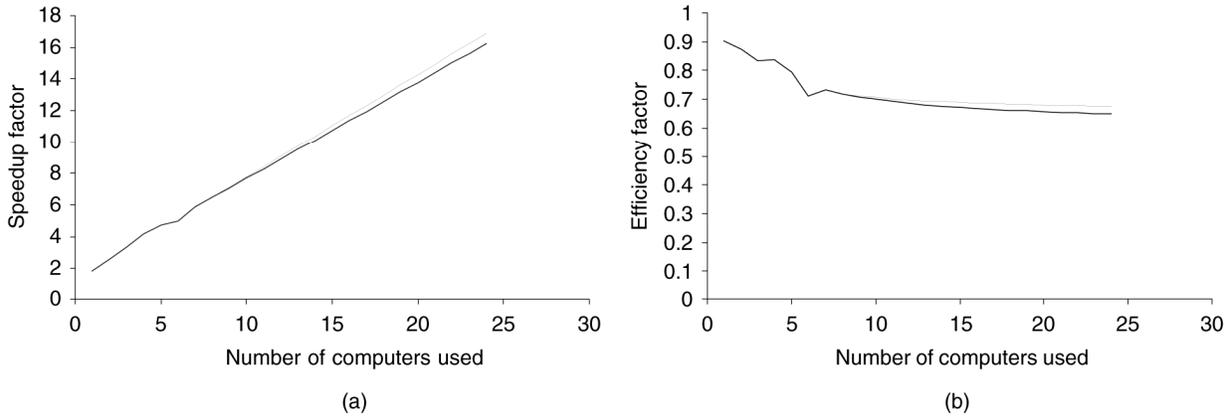


Figure 2. Results of the second and third experiments.

function of the number of targeted computers involved. The topmost curve of Fig. 1(a) corresponds to the sequential execution to recognize the exact same amount of text (90 words time the number of used computers). We notice in particular the following:

- The sequential mode requires less execution time than the distributed mode when the entire text to be recognized is smaller than 450 words; namely when the number of used targeted computers is less than or equal to five. Beyond that, the distributed mode prevails. This amounts to say that there is neither a need nor a benefit to run the DTW Arabic OCR application in a distributed manner when only a small amount of text is to be recognized. In other words, the underlying communication time prevails.
- The effect of the communication time weakens as the entire amount of text to be recognized gets larger (larger than 450 words), that is when more than five computers are used.
- The speedup factor represented in Fig. 1(b) remains below the unity until the entire text to be recognized gets larger than 450 words, or equivalently until the number of targeted used computers gets above five. When 25 computers are used the speedup factor leverages 7. To reach higher and more attractive speedup factors, we need to use much larger amounts of text as done in the following two experiments.

The second and the third experiments have been conducted on much larger Arabic texts, respectively, composed of 15,000 and 75,000 words. Here too, we varied the number of targeted computers from 2 to 25. The assignment of the text to targeted computers is done as described before in such a manner to enforce balancing their loads. The entire text is divided approximately equally among them (in fact to a given pseudo word). Consequently, the amount of text assigned per computer depends on the number of targeted computers. An XML file is designed to include the number of target computers, the data to be processed and the code to be executed by each participating computer. Fig. 2 portrays the speedup factor and the efficiency factor provided by each experiment.

The lowest curves in both Figs. 2(a) and 2(b) relate to the second experiment and the top most ones correspond

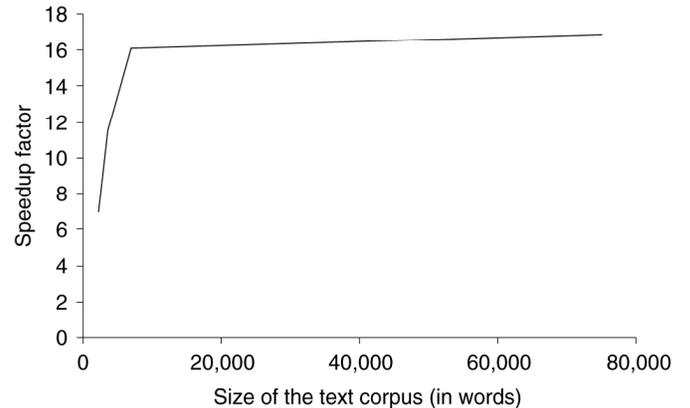


Figure 3. Speedup factor versus size of the text corpus.

to the third experiment. In particular, we notice the following:

- The speedup factor is an increasing function of the number of computers used.
- The speedup factor attained by the third experiment gets slightly higher than that reached by the second experiment as the number of used computers gets bigger. Although this still confirms the results of the first experiment, it also shows that once the text to be recognized gets beyond a certain amount, the speedup becomes rather sensible to the number of used computers and not to the amount of the text to be recognized [31]. This is mainly due to the complete fading out of the communication latency against the text recognition execution time. Fig. 3 portrays the speedup factor as a function of the size of the text corpus when all 25 computers are used. It shows that beyond a certain size of the text corpus, the speedup leverages its maximum.
- The speedup factor reaches the value of 16 (the top most curve) when the number of the SRTG computers used is 25. This amounts to a recognition rate of approximately 650 characters per second as the recognition rate of the sequential mode is around 40 characters per second. This clearly confirms that benevolent grid computing provides an adequate infrastructure for the DTW-based Arabic OCR. We recall that currently

available commercialized systems leverage much lower recognition rates [32].

- Fig. 2(a) shows that further increasing the text to be recognized amounts only to a slight increase in the efficiency factor. Only around 70% of each computer processing capacity is required when 25 computers are used. This amounts to leave around 30% processing capacity to users' running tasks and applications which is an appropriate percentage within the context of a benevolent grid or LAN computers. One should then target computers (such as office computers) that are not CPU extensively involved.

## 5. Conclusion and Perspectives

This paper showed that a P2P grid architecture such as the SRTG provides a very interesting platform to speed up the execution time of the DTW-based Arabic OCR application especially for large quantities of text. Conducted experiments showed that speedup and efficiency factors can be easily governed by users through appropriate tuning of the load-balancing procedure, the overlay topology connecting the targeted computers and the number of targeted computers. The recognition speed can reach very high values otherwise impossible on a single dedicated and specialized machine. We showed that such a speed is an increasing function of both the size of the text and the number of targeted computers.

Further investigations are underway to dynamically select the targeted processors and develop the necessary software tools that implement all the preprocessing phases in a distributed manner. This will necessarily contribute to the betterment of our distributed Arabic OCR application, yet it will yield an enhanced and more accurate assessment of our data distribution proposal performance. Refinements to the load-balancing procedure are also investigated to conduct further experiments on heterogeneous computers.

## References

- [1] M. Khemakhem & C. Fehri, Arabic type written character recognition using dynamic comparison, *Proc. First Kuwait Conf. on Computer Science*, Kuwait, March 1989, 448–462.
- [2] M. Khemakhem & A. Belghith, A multipurpose multi-agent system based on a loosely coupled architecture to speedup the DTW algorithm for Arabic printed cursive OCR, *Proc. IEEE-AICCSA-2005*, Cairo, Egypt, January 2005.
- [3] M. Khemakhem & A. Belghith, The DTW algorithm for distributed printed cursive OCR within a multi agent system, *Proc. ACM, ICICIS*, Cairo, Egypt, March 14–18, 2007.
- [4] H.D. Cheng & K.S. Fu, VLSI architecture for pattern matching using space-time domain expansion approach, *Proc. IEEE Int. Conf. on Computer Design VLSI and Computing*, NY, October 1985, 7–10.
- [5] H.-D. Cheng & K.S. Fu, A VLSI architecture for dynamic time-wrap recognition of handwritten symbols, *IEEE Transactions on Acoustics, Speech, & Signal Processing*, 34(3), June 1986.
- [6] G. Quénot, J.L. Gauvain, J.J. Gangolf, & J. Mariani, A dynamic programming processor for speech recognition, *IEEE Journal of Solid-State Circuits*, 24, N(F 9)q.20, April 1989.
- [7] M. Khemakhem, A. Belghith, & M. Ben Ahmed, Etude et Evaluation de deux méthodes de distribution de l'algorithme de comparaison dynamique pour la reconnaissance de caractères arabes, *Proc. First Maghrebien Symp. on Programming and Systems*, Algeria, October 1991.
- [8] P.G. Bradford, Efficient parallel dynamic programming, *Proc. 30th Annual Allerton Conf. on Communication, Control and Computing*, University of Illinois, 1992, 185–194.
- [9] M. Khemakhem, A. Belghith, & M. Ben Ahmed, Modélisation architecturale de la Comparaison Dynamique distribuée, *Proc. 2 Intl. Congress on Arabic and Advanced Computer Technology*, Casablanca, Morocco, December 1993.
- [10] Alves, E.N. Cáceres, & F. Dehne, Parallel dynamic programming for solving the string editing problem on CGM/BSP, *Proc. SPAA '02*, Winnipeg, Manitoba, Canada, August 10–13, 2002.
- [11] R. Buyya & S. Venugopal, A Gentle introduction to grid computing and technologies, *Proc. CSI*, India, May 7–19, 2005.
- [12] Z. Shi, H. Huang, J. Luo, F. Lin, & H. Zhang, Agent based grid computing, *Journal of Applied Mathematical Modelling*, 30, 2006, 629–640, available at: [www.sciencedirect.com](http://www.sciencedirect.com).
- [13] J.S. Bridle, M.D. Brown, & R.M. Chamberlain, An algorithm for connected word recognition, *Proc. IEEE, ICASSP*, May 1982, 899–902.
- [14] V. Vuori, J. Laaksonen, E. Oja, & J. Kangas, Experiments with adaptation strategies for a prototype-based recognition system for isolated handwritten characters, *International Journal of Document Analysis and Recognition*, 3, 2001, 150–159.
- [15] A. Kumar, A. Balasubramanian, A.M. Nambodiri, & C.V. Jawahar, Model-based annotation of online handwritten datasets, *Technical report, International Institute of Information Technology*, Gachibowli, Hyderabad, India, 2006.
- [16] E. Tapia & R. Rojas, A survey on recognition of on-line handwritten mathematical notation, *Technical Report B-07-01 Freie University at Berlin, Institut für Informatik Takustr. 9*, 14195 Berlin, Germany, January 26, 2007.
- [17] A. Amin, Off-line Arabic character recognition: The state of the art, *Pattern Recognition*, 31(5), 1998, 517–530.
- [18] M.A. Cheung, M. Bennamoun, & N. Bergmann, An Arabic optical character recognition system using recognition based segmentation, *Pattern Recognition*, 34, 2001, 215–233.
- [19] N.E.B. Amara, O. Mazhoud, N. Bouzrara, & N. Ellouze, A relational database for Arabic OCR system, *LAJIT*, 2(4), October 2005, 259–266.
- [20] F. Capello, The evolution of GRID5000, *Workshop on Grid Computing: e-Infrastructure, Applications and Research*, ES-STT, UTIC, Tunisia, 2007.
- [21] N. Abdennadher, Using the volunteer computing platform XtremWeb-CH: Lessons and perspectives, *Workshop on Grid Computing: e-Infrastructure, Applications and Research*, ES-STT, UTIC, Tunisia, 2007.
- [22] I. Foster, C. Kesselman, & S. Tuecke, The anatomy of the grid, *International Journal of Supercomputer Applications*, 2002.
- [23] Available at: <http://www.globus.org>.
- [24] IBM, *Introduction to Grid Computing with Globus IBM Redbook*, SG24-6895-01 ISBN 0738427969, September 2003.
- [25] I. Foster, N.R. Jennings, & C. Kesselman, Brain meets brawn: why grid and agents need each other, *Proc. AAMAS'04*, Leeds, UK, March 2004.
- [26] Available at: <http://www.esstt.rnu.tn/utic/gtrs/>.
- [27] Available at: <http://www.xtremwebch.net>.
- [28] Available at: <http://www.xtremweb.net>.
- [29] N. Abdennadher, XtremWeb-CH: A global framework for high performance computing applications, *Internal Report*, Switzerland, HES-SO/EIG, August 2004.
- [30] N. Abdennadher, Towards Peer-to-Peer tool for intensive computing, *Flash Informatique, EPFL*, Switzerland, August 2005.
- [31] M. Khemakhem & A. Belghith, Agent based architecture for parallel and distributed complex information processing, *International Revue on Computers and Softwares (IRECOS)*, 2(1), 2007, 38–44.
- [32] Worldlanguage products available at: <http://www.worldlanguage.com/Products/Arabic/OCR/Page1.htm>.

## Biographies



*Maher Khemakhem* received his M.Sc. and Ph.D. degrees from the University of Paris 11 (Orsay), France, respectively in 1984 and 1987. He is currently an Assistant Professor in Computer Science at the Faculty of Economy and Management Sciences at the University of Sfax, Tunisia. His research interests include distributed systems, performance evaluation and pattern recognition.



*Abdelfettah Belghith* received his M.Sc. and Ph.D. degrees from the University of California at Los Angeles (UCLA), respectively in 1982 and 1987. Since 1992 he is a Full Professor at the National School of Computer Science (ENSI), University of Mannouba, Tunisia. His research interests include computer networks, wireless networks, multimedia internet, mobile computing, distributed

algorithms, simulation and performance evaluation. He

runs several projects in cooperation with other universities and research laboratories and institutions. He is currently responsible for the graduate studies department and the Head of the HANA research group at ENSI.